

Geliş Tarihi/Received: 25.08.2025

Yayın Tarihi/Published: 28.03.2026

www.journalofstrategy.org

Kabul Tarihi/Accepted: 11.12.2025

[Derleme Makalesi/ Review Article](#)

Doi: 10.5281/zenodo.19282189

YAZILIM PROJE YÖNETİMİNDE ÇEVİK YAKLAŞIMLAR VE TEMEL İLKELER

AGILE APPROACHES AND BASIC PRINCIPLES IN SOFTWARE PROJECT MANAGEMENT

Eren ADIGÜZEL

Yalova Üniversitesi, Türkiye

ORCID ID: 0000-0002-1995-3181, erena1653@gmail.com**Enes ADIGÜZEL**

Yalova Üniversitesi, Türkiye

ORCID ID: 0000-0002-0990-0115, enesa3453@gmail.com**Prof. Dr. Ümit ÜNVER**

Yalova Üniversitesi, Türkiye

ORCID ID: 0000-0002-6968-6181, umitunver@gmail.com

ÖZET

Bu makale, geleneksel ve çevik (agile) proje yönetim yaklaşımlarını karşılaştırarak, agile metodolojilerin günümüzün dinamik ve belirsiz proje ortamlarında nasıl daha etkin sonuçlar sunduğunu incelemektedir. Geleneksel proje yönetimi detaylı planlama ve katı süreçlere dayanırken, agile metodolojiler esneklik, müşteri odaklılık ve hızlı geri bildirim mekanizmalarıyla öne çıkmaktadır. Çalışmada, scrum ve kanban gibi yaygın Agile metodolojilerin yapıları, avantajları ve süreçleri ele alınmış, jira, trello, asana ve MS Project gibi Agile proje yönetim yazılımları incelenmiştir. Ayrıca, Agile dönüşüm süreçlerinde karşılaşılan kültürel ve organizasyonel zorluklar incelenmiş ve bu süreçlerin başarıya ulaşabilmesi için organizasyonel kültürün uyumu, ekip eğitimleri ve doğru araç seçiminin kritik rolü vurgulanmıştır. Bu çalışmada, geleneksel ve Çevik proje yönetim metodolojilerini karşılaştırmalı olarak ele alarak, bağlamsal uygunluğun yöntem seçiminde belirleyici olduğunu ortaya koyulmuştur. Bulgular, özellikle Scrum ve Kanban gibi çevik yaklaşımların belirsizlik içeren projelerde daha yüksek uyarlanabilirlik ve başarı sağladığını göstermektedir. Bununla birlikte, çevik uygulamaların etkinliği; organizasyonel yapı, kültürel faktörler ve dönüşüm süreçlerinin yönetimi ile doğrudan ilişkilidir.

Anahtar Kelimeler: Çevik proje yönetimi, geleneksel proje yönetimi, proje yönetim araçları.

ABSTRACT

This article compares traditional and agile project management approaches and examines how agile methodologies deliver more effective results in today's dynamic and uncertain project environments. While traditional project management relies on detailed planning and rigid processes, agile methodologies stand out with their flexibility, customer focus, and rapid feedback mechanisms. The study examines the structures, advantages, and processes of common agile methodologies such as Scrum and Kanban, and examines agile project management software such as Jira, Trello, Asana, and MS Project. Furthermore, the study examines the cultural and organizational challenges encountered in agile transformation processes, emphasizing the critical role of organizational cultural alignment, team training, and the right tool selection for the success of these processes. This study compares traditional and agile project management methodologies, demonstrating that contextual suitability is a determining factor in method selection. Findings indicate that agile approaches, particularly Scrum and Kanban, provide greater adaptability and success in projects characterized by uncertainty. Furthermore, the effectiveness of agile practices is directly related to organizational structure, cultural factors, and the management of transformation processes.

Keywords: Agile project management, traditional project management, project management tools.



1. Giriş

21. yüzyılda bilim ve teknoloji alanındaki hızlı gelişmeler, işletmeleri rekabetçi ortamda ayakta kalabilmek için yenilikçi projeler geliştirmeye zorunlu kılmaktadır. Bu durum, proje yönetiminin yalnızca teknik bir uygulama alanı olmaktan çıkarak farklı disiplinlerin ilgi duyduğu stratejik bir faaliyet hâline gelmesine katkı sağlamıştır (Kir, 2007).

Proje kavramı literatürde çeşitli şekillerde tanımlanmakla birlikte, temel bileşenler açısından ortak bir çerçeveye sahiptir. Proje; belirli bir başlangıç ve bitiş tarihine sahip, benzersiz bir ürün veya hizmet üretmeyi amaçlayan ve zaman, maliyet ile kapsam gibi kısıtlar altında yürütülen geçici bir çabadır (PMI, 2003; Kerzner, 2017). Bu tanım, projelerin hem geçicilik özelliğini hem de hedef-kısıt dengesini vurgulayarak farklı disiplinlerde yapılan tanımları ortak bir paydada birleştirmektedir.

Projelerin kapsam ve yapılarının giderek daha karmaşık hâle gelmesi, proje yönetiminin önemini artırmakta ve organizasyonların bu süreci yalnızca teknik bir gereklilik değil, kurumsal bir başarı unsuru olarak benimsemelerine neden olmaktadır (Kömürlü & Toltar, 2018).

Literatürdeki tanımlar birlikte ele alındığında, proje kavramının üç ortak ekseninde toplandığı görülmektedir; belirli bir başlangıç ve bitiş noktası olan geçici bir organizasyon olması, benzersiz bir ürün/hizmet/sonuç üretmesi ve zaman, maliyet ve kapsam gibi kısıtlar altında yürütülmesi (Cleland, 1999; Project Management Institute, 2003; Shenhar, 1998). Bu çalışma kapsamında proje, özgün bir ürün veya hizmet üretmek üzere, sınırlı süre ve kaynak altında, belirsizlik içeren ve çok paydaşlı bir değişim girişimi olarak ele alınacaktır (Shenhar, 1998; PMI, 2003).

Yönetim kavramının köklü bir geçmişi olmakla birlikte, günümüz bilgi çağında önemi daha da artmıştır. Geçmişte insanlar, bugünkü araçlar ve metodolojiler olmadan projeleri tasarlayıp planlamış ve yönetmişlerdir (Ece & Kovancı, 2004). Mısır Piramitleri ve Çin Seddi gibi yapılar bazı çevrelerce önemli projeler olarak kabul edilmektedir (Ece & Kovancı, 2004; Demirel, 2014). Ancak proje yönetiminin kökeninin, ABD ordusunun 1941'de başlattığı Manhattan Projesi'ne dayandığı daha geniş çapta kabul görmektedir. Proje yönetimi üzerine ilk bilimsel çalışmalar Frederick Taylor (1856–1915) tarafından 1900'lerin başlarında yapılmıştır. Taylor, yönetim tekniklerinin bilimsel yöntemlerle analiz edilip geliştirilebileceğini göstererek modern yönetim uygulamalarının temelini atmıştır (Demirel, 2014).

Bazı kaynaklarda proje yönetimi şu şekilde tanımlanmaktadır:

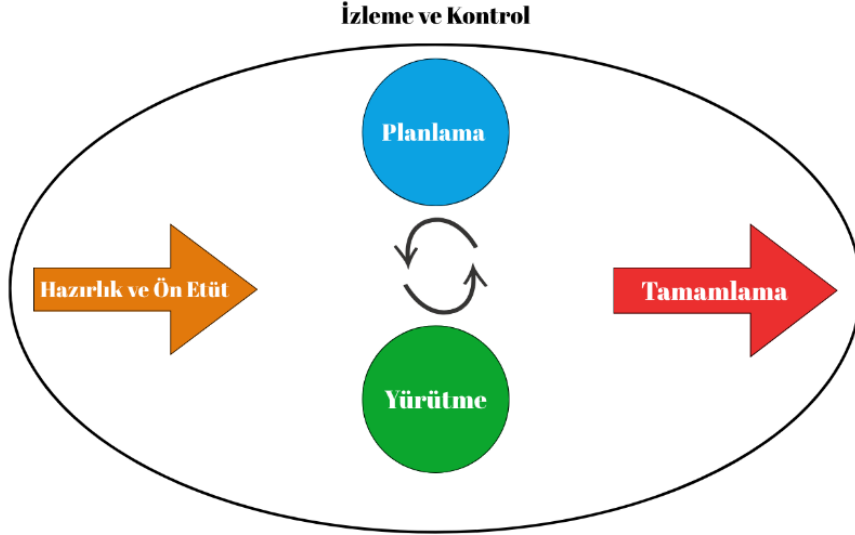
- Proje yönetimi; bir projenin belirlenen hedeflerine ulaşmasını sağlamak için kaynakları planlama, organize etme, temin etme ve etkin şekilde yönetme disiplini (Anonim, 2025).
- Proje yönetimi, farklı organizasyonların proje kapsamı, süresi, maliyeti ve ölçeğine uygun şekilde bir araya getirilmesi ve uyum içinde yönetilmesi sürecidir (Kocakulak, 1997).
- Proje yönetimi; proje hedeflerine ulaşmak için faaliyetlerin planlanması, zamanlanması ve sürekli izlenmesidir (Lewis, 1996).



Proje yönetim metodolojisinin temel amacı, projelerin başarıyla tamamlanma olasılığını artırmaktır (Chin & Spowage, 2010; Kerzner, 2001; Milosevic & Patanakul, 2005). Genel olarak proje yönetim süreçleri; Hazırlık ve Ön Etüt, Planlama (Tanımlama), Uygulama (Yürütme, Kontrol ve İzleme), Tamamlama olmak üzere dört ana aşamadan oluşur (Gido & Clements, 1999):

Şekil 1

Proje Yönetim Süreçleri



Kaynak: Yazarlar tarafından oluşturulmuştur.

En yaygın proje yönetim metodolojileri; Project Management Institute methodology (PMI, Proje Yönetimi Enstitüsü metodolojisi), International Project Management Association methodology (IPMA, Uluslararası Proje Yönetimi Derneği metodolojisi), Projects IN Controlled Environments (PRINCE2, Kontrollü Ortamlarda Projeler metodolojisi), Youth Project Management Association methodology (YUPMA, Gençlik Proje Yönetimi Derneği metodolojisi), Association for Project Management methodology (APM, Proje Yönetimi Derneği metodolojisi), Harvard Business School methodology (HBS Harvard İşletme Okulu metodolojisi), Çevik Yöntem (Agile methodology) olarak sıralanabilir. Bu proje yönetim sistemleri arasında hem basitliği ile hem de yaygın olması sebebiyle en çok tercih edilen yönetim sistemi geleneksel yaklaşımların katı ve zayıf yönlerini ortadan kaldıran çevik yaklaşımdır.

Son yıllarda yazılım projelerinin bağlamı, klasik çevik literatürün ötesinde yeni bileşenlerle daha da karmaşık bir hâle gelmiştir. Birincisi, yazılım geliştirmede yapay zekâ destekli araçlar (ör. GitHub Copilot, ChatGPT tabanlı asistanlar) yaygınlaşmış ve geliştirici üretkenliğinde anlamlı artışlar sağlandığı ampirik çalışmalarla gösterilmiştir (Peng vd., 2023; Dohmke vd., 2023; Frank & Godwin, 2024). İkincisi, COVID-19 sonrası dönemde uzaktan ve hibrit çalışma düzenleri kalıcı hâle gelmiş; bir yılı aşkın sürede yürütülen uzunlamasına çalışmalar, çevik ekiplerde özellikle iletişim ve takım içi sosyal etkileşimin etkilendiğini, buna karşın teknik uygulamaların görece daha az değiştiğini ortaya koymuştur (Ågren vd., 2022). Üçüncüsü, çevik yaklaşımlar giderek DevOps, bulut bilişim ve sürekli teslim pratikleriyle bütünleşmekte; bu entegrasyonun hem faydaları hem de uygulama zorlukları sistematik derlemelerle ortaya konmaktadır (Aouni vd., 2024). Son olarak, büyük



ölçekli kurumsal dönüşümlerde SAFe 6.0 gibi ölçekli çevik çerçeveler, iş çevikliğini ve değer akışı odaklılığını vurgulayarak çevik uygulamaların organizasyon geneline yaygınlaştırılmasına yönelik güncel rehberlik sunmaktadır (Ahrend, 2023; Scaled Agile, 2023).

Bu çalışmada geleneksel yaklaşım, çevik yaklaşımla kıyaslanarak tanıtılacaktır. Herhangi bir projenin yönetiminde kullanılacak yaklaşım, projenin özel gereksinimlerine ve doğasına göre belirlenir.

1.1 Geleneksel Proje Yönetimi

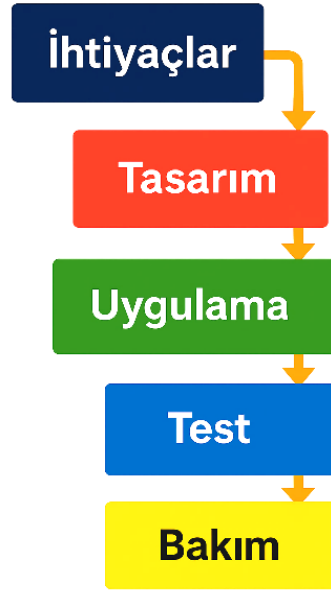
Geleneksel proje yönetimi anlayışı, özellikle savunma sanayi (Andersen, 2006; Boehm, 2002; Boehm & Turner, 2003), inşaat (Cicmil vd., 2009; Collyer vd., 2010; DeCarlo, 2004] ve üretim (Leffingwell, 2007; Saynisch, 2010; Shenhar & Dvir, 2007) gibi hataya yer olmayan sektörlerde doğmuş ve gelişmiştir. Bu yaklaşım, projelerin başlangıçta net şekilde tanımlandığı, değişimin minimumda tutulduğu ortamlarda yüksek verimlilik sağlamıştır (Williams, 2005). 1950’li yıllarda geliştirilen bu yöntemler, süreçlerin aşamalı, sıralı ve kontrol odaklı yönetilmesini esas alarak, tutarlılığı ve kaynak yönetiminde disiplini ön plana çıkarmıştır. Geleneksel yaklaşımın temelinde, projelerin genellikle öngörülebilir, doğrusal ve sınırlarının net bir şekilde tanımlanmış olduğu varsayımı yer almaktadır. Bu öngörülebilirlik, detaylı planlamayı mümkün kılar ve proje planlarının minimum sapmayla hayata geçirilmesini sağlar (Wysocki, 2007).

Geleneksel proje yönetiminin temel amacı; projelerin önceden tanımlanmış zaman, bütçe ve kapsam sınırları içinde, ayrıntılı bir planlama ile verimli ve etkili şekilde yürütülmesini ve başarılı sonuçların elde edilmesini sağlamaktır (Shenhar & Dvir, 2007; Wysocki, 2007).

Geleneksel proje yönetim çerçeveleri içinde en yaygın kullanılan yöntemlerden biri, Şelale Metodolojisi’dir. Bu yöntemde her aşama, tamamlanmadan bir sonrakine geçilmez ve süreç doğrusal bir ilerleyiş izler. Şelale metodolojisi yaklaşımı Şekil 2’de gösterilmiştir.

Şelale yaklaşımı doğası gereği basamaklıdır ve projenin başlangıcında tüm gereksinimlerin net şekilde tanımlanmasını zorunlu kılar (Şekil 2). Bu gereksinimlerin açık ve kapsamlı bir şekilde ortaya konması, proje ilerleyişi için ön şart olarak kabul edilir. Ancak, bu yaklaşımın en büyük sınırlamalarından biri, proje başladıktan sonra değişikliklere veya revizyonlara neredeyse hiç esneklik tanımamasıdır. Müşteriler çoğu zaman başlangıç aşamasında tüm gereksinim ve beklentilerini tam olarak ifade edemezler; bu durum, özellikle dinamik yapıya sahip projelerde ilerleyen safhalarda değişiklik taleplerinin ortaya çıkmasını kaçınılmaz hale getirir. Bu tür değişiklikler zaman çizelgelerini bozabilir ve planların kesintiye uğramasına sebep olabilir. Şelale metodolojisinin değişime hızlı uyum sağlayamaması ve esnek olmaması temel kısıtlarından biridir. Özellikle belirsizlik ve öncelik değişimlerinin sık yaşandığı uzun vadeli projelerde, bu katı ve değiştirilemez yapı, önemli sorunlara yol açabilmektedir (Akça, 2022).



Şekil 2*Şelale (Waterfall) Modeli*

Kaynak: Yazarlar tarafından oluşturulmuştur.

Ancak son yıllarda araştırmacılar, "herkese uyan tek bir yöntem" yaklaşımının modern projelerin karmaşıklığını karşılamada yetersiz olduğunu vurgulamaktadır (Aguanno, 2004; Chin, 2004; Shenhar, 1998). İş dünyasındaki hızlı değişimler, projelerin yapısında büyük ölçüde karmaşıklık yaratmıştır. Görevler, bağımlılıklar, sorumluluklar ve diğer proje öğeleri yönetmeyi daha zor hale getirmiştir.

Geleneksel proje yönetim yöntemlerinin katı, hiyerarşik ve doğrusal yapısı, çağdaş projelerin dinamik ve öngörülemeyen taleplerini karşılamada yetersiz kalmaktadır (Cicmil vd., 2006). Bu yapı, öngörülebilirliği esas alan planlara dayanırken, günümüz projeleri çoğu zaman belirsizlik, hızlı değişim gösteren ve çok paydaşlı karar süreçlerini içermektedir. Bu nedenle geleneksel yaklaşımlar, esneklik gerektiren projelerde işlevselliğini yitirebilmektedir. Proje yönetiminde bu esneklik eksikliği, yalnızca iç süreçlerle sınırlı kalmamakta; dış çevresel etkilere karşı da yetersiz kalmaktadır.

Projeleri dış çevresel faktörlerden izole etmeye çalışmak, geleneksel proje yönetimi anlayışının bir diğer temel sınırlılığı olarak öne çıkmaktadır (Shenhar & Dvir, 2007; Aguanno, 2004). Bu yaklaşımın temelinde, projenin kontrol edilebilir sınırlar içinde yürütülebileceği varsayımı yer almaktadır. Ancak günümüz koşullarında bu sınırların dışındaki ekonomik, teknolojik ya da sosyo-politik gelişmelerin projeler üzerindeki etkisi giderek artmaktadır. Bu artan dışsal baskılar, projeleri sabit planlardan uzaklaştırmakta ve değişimle başa çıkma yeteneğini zorunlu hale getirmektedir.

Günümüz iş dünyasında değişim, projelerin kaçınılmaz ve sürekli karşılaştığı bir gerçeklik haline gelmiştir (Highsmith & Cockburn, 2001). Dış çevrenin etkisine açık olan projeler, aynı zamanda içsel olarak da değişime uyum sağlayabilecek bir yönetsel esnekliğe ihtiyaç duymaktadır. Sürekli değişen dinamikler, modern proje yönetim uygulamalarının ayrılmaz bir parçası haline gelmiştir (Olsson, 2006).



1.2 Çevik Proje Yönetimi

Çevik metodoloji, değişen koşullara hızlı ve etkili şekilde uyum sağlama esnekliği, kapasitesi ve yeteneği olarak tanımlanmaktadır (Akdağ, 2020). Bu kavram, dış çevredeki değişimlere proaktif ve dinamik bir yaklaşım sergileyerek, organizasyonların değişen koşullarda çevik ve dayanıklı kalmalarına olanak tanımaktadır (Özbay vd., 2020). Çevik yöntemlerin pratikteki başarısını doğru biçimde değerlendirebilmek için, yaklaşımın Agile Manifesto’da yer alan değer ve ilkeler ile yalın düşünce felsefesi ve karmaşıklık kuramı gibi kuramsal temeller üzerine inşa edildiğinin ortaya konması önemlidir.

Çevik yaklaşımın felsefi temelini oluşturan Agile Manifesto, dört temel değer ve on iki ilke ile çevik ekiplerin karar alma ve uygulama biçimlerini tanımlar (Beck vd., 2001; Highsmith & Cockburn, 2001). Bu değerlere göre, süreç ve araçlardan çok bireyler ve etkileşimler, kapsamlı dokümantasyondan çok çalışan yazılım, sözleşme pazarlığından çok müşteri ile iş birliği ve plana bağlılıktan çok değişime hızlı uyum ön plana çıkar (Beck vd., 2001). İlkeler ise, sık teslimat, müşteriyle yakın iş birliği, kendi kendini organize eden ekipler ve düzenli geri bildirim döngüleriyle, belirsizlik ve değişimin yüksek olduğu projelerde öğrenme ve uyum kapasitesini sistematik olarak artırmayı hedefler (Highsmith & Cockburn, 2001; Denning, 2018).

Yalın düşünce, israfın sistematik olarak ortadan kaldırılması, değerın müşteri perspektifinden tanımlanması ve akışın sürekli iyileştirilmesi ilkelerine dayanır (Womack & Jones, 2003). Yazılım geliştirme bağlamında bu ilkeler, Poppendieck ve Poppendieck (2003) tarafından “yalın yazılım geliştirme” prensipleri olarak uyarlanmış; gereksiz özelliklerin azaltılması, erken geri bildirim ve öğrenme döngülerinin kısaltılması ile çevik yöntemlerin temel dayanaklarından biri hâline gelmiştir.

Karmaşıklık kuramı perspektifinden bakıldığında projeler, lineer neden–sonuç ilişkileriyle tam olarak öngörülemeyen, sürekli etkileşim hâlindeki çok sayıda aktörden oluşan uyumlu karmaşık sistemler olarak görülmektedir (Cicmil vd., 2006; Williams, 2005). Bu nedenle, özellikle yazılım projelerinde, ayrıntılı ve değişmez planlara dayanan yaklaşımların sınırlı kaldığı; bunun yerine deneme-yanılma, geri bildirim döngüleri ve küçük adımlarla ilerleyen çevik uygulamaların karmaşıklığı yönetmek için daha uygun olduğu öne sürülmektedir (Saynisch, 2010; Highsmith, 2004).

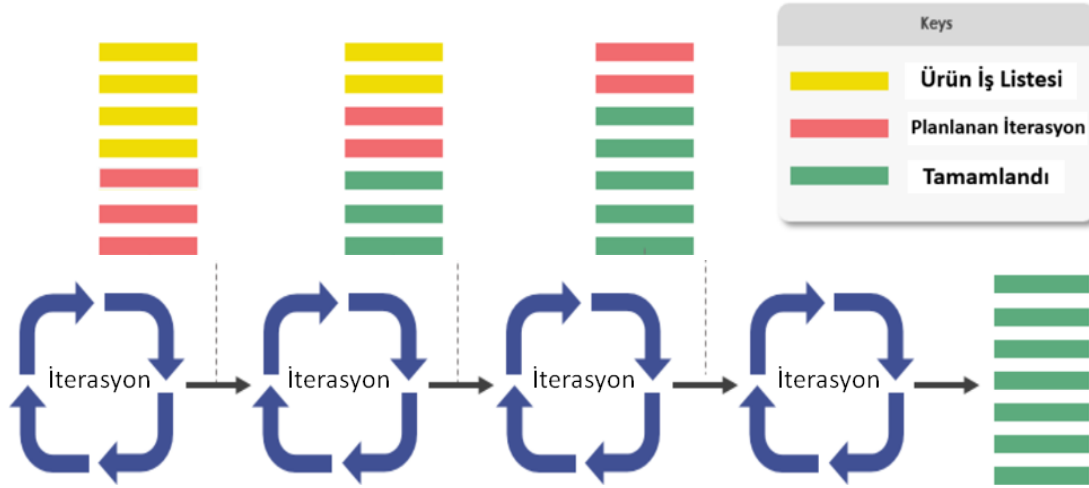
Çevik proje yönetimi, projelerin kademeli ve yinelemeli geliştirilmesini vurgular ve her aşamada ortaya çıkan yeni ihtiyaçlara ve değişen gereksinimlere uyum sağlama olanağı sunar. Bu yaklaşım, proje paydaşları arasında sürekli ve şeffaf iletişimi öncelikli hale getirerek, proje yaşam döngüsü boyunca zamanında uyarlamalar ve iyileştirmeler yapılmasını sağlayan etkili geri bildirim döngüleri oluşturur (Duyguvar vd., 2014).

Çevik yazılım geliştirme metodolojisi, iterasyon ve artımlı geliştirme (Iterative and Incremental Development) modellerinden türetilmiş olup, işlevsel yazılımların küçük ve yönetilebilir parçalarda sık sık teslim edilmesini teşvik etmektedir. Proje iş akışlarının daha küçük ve net bileşenlere bölünmesiyle, her iterasyon sonunda son kullanıcıya anında ve ölçülebilir değer sunulması amaçlanır. Bu döngüsel iyileştirme ve teslim süreci, "iterasyon" olarak adlandırılmakta olup, çevik yazılım geliştirme yaklaşımının temelini oluşturmaktadır (Yılmaz, 2018).



Şekil 3

Çevik Geliştirme Süresinde İterasyon Döngüsü



Kaynak: Yazarlar tarafından oluşturulmuştur.

Şekil 3, ürün iş listesinde (product backlog) yer alan iş gereksinimlerinin önceliklendirilerek her iterasyonda planlandığı, tamamlandığı ve çıktıya dönüştürüldüğü çevik geliştirme sürecini adım adım göstermektedir. Çevik model, müşteriye mümkün olan en kısa somut fayda sağlayan işlevsel bir çıktı teslim etmeyi hedefleyen yinelemeli ve işbirlikçi bir proje yönetimi yaklaşımını temsil eder. Geleneksel metodolojilerin uzun vadeli ve katı proje planlarına dayanmasına karşın çevik yaklaşım, birkaç haftalık kısa iterasyonlar şeklinde yapılandırılmış zaman dilimlerine sahiptir. Her iterasyonun başında proje ekipleri ve paydaşlar bir araya gelerek, öncelikli görevleri belirler ve geliştirme sürecinin odak noktalarını oluşturur. İterasyon sonunda bu görevler tamamlanır, gözden geçirilir ve iyileştirilir. Böylece ürün üzerinde aşamalı ilerleme ve somut iyileştirmeler sağlanır. Bu döngüsel süreç, üründe istenen işlevsellik ve olgunluk düzeyine ulaşana kadar devam eder. Çevik metodolojiler şeffaf iletişimi, sürekli bilgi paylaşımını ve kesintisiz geri bildirim döngülerini vurgulamakta; böylelikle geliştirme sürecinin etkinliğini, kalitesini ve uyarlanabilirliğini artırmaktadır. Ayrıca çevik yaklaşım, ekip üyeleri arasında güçlü bir iş birliği ve ortak sorumluluk kültürü oluşturarak, projelerin değişen gereksinimlere ve dinamik zorluklara hızlı yanıt verebilmesini mümkün kılmaktadır (Agile Alliance, 2025a; Agile Alliance, 2025b; Schwaber & Sutherland, 2012).

Şekil 4, çevik proje yönetiminde bir çözümün yaşam döngüsünü oluşturan aşamaları döngüsel biçimde göstermektedir. Kavramsallaştırmadan başlamak üzere, fikir oluşturma, geliştirme, sürüm yayınlama, üretim ve bakım süreçleriyle devam eden bu fazlar; ürün istenen değere ulaştığında emeklilikle (kaldırılmayla) sonuçlanır. Bu yapı, çevik metodolojilerin yalnızca yinelemeli geliştirme değil, aynı zamanda sürdürülebilir ve yönetilebilir bir yaşam döngüsü sağladığını ortaya koymaktadır (Spiceworks, 2025).



Şekil 4

Çevik Proje Yönetimi Döngüsünün Temel Fazları



Kaynak: Yazarlar tarafından oluşturulmuştur.

Çevik yaklaşımlar başlangıçta yazılım geliştirme süreçlerinde ortaya çıkmış olup, zamanla sundukları esneklik, uyarlanabilirlik ve verimlilik sayesinde farklı sektörlerde de hızla benimsenmiştir. Bu metodolojiler, modern proje ortamlarının karmaşıklığı ve belirsizlikleriyle başa çıkmada etkili olmuş, organizasyonların değişen gereksinimlere proaktif biçimde yanıt verebilmelerini sağlamıştır. Özellikle sağlık sektörü, yüksek değişkenlik gösteren gereksinimleri ve kritik zaman planları nedeniyle çevik yaklaşımların doğal bir uygulama alanı haline gelmiştir. Sağlık projelerinde çevik uygulamalar, hasta bilgi yönetimi sistemlerinin geliştirilmesinden hasta bakım süreçlerinin iyileştirilmesine kadar uzanan geniş bir yelpazede etkisini göstermiştir (JMIR, 2021). Literatürde, sağlık hizmeti sunumunda scrum ve benzeri çevik çerçevelerin kullanılmasıyla elde edilen başarı örneklerine rastlanmakta; bu sayede hasta memnuniyeti, iletişim ve süreç iyileştirmelerinde anlamlı gelişmeler kaydedildiği belirtilmektedir (Greenlight Guru, 2020). Nitekim sağlık kurumlarında çevik yöntemlerin uygulanması, hızlı değişen hasta ihtiyaçlarına uyum sağlama ve disiplinler arası ekiplerin etkin koordinasyonunda önemli faydalar sunmuştur (Taylor & Francis, 2020). Benzer şekilde, inşaat sektörü de çevik proje yönetimi prensiplerinden yararlanma yönünde girişimlere sahne olmaktadır. Geleneksel proje yönetimi tekniklerinin katılığı karşısında, inşaat projelerinde belirsizlik ve değişikliklerin yönetimi için daha esnek ve uyarlanabilir yaklaşımlara ihtiyaç duyulmaktadır. Bu kapsamda bazı inşaat projelerinde küçük ölçekli yinelemeli planlama, sık aralıklarla geri bildirim alma ve değişen şartlara hızla tepki verme gibi çevik uygulamaların uyarlandığı görülmüştür (Quickbase, 2020). Lakin, fiziksel üretime dayalı iş süreçlerinin getirdiği ardışıklık ve bağımlılıklar nedeniyle, çevik yaklaşımların inşaat projelerine entegrasyonu sınırlı kalmakta ve sektöre özgü uyarlamalara gereksinim duyulmaktadır (Prakash vd., 2022). Yine de literatürde, inşaat projelerinde çevik yöntemlerin kısmen de olsa uygulanabildiğine ve belirli alanlarda iyileştirmeler sağladığına dair bulgular rapor edilmiştir (Quickbase, 2020). Bununla birlikte, çevik proje yönetiminin en verimli ve etkin şekilde uygulandığı sektörün, doğası gereği bu yaklaşıma en uygun ortamı sunan yazılım sektörü olduğu vurgulanmalıdır.



Yazılım projeleri; değişen müşteri gereksinimleri, belirsizliklerle dolu geliştirme süreçleri ve sürekli teslimat gereksinimi gibi özellikleriyle çevik felsefenin ortaya çıktığı ve en olgunlaştığı alan konumundadır. Dolayısıyla bu çalışmada yazılım sektörü odak alınacaktır. Diğer sektörlere değinilmekle birlikte, çevik yaklaşımların temel ilkeleri ve uygulamaları yazılım projeleri ekseninde incelenecektir. Bu nedenle, yazılım sektöründe çevik metodolojilerin yaygınlaşmasını sağlayan yapısal avantajların daha yakından incelenmesi önem taşımaktadır.

Çevik metodolojilerin sağladığı avantajlar şu şekilde kategorize edilebilir:

- Doğrudan İletişim ve Kusursuz Entegrasyon: Çevik metodolojiler, müşteriler ile proje ekip üyeleri arasında yüz yüze iletişimi teşvik ederek, iş hedefleri ile yazılım geliştirme süreçleri arasında güçlü bir uyum sağlar. Bu doğrudan etkileşim, yanlış anlamaların önüne geçilmesini kolaylaştırır, gereksinimlerin net bir şekilde ifade edilmesine olanak tanır ve nihai ürünün işlevsel kalitesini artırarak müşteri memnuniyetine katkı sağlar (Özdemir, 2020).
- Sürekli Geri Bildirim ve Yinelemeli İyileştirme: Çevik yaklaşım, artımlı ve yinelemeli teslimatları teşvik ederek, her aşamada toplanan geri bildirimlerin bir sonraki iterasyonda hızla uygulanmasına imkân tanır. Bu döngü, süreç verimliliğini artırır, bakım süreçlerini kolaylaştırır ve uzun vadeli bakım maliyetlerini düşürür. Ayrıca, değişen proje dinamiklerine ve paydaş ihtiyaçlarına uyum sağlayabilen sürdürülebilir bir geliştirme modeli desteklenmiş olur.

2. Çevik Proje Yönetimi ile Geleneksel Proje Yönetiminin Karşılaştırılması

Çevik proje yönetimi, çeşitli açılardan geleneksel proje yönetim metodolojilerinden belirgin şekilde ayrılmaktadır. Başlıca farklılıklar Tablo 1’de özetlenmiştir (Şpundak, 2014).

Tablo 1

Geleneksel ve Çevik Yaklaşımlar Arasındaki Farklılıklar

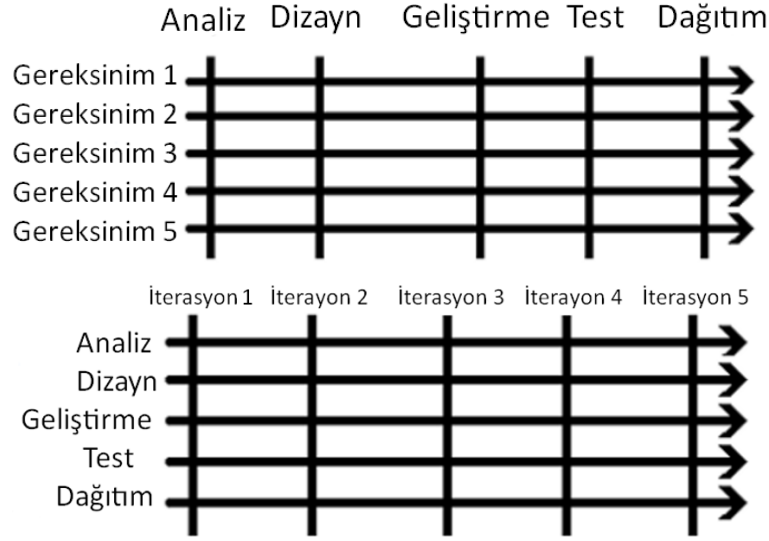
Özellik	Geleneksel	Çevik
İhtiyaçlar	Net ihtiyaçlar, düşük değişkenlik	Belirsiz ihtiyaçlar ve artan değişkenlik
Kullanıcılar	Katılımcılar	Yakın ve iş birliğine dayalı
Dokümantasyon	Standart yazılı dokümantasyon	Yazılı ve sözlü
Proje Kapsamı	Daha büyük projeler	Göreceli olarak daha küçük projeler
Kurumsal Destek	Mevcut iş süreçleri	Çevik çerçeveler ve yeni süreçler
Ekip Üyeleri	Dağınık	Bir arada ve yakın çalışır
Başarısızlık Etkisi	Çok kritik	Daha az kritik
Proje Planı	Doğrusal	Karmaşık ve yinelemeli

Bu metodolojilerin temel süreç akışları aşağıda özetlenmiştir (Şekil 5). Şelale modeli, proje sürecini başta belirlenen gereksinimlere dayalı olarak sıralı aşamalarla ilerletir. Buna karşın, çevik metodolojiler esnek ve dinamik bir şekilde yinelemeli döngülerle ilerler. Şelale modelinde çıktı ürünler, projenin başlangıcında tanımlanan gereksinimler ve öncelikler doğrultusunda yapılandırılır ve bu doğrultuda düzenli retrospektif değerlendirmeler gerektirir. Çevik metodolojiler ise daha ileriye dönük bir bakış açısı benimseyerek, değişen gereksinimlere ve önceliklere uyum sağlar. Şelale modeli, yapılandırılmış ve sistematik süreçlere dayanan bir mühendislik yaklaşımını temsil ederken; Çevik metodolojiler bireyleri ve iletişimi daha ön planda tutarak yaratıcılığı destekler (Palmquist vd., 2013; Fair, 2012; Wells vd., 2015).



Şekil 5

Geleneksel ve Çevik Proje Yönetimi Süreç Akışı



Kaynak: Yazarlar tarafından oluşturulmuştur.

Çevik ve geleneksel yaklaşımlarla gerçekleştirilen projelerin karşılaştırmalı değerlendirmesi Tablo 2’de sunulmuştur (Dhani, 2016).

Tablo 2

Metodolojilerin Değerlendirilmesi

Tanım	Geleneksel Metodolojiler						Çevik (Agile) Metodolojiler		
	Şelale	Sarnal Model	V Model	Artımlı Model	Prototipleme	Evrimsel Geliştirme	Scrum	Kanban	RUP
Gereksinimlerin Belirlenmesi	Başlangıç	Belirli Sıklıkla	Başlangıç	Belirli Sıklıkla	Belirli Sıklıkla	Başlangıç	Belirli Sıklıkla	Belirli Sıklıkla	Belirli Sıklıkla
Dokümantasyon ve Eğitim Gereksinimi	Y	O	O	O	O	O	Y	Y	Y
Maliyet	Y	Y	Y	O	D	D	Y	Y	Y
Risk Hassasiyeti	Y	D	O	D	D	Y	D	D	D
Başarı Garantisi	D	Y	D	Y	O	Y	Y	Y	Y
Esneklik	D	Y	D	Y	Y	D	Y	Y	Y
Uzmanlık Gereksinimi	O	Y	O	O	O	O	Y	Y	Y
Süre Uzunluğu	Y	Y	Y	Y	Y	Y	D	O	O
Bakım	O	Y	O	Y	Y	D	Y	Y	Y
Yönetilebilirlik / Basitlik	Y	D	O	O	O	D	D	D	D
Uygulama Derecesi	Y	O	Y	Y	Y	Y	Y	Y	Y

*D: Düşük, O: Orta, Y: Yüksek

Kaynak: Yazarlar tarafından oluşturulmuştur.

Tablo 2, geleneksel ve çevik proje yönetim metodolojilerini çok boyutlu olarak değerlendirerek her yöntemin belirli kriterler üzerindeki konumunu ortaya koymaktadır.



Şelale, Sarmal (Spiral) ve V Model gibi geleneksel yaklaşımlar; yüksek dokümantasyon ihtiyacı, maliyet ve zaman uzunluğu ile öne çıkarken, genellikle daha az esnek ve daha yüksek risk taşıyan yapılar sunmaktadır. Bu yöntemler, planlamanın ön planda olduğu ve değişime daha kapalı süreçler için yapılandırılmıştır. Örneğin, askeri sistem geliştirme projeleri, büyük altyapı yatırımları ya da havacılık sektörü gibi yüksek güvenlik ve düzenleyici gereksinimlerin olduğu alanlarda genellikle geleneksel yöntemler tercih edilir. Bu tür projelerde değişikliklerin minimumda tutulması ve ayrıntılı belgelerin bulunması zorunlu olduğundan dolayı, geleneksel modeller daha uygun görülmektedir.

Buna karşılık, Scrum ve Kanban gibi çevik yöntemler; yüksek esneklik, düşük risk hassasiyeti, kısa zaman döngüleri ve daha yüksek başarı garantisi ile tanımlanmaktadır. Ayrıca bu yöntemler, uzmanlık gereksinimi ve bakım kolaylığı açısından da daha olumlu puanlanmış görülmektedir. Çevik modellerin zaman, maliyet ve geri bildirim açısından daha verimli çalıştığı çıkarımı yapılabilir. Özellikle yazılım geliştirme, mobil uygulamalar, start-up projeleri ve müşteri ihtiyaçlarının hızla değiştiği dijital ürünlerde çevik yöntemlerin etkinliği daha yüksektir. Örneğin, bankacılık sektöründe geliştirilen kullanıcı arayüzü uygulamaları, e-ticaret platformları ya da mobil finans çözümleri gibi sürekli güncellenen ve müşteri geri bildirimlerine göre şekillenen projelerde çevik yaklaşımlar daha yaygın olarak kullanılmaktadır.

RUP ve Evrimsel Geliştirme (Evolutionary Development) gibi yöntemler ise bazı kriterlerde çevik, bazılarında geleneksel özellikler göstererek hibrit bir yapı sergilemektedir. Özellikle uygulanabilirlik ve esneklik açısından bu yöntemlerin daha dengeli bir model sunduğu görülmektedir. RUP, örneğin iteratif bir yapıya sahip olması sayesinde değişen gereksinimlere kısmen cevap verebilse de süreçlerin belirgin aşamalarla tanımlı olması nedeniyle planlama ve dokümantasyon açısından geleneksel modellere yakın durur. Bu nedenle hem değişken kullanıcı taleplerine hem de düzenleyici çerçeveye uyum sağlaması gereken projelerde örneğin kamu kurumlarına yönelik büyük çaplı yazılım sistemleri, kurumsal ERP çözümleri ya da sigortacılık sistemleri RUP gibi hibrit yöntemler daha sık tercih edilmektedir. Benzer şekilde, Evrimsel Geliştirme modeli de sistemin işlevsel versiyonlarının sürekli geliştirilmesiyle ilerlediğinden çevik metodolojilere benzer yönler taşır. Ancak, ilk yapı taşlarının belirli bir plan çerçevesinde oluşturulması ve sistemin evrimsel olarak şekillendirilmesi sebebiyle, bu yöntem daha kontrollü ve geleneksel bir temel sunar. Bu yaklaşım; örneğin medikal yazılımlar, gömülü sistemler veya otomotiv sektörü gibi hem teknik karmaşıklığın hem de değişken kullanıcı ihtiyaçlarının söz konusu olduğu alanlarda tercih edilmektedir. Bu tür projelerde hem kalite güvencesi hem de yinelemeli gelişim süreci önem taşıdığı için hibrit modeller uygun bir denge sağlar.

Sonuç olarak, tablo; her metodolojinin belirli bağlamlara göre avantaj ve sınırlılık taşıdığını, “en iyi” yöntemden ziyade “duruma en uygun” yöntemin seçilmesi gerektiğini göstermektedir.

Büyük, orta ve küçük ölçekli projelerde, proje yönetiminde Çevik ve geleneksel yaklaşımların kullanım oranları aşağıdaki tabloda sunulmuştur (Standish Group, 2015; Çalışkan vd., 2021).



Tablo 3

Proje Başarı Ölçütlerinin Karşılaştırılması

	Proje Boyutu	Yaklaşım	D.Çalışkan (Çalışkan vd., 2021)	Chaos Raporu (Standish Group, 2015)
Başarılı (%)	Büyük Projeler	Çevik (Agile) Yaklaşım	%58	%18
		Geleneksel Yaklaşım	%42	%3
	Orta Projeler	Çevik (Agile) Yaklaşım	%77	%27
		Geleneksel Yaklaşım	%62	%7
	Küçük Projeler	Çevik (Agile) Yaklaşım	%86	%58
		Geleneksel Yaklaşım	%77	%44
Başarısız (%)	Büyük Projeler	Çevik (Agile) Yaklaşım	%5	%23
		Geleneksel Yaklaşım	%11	%42
	Orta Projeler	Çevik (Agile) Yaklaşım	-	%11
		Geleneksel Yaklaşım	%6	%25
	Küçük Projeler	Çevik (Agile) Yaklaşım	%1	%4
		Geleneksel Yaklaşım	%5	%11

Kaynak: Yazarlar tarafından oluşturulmuştur.

Tablo 3, iki farklı kaynaktan elde edilen verilerle çevik ve geleneksel proje yönetimi yaklaşımlarının proje başarı oranlarını karşılaştırmaktadır. D. Çalışkan ve CHAOS Report verileri her ne kadar farklı örneklemelerden geliyor olsa da benzer bir eğilimi yansıtmaktadır: Çevik yaklaşımlar, tüm proje ölçeklerinde geleneksel yöntemlere göre daha yüksek başarı oranlarına sahiptir.

D. Çalışkan verilerine göre, özellikle küçük projelerde çevik yöntemlerin başarı oranı %86 ile oldukça yüksektir. Orta ve büyük projelerde bu oran sırasıyla %77 ve %58'dir. Aynı veri setinde geleneksel yöntemlerin başarı oranları daha düşük olmakla birlikte, küçük projelerde dahi çevik yaklaşımın gerisindedir.

CHAOS Repor verileri daha çarpıcıdır: büyük projelerde geleneksel yöntemlerin başarı oranı sadece %3 iken, çevik yöntemlerle bu oran %18'e çıkmaktadır. Orta ve küçük projelerde de benzer şekilde çevik yaklaşımlar gelenekselin çok üzerindedir.

Başarısızlık oranları incelendiğinde, geleneksel yöntemler özellikle büyük projelerde %42 gibi yüksek başarısızlık oranlarına sahiptir. Buna karşılık çevik yaklaşımların



başarısızlık oranları oldukça düşüktür. Bu durum, çevik yaklaşımın esnek yapısının, özellikle belirsizlik ve değişkenlik içeren projelerde daha dirençli olduğunu göstermektedir.

Tablodaki iki farklı veri seti, farklı örneklem ve yöntemlerle elde edilmiş olsa da ortak bir sonucu desteklemektedir: Çevik yöntemler, proje başarısını artırma potansiyeline daha fazla sahiptir ve bu durum proje ölçeğinden bağımsız şekilde gözlemlenmektedir.

2.1 Çevik Yaklaşımların Sınırlılıkları ve Uygunluk Koşulları

Çevik yaklaşımlar, değişimin yüksek olduğu ve paydaş geri bildiriminin hızlı alınabildiği ortamlarda önemli avantajlar sunsa da her projeye ve kurumsal bağlama “evrensel çözüm” olarak da görülmemelidir. Boehm ve Turner (2003), çevik ile plan odaklı yöntemler arasında seçim yaparken proje kritikliği, büyüklük, ekip yetkinliği, kurumsal kültür ve gereksinim belirsizliği gibi beş temel boyut önererek “tek boyutlu” çevik savunularının risklerine dikkat çekmektedir. Ölçek büyüdükçe, güvenlik ve uyumluluk gibi gereksinimler ağır bastıkça ve sözleşmesel belirlilik gereksinimi arttıkça, saf çevik yaklaşımlar yerine daha fazla dokümantasyon, formal karar süreçleri ve faz kapıları içeren modellerin gerekebildiği görülmektedir (Boehm & Turner, 2003; Fitzgerald & Stol, 2017). Ayrıca ampirik çalışmalar, çevik ekiplerin deneyimsiz olduğu, üst yönetimin desteğinin zayıf kaldığı ve organizasyonel yapının hiyerarşik olduğu bağlamlarda çevik dönüşümlerin beklenen performans kazanımlarını sağlayamadığını ortaya koymaktadır (Dikert vd., 2016).

2.2 Geleneksel Yaklaşımların Tercih Edilmesi Gereken Durumlar ve Hibrit Modeller

Literatürde, özellikle yüksek düzenleme ve güvenlik gerektiren savunma, sağlık ve havacılık gibi sektörlerde, ayrıntılı ön tasarım, kapsamlı dokümantasyon ve resmi doğrulama adımlarını içeren plan odaklı yaklaşımların hâlen önemli bir rol oynadığı vurgulanmaktadır (Williams, 2005; Hayes vd., 2016). Bu bağlamlarda, gereksinimlerin büyük ölçüde baştan belirlenebilir olduğu ve değişiklik maliyetinin yüksek kaldığı projelerde, geleneksel “şelale” veya V-model temelli süreçler, risk yönetimi ve düzenleyici uyum açısından avantaj sağlayabilmektedir (Fairley, 2019). Öte yandan son yıllarda, çevik ve geleneksel ilkeleri bilinçli biçimde birleştiren hibrit modeller öne çıkmaktadır. Örneğin “Water-Scrum-Fall” yaklaşımı, üst seviyede faz tabanlı bir yönetim yapısını korurken, uygulama geliştirme faaliyetlerini Scrum benzeri çevik sprint’lerle yürütmeyi önermektedir (West, 2011). Benzer şekilde, hibrit ölçekli çevik yaklaşımları inceleyen sistematik derlemeler, kurumsal bağlamda çevikliğin çoğu zaman portföy ve program seviyesinde plan odaklı yönetim mekanizmalarıyla birlikte uygulandığını göstermektedir (Moe & Šmite, 2022; Buchmann vd., 2023).

3. Çevik Proje Yönetim Metodolojileri

Çevik yaklaşımlar, özellikle değişken gereksinimlerin söz konusu olduğu koşullarda tercih edilmektedir. Bu bölümde en yaygın kullanılan iki çevik metodoloji olan Scrum ve Kanban detaylandırılacaktır.

3.1 Scrum

Scrum, ilk kez 1980’li yıllarda Jeff Sutherland tarafından geliştirilen ve daha sonra Ken Schwaber tarafından olgunlaştırılan bir Çevik proje yönetimi çerçevesidir (Schwaber, 1995). Sutherland ve Schwaber, Microsoft, Borland ve Hewlett-Packard gibi kuruluşlardaki



metodolojik deneyimlerinden yola çıkarak, endüstriyel kontrol teknikleri ile proje yönetim uygulamalarını birleştirerek sistematik ve etkili bir çerçeve oluşturmuşlardır.

"Scrum" terimi, Rugby sporundan alınmıştır; burada oyuncular kısa bir süre için bir araya gelerek bir sonraki hamlelerini planlarlar. Bu durum, Scrum metodolojisinin merkezinde yer alan kısa ve odaklı toplantılara benzemektedir (Anaral, 2012). Scrum, geleneksel hiyerarşik yapılardan uzaklaşarak, ekip çalışmasını ve ortak sorumluluğu vurgulamaktadır (Jacobson vd., 1999).

Scrum, çalışmalarını genellikle 2 ila 4 hafta süren iteratif döngülere, yani sprintlere ayırır. Her sprintin temel amacı, müşterilere veya paydaşlara sunulabilecek işlevsel, yüksek kaliteli bir ürün artımı teslim etmektir (Yılmaz, 2018).

3.1.1 Scrum'daki Temel Kavramlar

Scrum metodolojisinin dört temel kavramı bulunmaktadır (Anaral, 2012).

1. Ürün Listesi (Product Backlog): Projedeki tüm özellikleri ve gereksinimleri içeren önceliklendirilmiş listedir.
2. Sprint Listesi (Sprint Backlog): Belirli bir sprint süresince tamamlanacak görevlerin detaylı listesidir.
3. Engel Listesi (Impediment Backlog): İlerlemenin önünde engel oluşturan durumların kaydedildiği listedir.
4. İş Tamamlama Grafiği (Burndown Chart): Sprint süresince kalan işlerin takibini sağlayan görsel araçtır.

3.1.2 Scrum'daki Temel Toplantılar

Scrum dört temel toplantı içermektedir (Anaral, 2012; Yılmaz, 2018):

1. Sprint Planlama Toplantısı: Sprint süresince yapılacak işlerin belirlendiği toplantıdır.
2. Sprint Gözden Geçirme Toplantısı: Tamamlanan işlerin değerlendirildiği ve gözden geçirildiği toplantıdır.
3. Sprint Retrospektif Toplantısı: Ekip süreçlerini analiz ederek iyileştirme alanlarının belirlendiği toplantıdır.
4. Günlük Scrum Toplantısı: Ekip üyelerinin günlük olarak ilerlemelerini ve karşılaştıkları zorlukları paylaştıkları kısa toplantıdır.

3.1.3 Scrum'daki Roller

Scrum, üç temel rol tanımlamaktadır (Anaral, 2012; Yılmaz, 2018):

1. Ürün Sahibi (Product Owner): Görevleri önceliklendirerek ve müşteri ihtiyaçlarına uygun hale getirerek ürün değerini en üst düzeye çıkarmaktan sorumludur.
2. Scrum Master: Scrum sürecinin doğru bir şekilde uygulanmasını sağlamak ve ekibin karşılaştığı engelleri ortadan kaldırmakta rol oynar.



3. Geliştirme Ekibi (Development Team): Sprint hedeflerine ulaşmaya adanmış, farklı uzmanlıklara sahip profesyonellerden oluşan ekip.

3.1.4 Scrum'daki Temel Araçlar

Scrum'da iki ana araç kullanılmaktadır:

1. Ürün Listesi (Product Backlog): Proje için planlanan tüm özellikleri ve gereksinimleri içeren kapsamlı listedir.
2. Sprint Listesi (Sprint Backlog): Belirli bir sprint içinde tamamlanması hedeflenen görevlerin seçilmiş listesidir.

Çalışmaların yönetilebilir iterasyonlara bölünmesi sayesinde, scrum ekipleri değişen gereksinimlere uyum sağlayabilir, yüksek düzeyde iş birliği sürdürebilir ve proje yaşam döngüsü boyunca kademeli değer sunabilir. Şekil 6'da, Scrum sürecinde görevlerin sprint süresince nasıl yönetildiğini gösteren bir örnek görev panosu (Scrum Task Board) yer almaktadır. Bu pano, her bir kullanıcı hikâyesine ait görevlerin “Backlog”, “Yapılacak”, “Devam Eden”, “Test” ve “Tamamlandı” gibi aşamalarda nasıl takip edildiğini görsel olarak sunmaktadır. Bu da scrum yönteminin şeffaflık, izlenebilirlik ve görev takibi açısından sağladığı yapısal düzeni ortaya koymaktadır.

Şekil 6

Scrum Görev Listesi Örneği

BACKLOG	YAPILACAK	DEVAM EDEN	TAMAMLANDI	
Geliştirme	Analiz	Analiz	Geliştirme	Test
Geliştirme	Geliştirme	Geliştirme	Test	Analiz
	Test	Test		

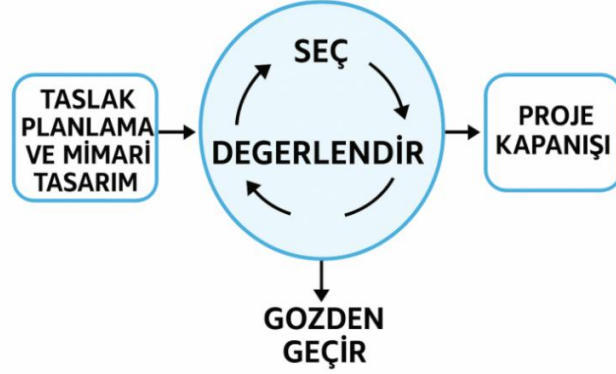
Kaynak: Yazarlar tarafından oluşturulmuştur.

Scrum süreci, genel proje planlamasından başlayarak sprint döngüleri aracılığıyla ilerleyen ve proje kapanışıyla sonuçlanan dört temel aşamadan oluşur: planlama, seçim/geliştirme, değerlendirme ve kapanış (Golden & Wasil, 1989). Şekil 7'de bu süreç hem sprint döngüsünün içsel işleyişi hem de genel proje akışı açısından şematik olarak sunulmaktadır. Süreç, "Taslak Planlama ve Mimari Tasarım" ile başlar, ardından “Seç” ve “Değerlendir” adımlarından oluşan döngüsel geliştirme süreciyle ilerler. Her sprint sonunda yapılan “Gözden Geçirme” ile çıktı değerlendirilir ve süreç, proje hedefleri tamamlandığında “Proje Kapanışı” ile sonlanır. Bu yapı, Scrum'ın sürekli iyileştirme ve esneklik odaklı yaklaşımını simgeler.



Şekil 7

Scrum Akışları

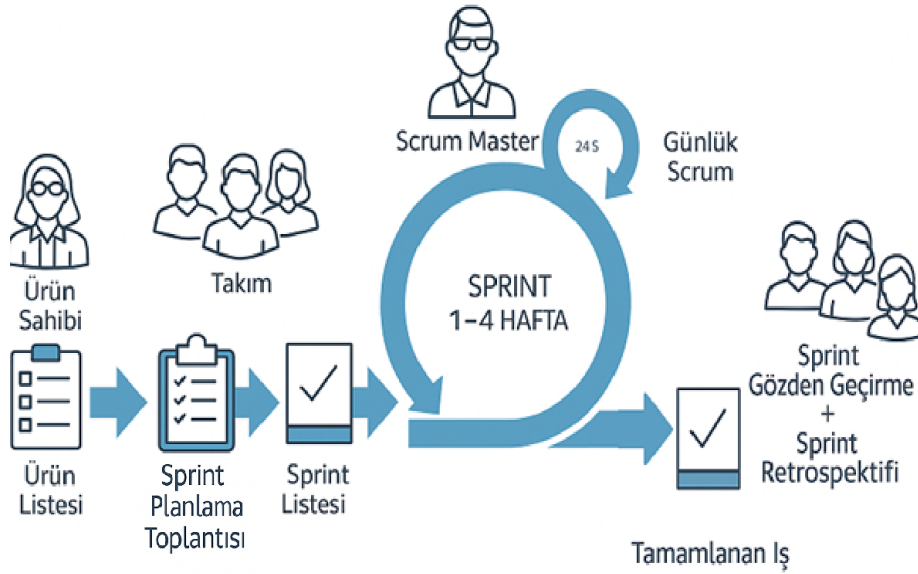


Kaynak: Yazarlar tarafından oluşturulmuştur.

Şekil 8’de scrum proje yönetim süreçleri gösterilmiştir. Scrum çerçevesinde, Ürün Sahibi (Product Owner), projenin başarılı bir şekilde tamamlanması için gerekli tüm görev ve gereksinimleri tanımlar ve bunları Ürün Listesi (Product Backlog) adı verilen belgede toplar. Sonrasında Scrum Ekibi bu listeyi detaylı şekilde inceleyerek görevleri daha küçük, net ve yönetilebilir parçalara ayırır. Bu parçalar, sprint süresince tamamlanacak görevleri içeren Sprint Listesi (Sprint Backlog) olarak düzenlenir.

Şekil 8

Scrum Proje Yönetim Süreci



Kaynak: Yazarlar tarafından oluşturulmuştur.

Sprintler, genellikle 1 ila 4 hafta süren sabit süreli çalışma döngüleridir. Bu süre zarfında ekip üyeleri her gün kısa ve odaklı bir Günlük Scrum Toplantısı düzenler. Bu toplantılarda her üye, tamamladığı işleri, karşılaştığı engelleri ve bir sonraki adımlarını paylaşır. Scrum Master bu toplantıların etkinliğini sağlar ve sürecin sorunsuz ilerlemesini destekler.



Sprint sonunda, Ürün Sahibi ve ilgili paydaşlar tarafından ortaya çıkan çıktılar gözden geçirilir ve değerlendirilir. Sprint çıktılarının müşteri teslimatına, satışa sunulmaya veya paydaşlara sunulmaya hazır kalitede olması beklenir.

Her sprint iki kritik aşama ile tamamlanır (Özdemir, 2015):

- Sprint Gözden Geçirme (Sprint Review): Sprint sonunda elde edilen çıktılar değerlendirilir ve gerekli geri bildirimler alınır.
- Sprint Retrospektifi (Sprint Retrospective): Ekip, karşılaşılan sorunları analiz eder ve bir sonraki sprint için daha etkili bir yol haritası oluşturur.

Bu yapı, scrum metodolojisinin sürekli iyileştirme ve uyarlanabilirlik odaklı yaklaşımını destekler ve projelerin daha verimli şekilde ilerlemesini sağlar (Özdemir, 2015).

3.2 Kanban

Kanban, Japonca'da "Tabela" veya "İşaret Tahtası" anlamına gelmektedir (Millweber, 2018). Bu metodolojinin temel amacı, belirli kartlar kullanılarak görevlerin bir çizelge üzerinde görsel şekilde organize edilmesini sağlamaktır. Kanban, 1940'lı yıllarda ortaya çıkmış ve 2004 yılından itibaren bilişim sektöründe geniş çapta kullanılmaya başlanmıştır. David J. Anderson, Microsoft'ta küçük bir ekibe destek olurken Kanban'ı yazılım geliştirme süreçlerine entegre etmiş ve bu yöntem hızla benimsenmiştir (Ahmad vd., 2013).

Kanban metodolojisi üç temel ilkeye dayanır (Başak, 2020):

- Görselleştirme: Proje veya görevlerin durumlarının görsel araçlar kullanılarak takip edilmesini sağlar ve ekip içi şeffaflığı artırır. Bu süreç fiziksel bir tahta veya dijital araçlar aracılığıyla uygulanabilir. İş akışı, tanımlı sütunlar ile yapılandırılır.
- Devam Eden İş (WIP) Limitleri: Çoklu görevlerin verimliliği düşürmesini önlemek için, aynı anda yürütülen işlerin sayısı sınırlanır. Bu, odaklanmayı artırır ve kaliteyi yükseltir.
- Akışı İyileştirme: Görevlerin sürekli ve sistematik bir akış içinde ilerlemesi sağlanır, böylece teslimat süreleri azaltılır ve süreç verimliliği artırılır. Bu ilke, darboğazların her aşamada tanımlanarak giderilmesini vurgular.

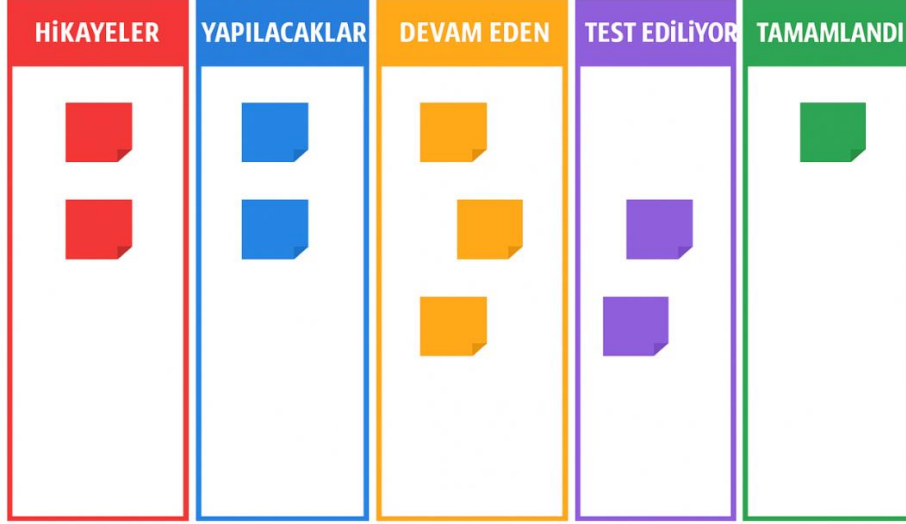
Sonuç olarak, görselleştirme, Kanban metodolojisinin en temel ve vazgeçilmez bileşenidir ve iş akışlarının şeffaf yönetimini mümkün kılar (Başak, 2020).

Şekil 9, çevik proje yönetiminde sıklıkla kullanılan Kanban panosuna ait örnek bir görselleştirme sunmaktadır (abc.com.tr, 2025). Panoda, işlerin hikâyelerden başlayarak yapılacaklar, devam eden, test edilen ve tamamlanan aşamalarına doğru ilerleyişi gösterilmektedir. Bu yapı, görevlerin görselleştirilmesi ve süreç yönetiminin şeffaf bir şekilde yürütülmesine katkı sağlamaktadır.



Şekil 9

Kanban Panosu Örneği



Kaynak: Yazarlar tarafından oluşturulmuştur.

Güncel yazılım endüstrisinde çevik yöntemler çoğu zaman DevOps ve sürekli teslim boru hatlarıyla birlikte ele alınmaktadır. Agile, bulut ve DevOps entegrasyonuna odaklanan sistematik derlemeler, otomasyon ve sürekli entegrasyon uygulamalarının, çevik değer akışının son aşamalarında teslim süresini kısalttığını ve geri bildirim döngülerini hızlandırdığını göstermektedir (Aouni vd., 2024; Jabbari vd., 2016). Bu nedenle, özellikle büyük ölçekli yazılım projelerinde Scrum veya Kanban uygulamalarının, altyapı ve operasyon ekiplerini de içeren DevOps pratikleriyle bütünleşik olarak tasarlanması kurumsal çevikliğin önemli bir bileşeni hâline gelmiştir.

4. Scrum, Kanban ve Şelale Proje Yönetim Yaklaşımlarının Karşılaştırılması

Çevik proje yönetim yaklaşımlarından olan Scrum ve Kanban modelleri, geleneksel proje yönetim yöntemi olan Şelale (Waterfall) modeli ile çeşitli parametreler üzerinden karşılaştırılmış ve bu karşılaştırma aşağıdaki tabloda sunulmuştur.

Tablo 4

Scrum-Kanban-Waterfall Proje Yönetim Yöntemlerinin Karşılaştırılması



Özellik	Kanban	Scrum	Şelale (Waterfall)
Roller ve Sorumluluklar	Takım için önceden belirlenmiş roller yoktur. Bir proje yöneticisi mevcut olabilir, ancak her takım üyesi gerektiğinde iş akışına katkıda bulunmaya teşvik edilir (Schwaber & Sutherland, 2017).	Her takım üyesinin önceden belirlenmiş bir rolü vardır. Ürün Sahibi hedefleri belirler, Scrum Master zaman çizelgesini oluşturur ve takımın iş akışını düzenler (Schwaber & Sutherland, 2017).	Roller ve sorumluluklar baştan kesin olarak belirlenmiştir ve değiştirilemez (Schwaber & Sutherland, 2017).
Süreler ve Zaman Çizelgeleri	Ürünler ihtiyaç doğrultusunda teslim edilir (bitiş tarihleri operasyonel ihtiyaçlara göre ayarlanır) (Schwaber & Sutherland, 2017).	Teslimatlar, belirlenen zaman dilimi içerisinde hazır olmalı ve tamamlanmalıdır (Schwaber & Sutherland, 2017).	Süreçler sıralı şekilde birbirini takip eder. Bir aşama tamamlandıktan sonra geri dönülemez (Schwaber & Sutherland, 2017).
Güçlendirme ve Önceliklendirme	"Çekme sistemi" kullanılır; ekip üyeleri önceki işlerini tamamlamadan yeni görev almaz. Böylece sistematik bir iş akışı sağlanır (Anderson, 2010).	Benzer şekilde bir "çekme sistemi" kullanılır, ancak görevler tüm ekip tarafından kolektif olarak üstlenilebilir (Anderson, 2010).	Görev akışı mevcut plana dayalıdır (Anderson, 2010).
Değişiklikler	Proje tamamlanmadan önce değişikliklere, yinelenmelere ve sürekli iyileştirmelere izin verilir (Anderson, 2010).	Sprint süresince değişiklik yapılması genellikle teşvik edilmez (Anderson, 2010).	Değişiklikler kabul edilmez; müşteri ürünü yalnızca tamamlandığında gözden geçirir (Anderson, 2010).
Verimlilik Ölçümü	"Çevrim süresi" ile ölçülür; bir görevin başlangıçtan tamamlanmaya kadar geçen süresi hesaplanır (Özdemir, 2020).	Sprintler aracılığıyla hız ve çıktı ölçülür. Her sprint ardışık veya paralel şekilde planlanır ve küçük artımlarla başarı sağlanır (Özdemir, 2020).	Planlama aşamasında belirlenen zaman, kaynak ve maliyet sınırlamalarına bağlılık üzerinden ölçülür (Özdemir, 2020).
En İyi Uygulama Alanı	Çok sayıda önceliği olan, değişken ihtiyaçlara sahip projeler için en uygundur (Özdemir, 2020).	Belirli zaman dilimlerinde görev odaklı çalışabilen ekipler için uygundur (Özdemir, 2020).	Kısa vadeli, ayrıntılı işler ve sınırlı kaynaklarla yürütülen projeler için uygundur (Özdemir, 2020).
Zaman Çerçevesi	Sürekli akışa odaklanır, belirli bir zaman sınırı yoktur (Ceylan & Gürsev, 2020).	Belirli zaman aralıklarında (Sprintler) çalışır (Ceylan & Gürsev, 2020).	Waterfall modeli ardışık (sequential) bir yapıya sahiptir. Her aşama sırasıyla gerçekleştirilir ve bir sonraki adıma geçmeden önce mevcut aşamanın tamamlanması gerekir. Bu, sabit zaman çizelgesi sağlar (Haralayya, 2024).
Roller	Belirli roller yoktur; herkes iş akışına katkıda bulunur (Ceylan & Gürsev, 2020).	Scrum Master, Ürün Sahibi, Scrum Ekibi vardır (Ceylan & Gürsev, 2020).	Waterfall'da roller önceden belirlenmiştir: proje yöneticisi, analist, geliştirici, test uzmanı gibi roller süreç boyunca değişmeden kalır (Saravanas & Curinga, 2023).
Yönetim	Görsel bir pano üzerinden sürekli izlenir ve optimize edilir (Anderson, 2010).	Sprintler ile planlanır ve yönetilir (Anderson, 2010).	Proje yönetimi, kapsamlı bir başlangıç planlamasına dayanır. Yönetim aşamaları arasında net sınırlar koyar ve değişiklik yapılmasını zorlaştırır (Sirisha vd., 2024).
Toplantılar	Toplantılar gerekli olduğunda yapılır (Schwaber & Sutherland, 2017).	Günlük toplantı, sprint gözden geçirme, retrospektif yapılır (Schwaber & Sutherland, 2017).	Toplantılar genellikle proje başında ve aşama geçişlerinde yapılır. Günlük ya da çevik yöntemlerdeki gibi sürekli değildir (Bassil, 2012).
Esneklik	Yüksek esneklik; işler sürekli yeniden düzenlenebilir (Ceylan & Gürsev, 2020).	Orta düzeyde; sprintler sabittir (Ceylan & Gürsev, 2020).	Esneklik düşüktür. Proje ilerledikçe geri dönüş maliyeti artar. Bu yüzden değişiklikler genellikle kabul edilmez (Saravanas & Curinga, 2023).
Teslimat	İş tamamlandığında anlık teslimatlar yapılabilir (Özdemir, 2020).	Sprint sonunda yapılır (Özdemir, 2020).	Teslimat, tüm proje aşamaları tamamlandıktan sonra gerçekleşir. Yani tek seferde toplu teslim yapılır (Hygger.io, 2025).

Tablo 4, Scrum, Kanban ve Şelale metodolojilerini on iki farklı kriter üzerinden karşılaştırarak her yaklaşımın uygulama karakteristiğini net bir biçimde ortaya koymaktadır. Scrum ve Kanban, birçok başlıkta benzer özelliklere sahipken, Şelale yönetimi bu iki çevik yönetime kıyasla daha yapılandırılmış ve katı bir model olarak öne çıkmaktadır.

Özellikle esneklik, teslimat süreci ve değişiklik yönetimi gibi başlıklarda, Kanban ve Scrum'ın daha dinamik ve uyarlanabilir yapılar sunduğu görülmektedir. Her iki çevik yaklaşım, sık teslimat döngüleri ve değişime açık iş akışları ile çevresel değişkenliklere daha hızlı yanıt verebilme özelliğine sahiptir.



Buna karşın Şelale yöntemi, rollerin ve süreçlerin net şekilde tanımlandığı, aşamalar arasında geçişin katı kurallara bağlı olduğu bir yapıya sahiptir. Esneklik düzeyinin düşük olması ve değişikliklerin süreç sonunda değerlendirilmesi, bu yöntemi daha çok öngörülebilir ve sabit gereksinimli projelere uygun hale getirmektedir.

Toplantı yapısı açısından; Scrum belirli ritüelleri içeren daha sistematik bir yaklaşım sergilerken, Kanban ihtiyaçlara bağlı olarak toplantı düzenlemektedir. Şelale ise toplantıları yalnızca kritik geçiş noktalarında yoğunlaştırmakta, düzenli iletişimden ziyade plan takibini esas almaktadır.

Verimlilik ölçümünde; Scrum hız odaklı metrikler kullanırken, Kanban çevrim süresini esas almakta, Şelale ise plana sadakati ölçüt olarak ilerlemektedir. Bu da her yöntemin başarıyı farklı şekillerde tanımladığını göstermektedir.

5. Çevik Proje Yönetim Yazılım ve Teknolojileri

5.1 Jira

Şirketler tarafından proje ve süreç yönetimi amacıyla tercih edilen pek çok uygulama arasında, Jira öne çıkan lider bir araçtır. Kullanıcı dostu arayüzü, yüksek performansı ve Çevik proje yönetim metodolojilerine uyumluluğu sayesinde, Jira yazılım geliştirme süreçlerinde vazgeçilmez bir iş takibi aracı haline gelmiştir. Ancak Jira'nın kullanımı yalnızca yazılım sektörüyle sınırlı değildir; farklı sektörlerde de etkin bir şekilde kullanılmaktadır. Jira, Çevik yazılım projelerinin planlanması ve yönetilmesi için esnek bir altyapı sunmakta ve Scrum metodolojisinin tüm özelliklerinin kusursuz ve verimli bir şekilde uygulanmasını sağlamaktadır (Borandag & Yücalar, 2020).

5.2. Trello

Trello, yazılım projelerindeki görevlerin planlanması, izlenmesi ve yönetilmesi için hem web hem de mobil cihazlar üzerinden kullanılabilen bir proje yönetim aracıdır (Writtenhouse, 2016). Bu araç sayesinde bir projenin görevleri, tamamlanan işleri ve geleceğe yönelik hedefleri tek bir platform üzerinden organize edilebilir ve takip edilebilir. Ayrıca, Trello, proje süresince görevlerin belirli zaman dilimlerinde ekip üyelerine atanmasını sağlayarak ekip tabanlı projelerde daha şeffaf bir yönetim süreci oluşturulmasına yardımcı olmaktadır.

5.3. Asana

Asana, basit ve sezgisel kullanıcı arayüzü sayesinde kullanıcıların projeler ve görevler oluşturmasını ve bunları kolayca yönetmesini sağlayan bir proje yönetim aracıdır (Asana, 2025). Asana üzerinden görevler herhangi bir karmaşık form doldurmadan hızlıca eklenebilir ve düzenlenebilir. Ayrıca hangi ekip üyesinin hangi görev üzerinde çalıştığı net bir şekilde görülebilir ve görevler kolaylıkla ekip üyeleri arasında aktarılabilir. Bu özellikleriyle Asana, ekip içi iş birliğini destekleyen güçlü ve verimli bir araç olarak öne çıkmaktadır.

5.4. MS Project

MS Project, Microsoft tarafından geliştirilmiş ve birçok kurum tarafından tercih edilen masaüstü tabanlı bir proje yönetim yazılımıdır (Çandur, 2010). Bu araç, projelerin dijital



ortamda planlanmasını, organize edilmesini ve sunulmasını kolaylaştırmaktadır. MS Project kullanımının sunduğu bazı avantajlar şunlardır:

- Faaliyetlerin Tanımlanması ve Zaman Çizelgelerinin Görselleştirilmesi: Proje içindeki görevler, Gantt Diyagramı gibi zamana bağlı görsel araçlarla tanımlanabilir ve takip edilebilir.
- İş Kırılım Yapısı (WBS): Proje görevleri, proje yönetim ilkelerine uygun şekilde daha küçük ve yönetilebilir alt görevlere bölünebilir. Bu yapı, süreçlerin daha net ve verimli yönetilmesine katkı sağlar.
- Görevler Arası Bağlantı Kurulması: Görevlerin tamamlanma sırası ve birbirleriyle olan bağımlılıkları (öncelik ve ardılık ilişkileri) tanımlanarak iş akışı süreci net bir şekilde ortaya konabilir.

Farklı Çevik proje yönetim araçlarının çeşitli parametreler açısından karşılaştırması Tablo 5'te verilmiştir (Majstorovic & Majstorovic, 2020). Tablo 5, dört yaygın proje yönetim aracını temel fonksiyonel kriterler üzerinden değerlendirerek araçların güçlü yönlerini karşılaştırmalı olarak sunmaktadır. İnceleme sonucunda, Jira ve Microsoft Project'in özellik kapsamı açısından daha gelişmiş, Trello ve Asana'nın ise daha sade ve kullanıcı dostu çözümler sunduğu anlaşılmaktadır.

Jira, kaynak yönetimi, proje planlama, risk analizi ve detaylı raporlama gibi birçok kurumsal özelliği desteklemesiyle öne çıkarken, Microsoft Project de benzer şekilde proje planlama ve çoklu proje yönetimi gibi özellikleriyle daha kapsamlı kontrol imkânı sunmaktadır.

Buna karşılık, Asana ve Trello, iş birliği araçları, ilerleme takibi ve e-posta bildirimleri gibi daha hafif, günlük kullanım odaklı fonksiyonlarla öne çıkmaktadır. Özellikle Asana'nın görev yönetimi ve kullanıcı deneyimi açısından geniş bir kullanım alanına hitap ettiği görülmektedir.

Trello'nun kaynak ve risk yönetimi gibi özellikleri desteklememesi, onu daha çok küçük ekipler ve bireysel kullanıcılar için uygun hale getirirken; ücretsiz kullanım avantajı sayesinde erişilebilirlik yönüyle pozitif ayrışmaktadır.

Tablo 5

Proje Yönetim Araçlarının Karşılaştırılması

Yönetim Özellikleri	Asana	Trello	Microsoft Project	Jira
Kaynak Yönetimi	x		x	x
Risk Yönetimi	x			x
Proje Zaman Takibi	x	x	x	x
İlerleme Takibi	x			x
Proje Planlama	x		x	x
Çoklu Proje Yönetimi	x	x	x	x
İş Birliği Araçları	x	x	x	x
Raporlama ve İstatistik	x		x	x
E-posta Bildirimi	x	x		x
Ücretsiz Kullanım		x		

Kaynak: Yazarlar tarafından oluşturulmuştur.



6. Tartışma ve Sonuç

Bu çalışmada elde edilen bulgular, çevik proje yönetimi yaklaşımlarının özellikle dinamik, belirsizlik düzeyi yüksek ve müşteri beklentilerinin hızlı değiştiği proje ortamlarında önemli avantajlar sunduğunu göstermektedir. Ancak sonuçlar, çevik yöntemlerin her bağlamda aynı etkiyi yaratmadığını; düzenleyici uyumluluk gerektiren ya da büyük ölçekli kurumsal yapılarda geleneksel veya hibrit yöntemlerin daha uygun olabileceğini ortaya koymaktadır. Bu durum, proje yönetim yaklaşımının teknik bir tercih değil, doğrudan stratejik bir karar olduğunu göstermekte ve yöntem seçiminin proje türü, sektör ihtiyaçları ve organizasyonel olgunluk düzeyi dikkate alınarak yapılması gerektiğine işaret etmektedir.

Tablo 2’de geleneksel yöntemlerin yüksek maliyet, uzun zaman çizelgesi ve düşük esneklik gibi sınırlamalara sahip olduğu görülmektedir. Buna karşılık çevik yöntemler, belirsizlik içeren projelerde daha uyarlanabilir bir yapı sunmaktadır. Bu bulgular, tek bir ‘en iyi’ yöntemin olmadığını göstermektedir. Bunun yerine, bağlama en uygun yöntemin seçilmesi daha anlamlıdır.

Tablo 3’te yer alan veriler, farklı araştırmalardan elde edilmiş olmasına rağmen ortak bir sonuca işaret etmektedir: Çevik yöntemler, proje başarısını artırma potansiyeline daha fazla sahiptir ve bu durum proje ölçeğinden bağımsız şekilde gözlemlenmektedir. Gerek D. Çalışkan verileri gerekse CHAOS Report çıktıları, büyük, orta ve küçük ölçekli projelerde çevik metodolojilerin geleneksel yöntemlere kıyasla daha yüksek başarı oranlarına sahip olduğunu ortaya koymuştur. Bu bulgu, Çevik metodolojilerin yalnızca küçük ölçekli projelere değil, aynı zamanda büyük çaplı, karmaşık yapılara da uyarlanabileceğini ve başarıyla uygulanabileceğini göstermektedir.

Tablo 5 ise, Scrum ve Kanban’ın çevik, sürekli gelişime açık yapılarıyla; Şelale yönetiminin kontrollü, sıralı ilerleyen yapısını açık biçimde karşılaştırmakta ve yöntem seçiminin proje koşullarına göre yapılması gerektiğini ortaya koymaktadır. Özellikle esneklik, değişikliklere açıklık, teslimat döngüsü ve iş birliği gibi başlıklarda Scrum ve Kanban’ın çok daha etkili olduğu görülmektedir. Buna karşılık şelale yönetimi, sabit gereksinimler ve düşük belirsizlik içeren projelerde hâlâ geçerliliğini korumaktadır. Bu tablo üzerinden, metodoloji seçiminin sadece teknik gerekliliklerle değil, aynı zamanda proje paydaşlarının beklentileri, organizasyonel kültür ve dış çevresel etkenlerle birlikte değerlendirilmesi gerektiği çıkarımı yapılabilir.

Son bölümde, yaygın olarak kullanılan Scrum ve Kanban gibi çevik proje yönetim metodolojileri incelenmiş; süreçleri ve sağladıkları avantajlar analiz edilmiştir. Ayrıca Jira, Trello, Asana ve MS Project gibi proje yönetim yazılımlarının çevik metodolojilerle entegrasyonu değerlendirilmiş ve bu araçların proje başarı oranlarına katkıları tartışılmıştır. Elde edilen bulgular, çevik yöntemlerin müşteri beklentilerine hızlı ve etkin şekilde yanıt verilmesini sağladığını ve proje süreçlerinde esneklik ile uyarlanabilirliği artırdığını göstermektedir. Geleneksel proje yönetim metodolojileri, net ve sabit hedeflere sahip projelerde etkili sonuçlar sunmaktadır. Ancak dinamik ve hızlı değişen ortamlarda aynı başarıyı gösterememektedir. Bu nedenle çevik yaklaşımlar; esneklik, müşteri odaklılık ve hızlı geri bildirim avantajlarıyla öne çıkmaktadır. Bununla birlikte, çevik metodolojiler her bağlam için ideal çözüm değildir.



Türkiye bağlamına bakıldığında, yazılım endüstrisinde çevik ve geleneksel yöntemlerin birlikte kullanıldığını ve kademeli bir çevikleşme eğiliminin bulunduğunu gösteren ampirik çalışmalar mevcuttur. Örneğin anket temelli bir çalışmada, Türkiye’de faaliyet gösteren yazılım ekiplerinde çevik yöntemlerin kullanım oranının geleneksel yaklaşımlarla başa baş olduğu, ancak zaman içinde çevik yaklaşımlara doğru belirgin bir yönelim gözlemlendiği; buna karşın katılımcıların çevik yöntemleri daha çok küçük ekipler ve düşük/orta ölçekli projeler için uygun gördükleri rapor edilmiştir (Yılmaz & arkadaşları, 2020). Daha yeni bir çalışma ise Türkiye’de çevik süreçlerle çalışan ekiplerin başarı ve başarısızlık faktörlerini Analitik Hiyerarşi Süreci ile incelemiş; üst yönetim desteği, ekip içi iletişim ve müşteri katılımını kritik başarı faktörleri olarak belirlemiştir (Yılmaz vd., 2025). Bu bulgular, bu çalışmada tartışılan kavramsal çerçevenin Türkiye yazılım endüstrisindeki pratiklerle de uyumlu olduğunu, ancak özellikle büyük ve düzenlemeye tabi projelerde hibrit ve bağlama duyarlı çözüm arayışlarının sürdürüğünü göstermektedir.

Sonuç olarak, çevik proje yönetimi metodolojileri yalnızca operasyonel süreçlerde verimlilik sağlamakla sınırlı değildir; aynı zamanda işletmelerin rekabet gücünü artıran stratejik bir araç hâline gelmiştir. Çalışmanın bulguları, çevik yaklaşımların karar alma süreçlerini hızlandığını, müşteri değerini sürekli kıldığını ve piyasa değişikliklerine uyumu geliştirdiğini göstermektedir. Bununla birlikte işletmeler için sadece doğru aracı seçmek yeterli değildir; ekiplerin çevik yetkinliklerinin güçlendirilmesi, üst yönetimin dönüşümü sahiplenmesi ve organizasyon kültürünün çevik ilkelerle uyumlu hâle getirilmesi gerekmektedir. Türkiye yazılım sektöründeki eğilimler dikkate alındığında işletmelerin hibrit yaklaşımları, ölçekli çevik çerçeveleri (ör. SAFe 6.0) ve yapay zekâ destekli geliştirici araçları değerlendirmesi stratejik avantaj sağlayacaktır. Bu çerçevede, proje yönetimi yaklaşımı işletmeler için teknik bir süreç değil, rekabet üstünlüğü yaratan bir stratejik yönetim bileşenidir.

6.1. Kısıtlar ve Gelecek Çalışmalar

Bu çalışmanın bazı kısıtları bulunmaktadır. Çalışma ağırlıklı olarak ikincil verilere ve literatür taramasına dayandığı için, elde edilen sonuçların farklı sektörler veya organizasyon yapılarına doğrudan genellenmesi sınırlıdır. Ayrıca kullanılan metodolojik çerçeve yazılım geliştirme bağlamına odaklandığından; donanım, siber-fiziksel sistemler veya yüksek düzenleme gerektiren alanlardaki çevik uygulamalar bu kapsam dışında kalmaktadır. Gelecek çalışmalarda Türkiye’deki işletmelerin çevik, geleneksel ve hibrit proje yöntemlerini nasıl uyguladıklarını inceleyen saha araştırmaları yapılması; proje başarısına etki eden faktörlerin (zaman, maliyet, kapsam, kalite, müşteri memnuniyeti vb.) daha detaylı analiz edilmesi ve özellikle büyük ölçekli kurumlarda çevik dönüşümün kritik başarı bileşenlerinin belirlenmesi önerilmektedir. Bunun yanında, yapay zekâ temelli geliştirme araçlarının (örn. GitHub Copilot, Tabnine) proje yönetim süreçlerine nasıl entegre edilebileceğinin araştırılması gelecekteki çalışmalara önemli katkı sağlayacaktır.

Kaynakça

- Agile Alliance. (t.y.). Manifesto for Agile Software Development. Erişim tarihi 2 Haziran 2025, <https://agilemanifesto.org/>
- Agile Alliance. (t.y.). Principles behind the Agile Manifesto. Erişim tarihi 4 Temmuz 2025, <https://agilemanifesto.org/principles.html>
- Agile Digest. (t.y.). Scrum board. Erişim tarihi 20 Haziran 2025, <https://agiledigest.com/scrum-board/>



- Aguanno, K. (2004). *Managing Agile Projects*. Multi-Media Publications.
- Ahmad, O. M., Markkula, J., & Ovio, M. (2013). Kanban in software development: A systematic literature review. *İçinde 39th Euromicro Conference Series on Software Engineering and Advanced Applications (SEAA)*.
- Ahrend, J. (2023). *SAFe 6.0 im Unternehmen implementieren: Lean und Agile erfolgreich skalieren*. mitp.
- Akça, H. K. (2022). *Bilgi teknolojilerinde proje yönetim metodolojilerinin bilişsel ergonomik açıdan değerlendirilmesi* [yüksek lisans tezi]. Sakarya Üniversitesi, Fen Bilimleri Enstitüsü.
- Akdağ, A. (2020). *Nasıl Agile olunur? Cesur çevik organizasyon pratikleri*. Seçkin Yayıncılık.
- Andersen, E. S. (2006). Perspectives on projects. *İçinde Proceedings of the PMI Research Conference*.
- Anderson, D. J. (2010). *Kanban: Successful evolutionary change for your technology business*. Blue Hole Press.
- Anaral, F. (2012). *Çok kriterli karar verme yöntemi ile yazılım geliştirme metodolojisi seçimi* [doktora tezi]. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
- Anonim. (1968). *Yatırım ve proje tanımı (DPT No: 741)*. Devlet Planlama Teşkilatı.
- Anonim. (2025). Proje yönetimi. *Vikipedi*. Erişim tarihi 17 Temmuz 2025, https://tr.wikipedia.org/wiki/Proje_y%C3%B6netimi
- Aouni, M. vd. (2024). A systematic literature review on Agile, Cloud, and DevOps integration. *Information and Software Technology*, 169, 107569. CoLab
- El Aouni, F., Moumane, K., Idri, A., Najib, M., & Jan, S. U. (2024). A systematic literature review on Agile, Cloud, and DevOps integration. *Information and Software Technology*, 169, 107569.
- Asana. (t.y.). Asana. Erişim tarihi 14 Haziran 2025, <https://www.asana.com>
- Ågren, P., Knoph, E., & Berntsson Svensson, R. (2022). Agile software development one year into the COVID-19 pandemic. *Empirical Software Engineering*, 27, 121.
- Ayaydın, E. (2000). *Bir yatırım projesi için uygulanan ve önerilen proje yönetim sistemi* [yüksek lisans tezi]. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
- Başak, D. (2020). *IT yazılım sektöründe Agile Scrum proje yönetimi uygulaması: Uluslararası bir proje örneği* [Yayımlanmamış yüksek lisans tezi]. Sosyal Bilimler Enstitüsü.
- Bassil, Y. (2012). A simulation model for the Waterfall software development life cycle, 1205.6904.
- Beck, K., vd. (2001). *Manifesto for Agile Software Development*. Agile Alliance.
- Boehm, B. (2002). Get ready for agile methods, with care. *Computer*, 35(1), 64–69.
- Boehm, B., & Turner, R. (2003). *Balancing agility and discipline: A guide for the perplexed*. Addison-Wesley.
- Borandag, E., & Yücalar, F. (2020). Artırılmış gerçeklik ile Scrum task board uygulaması. *Uluslararası Yönetim Bilişim Sistemleri ve Bilgisayar Bilimleri Dergisi*, 4(1), 1–12.
- Buchmann, M., Kuvaja, P., & Oivo, M. (2023). A systematic literature review on hybrid agile governance in large-scale organizations. *Journal of Systems and Software*, 196, 111–208.
- Ceylan, Ş., & Çığırkan, S. (2020). Scrum metodu ile proje yönetimi: Yazılım projelerinde çevik yaklaşımların kullanımı. *Uluslararası Yönetim ve Sosyal Araştırmalar Dergisi*, 7(13), 112–126.
- Chin, C. M. M., & Spowage, A. C. (2010). Defining & classifying project management methodologies. *PM World Today*, 12(5).
- Chin, G. (2004). *Agile project management: How to succeed in the face of changing project requirements*. AMACOM.
- Cicmil, S., Cooke-Davies, T., Crawford, L., & Richardson, K. (2009). *Exploring the complexity of projects: Implications of complexity theory for project management practice*. Project Management Institute.
- Cicmil, S., Williams, T., Thomas, J., & Hodgson, D. (2006). Rethinking project management: Researching the actuality of projects. *International Journal of Project Management*, 24(8), 675–686.
- Cleland, D. I. (1999). *Project management: Strategic design and implementation*. McGraw-Hill.
- Collyer, S., Warren, C., Hemsley, B., & Stevens, C. (2010). Aim, fire, aim – Project planning styles in dynamic environments. *Project Management Journal*, 41(4), 108–121.
- Çalışkan, D., Yavuz, A., Doğan, B., & Çalış, B. (2021). Türkiye’de çevik ve klasik yazılım geliştirme metodolojilerine dair kapsamlı bir değerlendirme. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 10(1), 149–159.
- Çandur, C. (2010). *Bankacılık IT proje yönetimi ve MS Project’le proje planlama* [Yayımlanmamış yüksek lisans tezi]. Fen Bilimleri Enstitüsü.
- DeCarlo, D. (2004). *eXtreme project management*. Jossey-Bass.
- Demirel, K. (2014). *Proje yönetimi el kitabı* [yüksek lisans tezi]. Kocaeli Üniversitesi, Sosyal Bilimler Enstitüsü.
- Denning, S. (2018). *The age of agile*. *Harvard Business Review*, 96(3), 14–20.
- Dhani, H. (2016). Comparative study and analysis of software process models on various merits. *International Journal of Advanced Research in Computer Science and Software Engineering*, 6(9).
- Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*, 119, 87–108.



- Dohmke, T., Iansiti, M., & Richards, G. (2023). *Sea change in software development*. arXiv.
- Duyguvar, A. S., Müh, E. Y., & Akıncı, G. (2014). Dinamik bir yaklaşım: Çevik proje yönetimi ve Ar-Ge projesi için değişime çevik uyum. İçinde *U.Y.M. PMP & Teknolojileri bildirileri*.
- Ece, E., & Kovancı, A. (2004). Proje yönetimi ve insan kaynakları ilişkisi. *Journal of Aeronautics and Space Technologies*, 1(4), 75–85.
- Fair, J. (2012). Agile versus Waterfall: Which approach is right for my ERP project? İçinde *PMI® Global Congress 2012—EMEA*. Marsilya, Fransa: Project Management Institute.
- Fairley, R. (2019). *Managing and Leading Software Projects*. Wiley.
- Fitzgerald, B., & Stol, K.-J. (2017). *Continuous software engineering*. JSS, 123, 176–189.
- Frank, E., & Godwin, O. (2024). *Enhancing developer productivity*. EasyChair.
- Gido, J., & Clements, J. P. (1999). *Successful project management*. South-Western College Publishing.
- Golden, B. L., Wasil, E. A., & Harker, P. T. (Eds.). (1989). *The analytic hierarchy process: Applications and studies*. Springer-Verlag.
- Greenlight Guru. (t.y.). Agile design & development for medical devices. Erişim tarihi 6 Ağustos 2025, <https://www.greenlight.guru/blog/agile-design-development-medical-devices>
- Haralayya, B. (2024). Project management methodologies: A comparative analysis of agile and Waterfall approaches.
- Hayes, R. H., Wheelwright, S. C., & Clark, K. B. (2016). *Dynamic Manufacturing*. Free Press.
- Highsmith, J. (2004). *Agile Project Management: Creating Innovative Products*. Addison-Wesley.
- Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, 34(9), 120–122.
- Hygger.io. (t.y.). Team roles in Waterfall methodology. Erişim tarihi 29 Mayıs 2025, <https://hygger.io/blog/team-roles-in-waterfall-methodology/>
- Jabbari, R., Ali, N. B., Petersen, K., & Tanveer, B. (2016). *What is DevOps? A systematic mapping study on definitions and practices*. Proceedings of XP2016, 1–11.
- Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The unified software development process*. Addison-Wesley Professional.
- Journal of Medical Internet Research. (2021). Erişim tarihi 21 Haziran 2025, <https://www.jmir.org/2021/2/e24691/>
- Kerzner, H. (2001). *Strategic planning for project management using project management maturity model*. John Wiley & Sons.
- Kerzner, H. (2017). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. Wiley.
- Kir, E. (2007). *Yazılım sektöründe proje yönetimi [yüksek lisans tezi]*. Kadir Has Üniversitesi, Sosyal Bilimler Enstitüsü.
- Kocakulak, M. (1997). *Proje yönetim danışmanlığı yapan firma bakışıyla proje yönetim sistemi ve uygulama örneği [yüksek lisans tezi]*. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
- Kömürlü, R., & Toltar, L. (2018). İnşaatta proje yönetimi; projenin başarısına etkisi. *Mimarlık ve Yaşam*, 3(2), 249–258.
- Leffingwell, D. (2007). *Scaling software agility: Best practices for large enterprises*. Addison-Wesley.
- Lewis, J. P. (1996). *Project planning, scheduling, and control*. McGraw-Hill.
- Majstorovic, A., & Majstorovic, V. (2020). Practical application of contemporary, *31st DAAAM International Symposium on Intelligent Manufacturing and Automation: Annals of DAAAM & Proceedings*.
- Millweber, F. (2018). *Kanban für Anfänger: Wie Kanban in der Praxis funktioniert*. Lucid Page Media.
- Milošević, D., & Patanakul, P. (2005). Standardized project management may increase development projects success. *International Journal of Project Management*, 23(3), 181–192.
- Moe, N. B., & Šmite, D. (2022). Understanding large-scale agile: A multiple-case study of agile transformation in large organizations. *Empirical Software Engineering*, 27, 1–34.
- Olsson, N. O. E. (2006). Management of flexibility in projects. *International Journal of Project Management*, 24(1), 66–74.
- Özbay, M., Koçak, G., & Özen, M. T. (2020). *Yönetim perspektifinden pandemi dönemi üzerine düşünceler*. Karahan Yayınevi.
- Özdemir, E. (2020). *Büyük ölçekli firmalarda proje yönetiminde çevik yaklaşımlar [yüksek lisans tezi]*. Işık Üniversitesi, Sosyal Bilimler Enstitüsü.
- Özdemir, M. (2015). Scrum Framework'üne giriş. *KOD5*. Erişim tarihi 25 Temmuz 2025, <http://kod5.org/scrum-metodolojisine-giris/>
- Palmquist, M. S., Lapham, M. A., Miller, S., Chick, T., & Ozkaya, I. (2013). *Parallel worlds: Agile and Waterfall—Differences and similarities* [Teknik rapor]. Software Engineering Institute, Carnegie Mellon University.
- Peng, S., Kalliamvakou, E., Cihon, P., & Demirel, M. (2023). *The impact of AI on developer productivity*. arXiv:2302.06590.
- Poppendieck, M., & Poppendieck, T. (2003). *Lean Software Development*. Addison-Wesley.



- Prakash, A., Maddulety, K., & Bhoola, V. (2022). Factors influencing adoption of agile project management in construction industry. *Mathematical Statistician and Engineering Applications*, 71(3s), 788–800.
- Project Management Institute. (2003). *A guide to the project management body of knowledge (PMBOK® Guide)*. Project Management Institute.
- Quickbase. (t.y.). Agile construction. Erişim tarihi 18 Temmuz 2025, <https://www.quickbase.com/blog/agile-construction>
- Saravanos, A., & Curinga, M. X. (2023). Simulating the software development lifecycle: The Waterfall model. *Applied System Innovation*, 6(4), 91.
- Saynisch, M. (2010). Beyond frontiers of traditional project management: An approach to evolutionary, self-organizational principles and the complexity theory—Results of the research program. *Project Management Journal*, 41(2), 21–37.
- Scaled Agile. (2023). *SAFe 6.0 Overview*. Erişim tarihi 17 Ekim 2025, <https://scaledagile.com>
- Schwaber, K. (1995, Ekim). The Scrum development process. İçinde *OOPSLA '95 Proceedings*. Austin, TX, ABD.
- Schwaber, K., & Sutherland, J. (2012). *Software in 30 days: How agile managers beat the odds, delight their customers, and leave competitors in the dust*. John Wiley & Sons.
- Schwaber, K., & Sutherland, J. (2017). *The Scrum Guide: The definitive guide to Scrum: The rules of the game*. Scrum.org.
- Shenhar, A. J. (1998). From theory to practice: Toward a typology of project-management styles. *IEEE Transactions on Engineering Management*, 45(1), 33–48.
- Shenhar, A. J., & Dvir, D. (2007). *Reinventing project management: The diamond approach to successful growth and innovation*. Harvard Business Press.
- Sirisha, V. S., John, E. P., Haralayya, B., RamaKrishna, B. V., & Bisen, D. (2024). Project management methodologies: A comparative analysis of Agile and Waterfall approaches. *Library Progress (International)*, 44(3), 17237–17246.
- Spiceworks. (t.y.). What is Agile Software Development? Life cycle, methodology, and examples. Erişim tarihi 11 Ocak 2025, <https://www.spiceworks.com/tech/devops/articles/what-is-agile-software-development/>
- Standish Group. (2015). *CHAOS Report 2015*. https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf
- Şpundak, M. (2014). Mixed agile/traditional project management methodology – Reality or illusion? *Procedia – Social and Behavioral Sciences*, 119, 939–948.
- Taylor & Francis. (2020). [Makale özeti]. Erişim tarihi 26 Temmuz 2025, <https://doi.org/10.1080/09537287.2020.1837931>
- Todorovic-Marinic, D., Gray, D., Zheng, Y., Larson, G., & Pelletier, J. (2003). Envelope of fracture density. İçinde *CSPG and CSEG Joint Convention bildirileri*.
- West, D. (2011). *Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today*. Forrester Research Report.
- Wells, H., Dalcher, D., & Smyth, H. (2015). The adoption of agile management practices in a traditional project environment: An IT/IS case study. İçinde *48th Hawaii International Conference on System Sciences (HICSS)* (ss. 4446–4453).
- Williams, T. (2005). Assessing and managing project complexity. *International Journal of Project Management*, 23(4), 269–279.
- Williams, T. (2005). Assessing and moving on from the dominant project management discourse in the light of project overruns. *IEEE Transactions on Engineering Management*, 52(4), 497–508.
- Womack, J. P., & Jones, D. T. (2003). *Lean Thinking: Banish waste and create wealth in your corporation*. Free Press.
- Writtenhouse, S. (2016). A guide to Trello for life and project management. *MakeUseOf*. Erişim tarihi 14 Haziran 2025, <https://www.makeuseof.com/tag/guide-trello-life-project-management/>
- Wysocki, R. K. (2007). *Effective project management* (4. baskı). John Wiley & Sons.
- Yılmaz, B. S., & ark. (2020). *Türkiye’de çevik ve klasik yazılım geliştirme yaklaşımları...* BEÜ Fen Bilimleri Dergisi.
- Yılmaz, B. S., Gürson Dolar N. G., & Öz, H. G. (2025). *Başarı ve başarısızlık faktörleri*. UBSUD, 1(1), 19–35.
- Yılmaz, H. Ç. (2018). *Çevik proje yönetiminin teknoloji alanında PRINCE2 proje yönetim metodolojisi ile karşılaştırılması* [yüksek lisans tezi]. Maltepe Üniversitesi, Sosyal Bilimler Enstitüsü.



Extended Summary

In the contemporary landscape of project management, organizations are increasingly confronted with dynamic environments characterized by uncertainty, complexity, and rapid change. Traditional project management methodologies, once dominant in industries such as defense, construction, and manufacturing, are often inadequate to address these evolving challenges due to their rigidity and sequential structures. Agile methodologies, by contrast, offer an adaptive, iterative, and collaborative framework that emphasizes customer satisfaction, continuous improvement, and flexibility. This study critically examines both traditional and agile project management approaches, with particular focus on their principles, practices, methodologies, and the contexts in which they are most effectively applied. It also analyzes widely used agile methods such as Scrum and Kanban, compares them with conventional models like Waterfall, and evaluates the role of project management tools including Jira, Trello, Asana, and Microsoft Project.

The discussion begins with an exploration of the evolution of project management as a discipline. From the early applications in monumental constructions like the Egyptian pyramids to the structured frameworks developed during the Manhattan Project, project management has evolved to address increasingly complex and large-scale endeavors. Traditional methodologies such as Waterfall are based on predictability, linear processes, and strict planning, which work well in environments where requirements are stable and changes are minimal. However, in modern business ecosystems characterized by volatility, these methodologies often fail to accommodate shifting needs and emerging uncertainties.

Agile project management emerged as a response to these limitations, initially within the software development industry. The Agile Manifesto emphasized individuals and interactions over processes, working solutions over documentation, customer collaboration over contract negotiation, and responsiveness to change over adherence to a plan. Iterative development cycles, known as sprints or iterations, ensure that value is delivered continuously and incrementally to stakeholders. This approach allows organizations to remain resilient in the face of uncertainty, while promoting transparency, communication, and stakeholder engagement. Agile methods are no longer confined to software projects; they have spread to sectors such as healthcare and construction, though their highest applicability remains in software due to the sector's inherent flexibility and dynamic requirements.

The study provides a comprehensive comparison between traditional and agile approaches. Traditional models, including Waterfall, Spiral, and V-Model, are characterized by long planning cycles, high documentation requirements, and low tolerance for change. Agile methodologies such as Scrum and Kanban, on the other hand, emphasize adaptability, iterative delivery, and strong stakeholder involvement. For example, Scrum structures projects into time-boxed sprints of 2–4 weeks, with roles such as Product Owner, Scrum Master, and Development Team clearly defined. The methodology incorporates practices such as daily stand-ups, sprint reviews, and retrospectives, fostering collaboration and continuous improvement. Kanban, meanwhile, relies on visual task boards, work-in-progress limits, and a pull-based workflow to ensure efficiency and balance in task management. While Scrum is structured and ritualized, Kanban offers greater flexibility, allowing tasks to flow continuously without fixed time constraints.



Through comparative analysis, this study highlights the strengths and weaknesses of each methodology. Agile methods excel in environments where requirements are ambiguous, changes are frequent, and rapid delivery of value is essential. Traditional models, however, retain relevance in sectors where predictability, compliance, and risk management are paramount, such as defense or infrastructure. Hybrid approaches, such as Rational Unified Process (RUP) and Evolutionary Development, attempt to combine the predictability of traditional models with the adaptability of agile frameworks, and are particularly useful in highly regulated industries like healthcare and automotive software development.

Empirical evidence from the CHAOS Report and other studies is also discussed, demonstrating that agile methodologies generally outperform traditional approaches across small, medium, and large-scale projects. Agile projects show higher success rates and lower failure rates, particularly in large-scale endeavors where flexibility and iterative feedback mechanisms are critical. For instance, while only 3% of large traditional projects in the CHAOS Report were deemed successful, agile projects achieved an 18% success rate in the same category. These findings underscore the growing recognition of agility as a cornerstone of modern project success.

The research also explores the technological dimension of agile project management, focusing on widely adopted software tools. Jira, a robust platform compatible with Scrum and Kanban, supports backlog management, sprint tracking, and advanced reporting. Trello, with its user-friendly interface and visual Kanban boards, is particularly effective for smaller teams and collaborative tasks. Asana provides streamlined task and workflow management with an emphasis on usability, while Microsoft Project offers comprehensive planning, scheduling, and reporting capabilities, albeit with less flexibility for iterative methodologies. The comparison shows that while Jira and Microsoft Project offer enterprise-level functionalities, Trello and Asana are more accessible and intuitive, making them suitable for small teams and startups.

The findings of this study reveal that methodology selection is not about choosing a universally superior approach, but about aligning project management practices with contextual needs. Agile approaches, especially Scrum and Kanban, are particularly advantageous for projects characterized by uncertainty, frequent requirement changes, and the need for rapid value delivery. Traditional approaches remain relevant in projects with fixed requirements, high regulatory oversight, and critical safety demands. The integration of project management software further enhances the effectiveness of these methodologies by facilitating communication, transparency, and monitoring.

Despite the advantages of agile methodologies, their implementation is not without challenges. Cultural resistance, lack of organizational alignment, inadequate training, and insufficient tool support can all undermine the success of agile transformations. Organizations must therefore invest in change management, build a culture that embraces collaboration and adaptability, and ensure that teams are adequately trained in agile principles and practices. Furthermore, leadership plays a crucial role in fostering an environment where agile can thrive, balancing autonomy with strategic direction.

In conclusion, this study underscores the transformative impact of agile methodologies in modern project management. By emphasizing flexibility, collaboration, and iterative value delivery, agile approaches address the shortcomings of traditional models and align better



with the demands of dynamic project environments. However, successful adoption depends on contextual suitability, organizational readiness, and the integration of appropriate tools and practices. Agile project management is not a universal solution, but when applied thoughtfully, it significantly enhances adaptability, efficiency, and project outcomes. Future research may explore the customization of agile practices across diverse industries, further optimizing their applicability and ensuring sustainable project success.



Ek bilgiler

Çıkar çatışması bilgisi: Sorumlu yazar, yazar ekibi adına çalışmada çıkar çatışması olmadığını kabul etmektedir.

Destek bilgisi: Çalışmada herhangi bir kuruluştan destek sağlanmamıştır.

Etik onay bilgisi: Çalışma etik kurul onayı gerektirmemektedir.

Katkı Oranı Bilgisi: Birinci yazarın katkı oranı %35, ikinci yazarın katkı oranı %35, üçünü yazarın katkı oranı %30 şeklindedir.

