



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



BÜYÜK ÖLÇEKLİ ORGANİZASYON
YAZILIM GELİŞTİRME PROJELERİNDE
KRİTERLERE DAYALI SÜREÇ-PROJE-
TAKIM EŞLEŞTİRMESİ

İbrahim YEL

DOKTORA TEZİ

Endüstri Mühendisliği Anabilim Dalı

Kasım-2023
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

İbrahim YEL tarafından hazırlanan “BÜYÜK ÖLÇEKLİ ORGANİZASYON YAZILIM GELİŞTİRME PROJELERİNDE KRİTERLERE DAYALI SÜREÇ-PROJE-TAKIM EŞLEŞTİRMESİ” adlı tez çalışması 28/11/2023 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Endüstri Mühendisliği Anabilim Dalı’nda DOKTORA TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Prof. Dr. Feyza ARIKAN ÖKTEMER

.....

Danışman

Doç. Dr. Mehmet Emin BAYSAL

.....

Üye

Prof. Dr. Orhan ENGİN

.....

Üye

Doç. Dr. Ahmet SARUCAN

.....

Üye

Doç. Dr. Hakan ÇERÇİOĞLU

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Mevlüt UYAN
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

İbrahim YEL

Tarih: 28/11/2023

ÖZET

DOKTORA TEZİ

BÜYÜK ÖLÇEKLİ ORGANİZASYON YAZILIM GELİŞTİRME PROJELERİNDE KRİTERLERE DAYALI SÜREÇ-PROJE-TAKIM EŞLEŞTİRMESİ

İbrahim YEL

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Danışman: Mehmet Emin BAYSAL

2023, 102 Sayfa

Jüri

Doç. Dr. Mehmet Emin BAYSAL
Prof. Dr. Feyza ARIKAN ÖKTEMER
Prof. Dr. Orhan ENGİN
Doç. Dr. Ahmet SARUCAN
Doç. Dr. Hakan ÇERÇİOĞLU

Proje Yönetimi, adında yer alan “yönetim” ifadesini dolduran, gerçek anlamda yönetsel bir faaliyettir. Yönetsel faaliyetlerin etkililiği ile kurulacak ve kurgulanacak olan yenilikler, kuruluşları başarıya ulaştıran kilometre taşlarındandır. Proje yönetimi kavramı içerisinde geleneksel yaklaşımlar ve çevik yaklaşımlar vasıtasıyla yönetsel bakış açısının projenin yönetimine dâhil edilmesi ile proje yönetimi standartlarına esneklik getirilmiştir.

Bu tez, yazılım geliştirme projelerinde karmaşık iş yürütme değişkenleri açısından Çok Kriterli Karar Verme (ÇKKV) yöntemlerinin uygulanabilirliği, çevik ve geleneksel yöntemlerin proje başarısına etkileri üzerine yapılmıştır. Tezde sırasıyla, yazılım projelerinin yönetilmesinin önemine dair detaylı açıklamalar, araştırma problemi olan proje-geliştirme süreci-takım atamasına dair detaylar, konusuna göre sınıflandırılmış literatür, yazılım geliştirme süreçlerine dair açıklamalar, tezde kullanılan bulanık ÇKKV yöntemlerinin detayları, önerilen yöntemin tanımlaması, doğrulama ve durum sonuçları ile yönetsel çıkarımlar yer almaktadır.

Aralık değerli Nötrosofik Z sayı kümeleri (NZN) ve bulanık karar verme yöntemleri, yazılım geliştirme projelerinde proje özelinde doğru geliştirme süreci seçimi ve bu sürece en yakın takımların projeye atanması probleminin çözümünde kullanılmıştır. NZN yöntemi temel alınarak geliştirilmiş, doğrusal mesafe yerine alan hesaplaması ile alternatiflerin sıralamasını yapan yeni NZN-Alan yöntemi literatüre kazandırılmıştır. NZN-Alan yöntemi, NZN yöntemi ile birlikte aynı probleme uygulanmış ve görece daha iyi sonuçlar elde edilmiştir. Böylece bulanık ÇKKV yöntemlerinin ve NZN-Alan yönteminin etkinliği, çözüm önerisi getirilen problem kapsamında duyarlılık analizi ile çözüm uzayı genişletilerek gösterilmiştir.

Durum analizinde NZN ile NZN-Alan yöntemleri kullanılarak elde edilen sıralamalar açısından projelerdeki kusur sayıları ve bunların çözüm süreleri analizleri yapılmıştır. NZN-Alan’ın NZN yöntemine göre kusur sayısı bazında %22, kusurun çözülme süresi bazında ise %25 daha iyi sonuç verdiği belirlenmiştir.

Anahtar Kelimeler: Nötrosofik, ÇKKV, Çevik, Geleneksel, Yazılım geliştirme yaşam döngüsü

ABSTRACT

PhD THESIS

MATCHING PROCESS-PROJECT-TEAM BASED on CRITERIA in LARGE-SCALE ORGANIZATION SOFTWARE DEVELOPMENT PROJECTS

İbrahim YEL

**Konya Technical University
Institute of Graduate Studies
Department of Industrial Engineering**

Advisor: Assoc. Prof. Mehmet Emin BAYSAL

2023, 102 Pages

Jury

**Assoc. Prof. Mehmet Emin BAYSAL
Prof. Feyza ARIKAN ÖKTEMER
Prof. Orhan ENGİN
Assoc. Prof. Ahmet SARUCAN
Assoc. Prof. Hakan ÇERÇİOĞLU**

Project management is a true management activity that lives up to the word "management" in its name. The innovations that are created and built with the effectiveness of management activities are among the milestones that lead organizations to success. Flexibility has been added to project management standards by incorporating the management perspective into the project management concept through traditional and agile approaches.

This dissertation is about the applicability of Multi-Criteria Decision Making (MCDM) methods in relation to complex business execution variables in software development projects and the impact of agile and traditional methods on the success of the project. The dissertation includes detailed explanations on the importance of software project management, details on the research problem of project development process - team allocation, literature categorized by topic, explanations on software development processes, details on the fuzzy MCDM methods used in the dissertation, description of the proposed method, validation and case results and management implications.

Interval-valued Neutrosophic Z-number sets (NZN) and fuzzy decision making methods are used to solve the problem of selecting the right development process and assigning the most suitable teams to this process in software development projects. Based on the NZN method, the new NZN-Area method has been introduced in the literature. This method ranks the alternatives by calculating the area instead of the linear distance. The NZN-Area method was applied to the same problem as the NZN method. Relatively better results were obtained. Thus, by expanding the solution space with sensitivity analysis within the framework of the proposed problem, the effectiveness of the fuzzy MCDM methods and the NZN-Area method is demonstrated.

In the case analysis, the number of defects in the projects and their solution times were analyzed in terms of the rankings obtained using the NZN and NZN-Area methods. It was found that NZN-Area gives 22% better results in terms of number of defects and 25% better results in terms of defect resolution time compared to the NZN method.

Keywords: Neutrosophic, MCDM, Agile, Traditional, Software development life cycle

ÖNSÖZ

Yazılım geliřtirmenin sürekli geliřen ortamında, büyük ölçekli kuruluşların başarısı, karmařık projeleri hassasiyet ve verimlilikle yönetme becerilerine baėlıdır. Özellikle yazılım geliřtirme projeleri, dijital çağın sürekli artan taleplerini karřılayan yenilikçi çözümler sunmak için insan yeteneklerinin, süreçlerin ve proje gereksinimlerinin uyumlu bir şekilde birleřtirilmesini gerektirir. "Büyük Ölçekli Organizasyon Yazılım Geliřtirme Projelerinde Kriterlere Dayalı Süreç-Proje-Takım Eřleřtirmesi" bařlıklı bu tez, bu dinamik alanın kritik bir yönünü arařtırmaktadır.

Yazılım geliřtirme dünyası, üretkenliėi, kaliteyi ve müşteri memnuniyetini artırmayı amaçlayan metodolojiler, araçlar ve teknolojilerde kayda deėer gelişmelere tanık olmuřtur. Bu dönüşüm ortamında, proje geliřtirme süreçlerinin ve geliřtirme takımlarının etkili bir şekilde uyumlařtırılması, proje başarısı için çok önemli bir faktör olarak ortaya çıkmıřtır. Gerekli beceri ve uzmanlıkla donatılmıř doėru ekibi bir araya getirme ve proje gereksinimlerine göre uyarlanmış uygun geliřtirme süreçlerini uygulama becerisi, büyük ölçekli kuruluşlarda zorlu bir mücadeledir. Uyumsuzlukların veya yanlış eřleřmelerin sonuçları, proje gecikmeleri, maliyet ařımları ve optimal olmayan ürün çıktıları dahil olmak üzere derin olabilir.

Bu tez, kriter tabanlı proje-geliřtirme süreci-takım eřleřtirme ve sınıflandırma tekniklerinin kapsamlı bir incelemesini sunarak bu zorluğun özüne inmektedir. İlgili literatürün, deneysel arařtırmaların ve öngörülerin kapsamlı bir şekilde incelenmesi yoluyla, yazılım geliřtirme projelerini büyük ölçekli kurumlar bağlamında en uygun süreçler ve takımlarla eřleřtirmenin nüanslarına ve karmařıklıklarına ıřık tutmayı amaçlamaktadır.

Burada sunulan arařtırma, titiz bir inceleme ve eleřtirel analizi temsil etmekte ve ařağıdaki temel soruları ele almaktadır:

- Büyük ölçekli kuruluşlarda yazılım geliřtirme süreçlerinin, projelerinin ve takımlarının eřleřtirilmesini etkileyen temel kriterler ve faktörler nelerdir?
- Daha bilinçli ve doėru proje-geliřtirme süreci-takım eřleřtirme kararlarını kolaylařtırmak için sınıflandırma teknikleri yaklařımlarından nasıl yararlanılabilir?
- Gerçek dünyadaki yazılım geliřtirme süreçlerinde kriter tabanlı eřleřtirme metodolojilerinin benimsenmesiyle iliřkili potansiyel faydalar, zorluklar ve sınırlamalar nelerdir?

Burada sunulan bulguların, yazılım geliřtirme alanındaki uygulayıcılar, arařtırmacılar ve karar vericiler için deėerli bir kaynak olarak hizmet edeceėini, proje sonuçlarını iyileřtireceėini ve büyük ölçekli kuruluşlar nezdinde yenilikçiliėi teřvik edeceėini umuyoruz.

Bu tez, özenli bir arařtırmanın, önemli bir tefekkür saatinin, bana rehberlik eden ve ilham veren birçok kiřinin sarsılmaz desteėini temsil etmektedir. Bu çalışmayı akademik camiaya sunmanın eřiğinde dururken, derin bir minnettarlık, tevazu ve heyecan duygusuyla doluyum.

Bu tezin ilham kaynaėı Endüstri Mühendisliėi mesleėine duyduğum derin tutkudur. Bu meslek merakımı cezbeden, aklıma zorlayan ve bana etrafımızdaki dünyaya dair eřsiz bir bakıř açısı saėlayan bir disiplindir. Bu arařtırma aracılıėıyla, yazılım geliřtirme ve endüstri mühendisliėi tekniklerini bu alanda uygulamanın geniř etkileri hakkındaki kolektif bilgiye katkıda bulunmaya çalıştım. Umudum, bu çalışmanın akademisyenler, uygulayıcılar ve proje yönetimi ile ilgilenen herkes için deėerli bir kaynak olmasıdır.

Her řeyden önce, tez danışmanım Doç. Dr. Mehmet Emin Baysal'a en içten řükranlarımı sunmak istiyorum. Sarsılmaz destekleri ve sabırlı rehberlikleri bu arařtırmanın řekillenmesinde etkili oldu.

Ayrıca, tez hazırlama sürecinde değerli geri bildirimleri ve yapıcı eleştirileri için tez komitemin değerli üyeleri Doç. Dr. Ahmet Sarucan'a ve Doç. Dr. Hakan Çerçioğlu'na teşekkürlerimi sunuyorum. Onların farklı bakış açıları bu çalışmanın derinliğini ve genişliğini zenginleştirdi.

Bu tezin hazırlanmasında sevgili eşim ve kızlarıma anlayışlı destekleri için teşekkürlerimi sunuyorum. Bugünlere gelmemi sağlayan, kıymetli babam Yusuf Yel ve annem Hatice Yel'e en derin şükranlarımı sunuyorum.

Son olarak, çalışmaları bu tezin üzerine inşa edildiği temeli oluşturan sayısız yazar, araştırmacı ve akademisyene derin saygı ve şükranlarımı sunuyorum. Katkılarınız kendi anlayışımı zenginleştirdi ve bu tezi hazırlamamı sağladı.

Bu tezi akademik camiaya sunarken, devam eden bilgi arayışında sadece küçük bir adımı temsil ettiğinin bilincindeyim. Bu entelektüel yolculukta bana katıldığınız için teşekkür eder, bu tezi endüstri mühendisliği tekniklerinin yazılım geliştirme projelerinde uygulanması alanındaki bilgi birikimine anlamlı bir katkı olarak görmenizi dilerim.

İbrahim YEL
KONYA-2023

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	viii
SİMGELER VE KISALTMALAR	x
1. GİRİŞ	1
1.1. Yazılım Geliştirme Proje Yönetiminin Önemi	1
1.2. Yazılım Geliştirme Yaşam Döngüsü ve Gereksinim Yönetimi	3
1.3. Yazılım Geliştirme Projelerinde ÇKKV Önemi	7
1.4. Araştırma Problemi ve Hedefleri	8
1.5. Proje Yönetiminde Süreç Atamasının Gerekçesi ve Önemi	11
2. LİTERATÜR TARAMASI.....	14
2.1. Yazılım Geliştirme Proje Yönetimi Metodolojilerine Genel Bakış.....	14
2.2. Proje Sınıflandırma ve Değerlendirme	15
2.3. Çevik ve Geleneksel Geliştirme Süreçleri	18
2.4. Yazılım Geliştirme Süreci Seçimine Dair Çalışmalar	23
2.5. Proje Başarısını Etkileyen Faktörler	25
2.6. Proje Çizelgeleme ve Atamalar	26
2.7. Nötrosifik Z Sayı Kümeleri ve Aralık Değerli Yöntemler	28
2.8. Bulanık AHS ve Nötrosifik Z Sayıları.....	30
2.9. Mevcut Model ile Diğerlerinin Eleştirel Analizi ve Motivasyon Noktaları	31
3. YAZILIM GELİŞTİRME SÜREÇLERİ	35
3.1. Geleneksel Proje Geliştirme Süreçleri	35
3.1.1. Şelale süreci.....	35
3.1.2. V-Model süreci	36
3.1.3. Spiral model süreci	37
3.2. Çevik Proje Geliştirme Süreçleri	38
3.2.1. Kanban süreci	38
3.2.2. Scrum geliştirme süreci	39
3.2.3. Extreme Programming (XP) süreci	40
4. BULANIKLIK ÇOK KRİTERLİ KARAR VERME TEKNİKLERİ	42
4.1. Bulanık AHS	43
4.2. Bulanık TOPSIS.....	44
4.3. Bulanık EDAS.....	45
4.4. Bulanık WASPAS	47
4.5. Nötrosifik Z Sayı Kümeleri (NZN).....	48
5. ÖNERİLEN YÖNTEMİN TANIMI	51

5.1.	NZN-Alan Yönteminin Ayrıntıları	51
5.2.	NZN-Alan Yöntemini Doğrulamak için Türetilen Durum Tanımları.....	55
5.3.	Bulanık ÇKKV'nin Yazılım Geliştirme Proje Yönetimine Uygulanması	56
5.4.	Problem Tanımı ve Kriter Değerlendirme Ayrıntıları	59
6.	DOĞRULAMA VE DURUM SONUÇLARI	68
6.1.	NZN- <i>Alanelips</i> için Tüm Durumların Duyarlılık Analizi	68
6.2.	Problemin Aralık Değerli WASPAS ile Çözümü	69
6.3.	Problemin Aralık Değerli EDAS ile Çözümü	70
6.4.	NZN-Alan ile Aralık Değerli WASPAS ve EDAS Kıyaslaması	72
7.	BULGULAR VE TARTIŞMA.....	74
7.1.	Bulgular	74
7.2.	Tartışma	75
7.3.	Gelecek Araştırmalar İçin Öneriler	76
8.	SONUÇLAR	78
KAYNAKLAR		80

SİMGELER VE KISALTMALAR

Simgeler

π	: Pi Sayısı
$\oplus$	: Bulanık Toplama Operatörü
$\otimes$	: Bulanık Çarpım Operatörü

Kısaltmalar

5N1K	: Ne, Neden, Nasıl, Niçin, Ne Zaman ve Kim
ACO	: Karınca kolonisi optimizasyonu (Ant Colony Optimization)
Ad-Hoc	: Amaca özel, niyere mahsus anlamına gelen Latince bir ifade
AHS	: Analitik Hiyerarşi Süreci
ACS-SPSP	: Karınca Kolonisi Strateji-Yazılım Proje Çizelgeleme Problemi (Ant Colony Strategy-Software Project Scheduling Problem)
BT	: Bilgi Teknolojileri
BWM	: En İyi-Kötü Yöntemi (The Best-Worst Method)
CHAOS	: Standish Group isimli araştırma şirketinin, yazılım geliştirme projelerindeki kaosu düzene çevirme fikrinden yola çıkarak hazırladığı rapora verdiği isim.
ÇKKV	: Çok Kriterli Karar Verme
CS	: Uzlaşma Çözümü (Compromise Solution)
CMM	: Yetenek Olgunluk Modeli (Capability Maturity Model)
CMMI	: Bütünleşik Yetenek Olgunluk Modeli (Capability Maturity Model Integration)
COPRAS	: Karmaşık Oransal Değerlendirme (Complex Proportional Assessment)
Covid 19	: Korona Virüs ve Hastalığı 2019 (CoronaVirusand Disease 2019)
DEMATEL	: Karar Verme Deneme ve Değerlendirme Laboratuvarı (The Decision Making Trial and Evaluation Laboratory)
DM	: Karar Verici (Decision Maker)
EBS	: Olay Tabanlı Çizelgeleyici (Event Based Scheduler)
ELECTRE	: Elimine Etme ve Seçenekleri Gerçekleştirme (Elimination and Choice Translating Reality)
ISBSG	: Uluslararası Yazılım Ölçüt Standartları Grubu (International Software Benchmarking Standards Group)
ISO	: Uluslararası Standardizasyon Organizasyonu (International Organization for Standardization)
IVHF	: Aralık Değerli Kararsız Bulanık (Interval Valued Hesitant Fuzzy)
IVN	: Aralık Değerli Nötrosifik (Interval Valued Neutrosophic)
IVHFE	: Aralık Değerli Bulanıkla Genişletilmiş (Interval Valued Fuzzy Extended)
HFCWW	: Kararsız Bulanık Kollektif Bilgelik Ağırlıklandırması (Hesitant Fuzzy Collective Wisdom Weighting)
KPMG	: Klynveld Peat Marwick Goerdeler
MCGA	: Çok Kriterli Grup Değerlendirme (Multi-Criteria Group Assessment)
MCGDM	: Çok Kriterli Grup Karar Verme (Multi-Criteria Group Decision Making)
NP	: Deterministik Olmayan Polinom (Non-Deterministic Polynomial)

NZN	: Aralık Değerli Nötrosofik Z Sayı Kümeleri (Interval Valued Neutrosophic Z Number Sets)
NZN-Alan	: Aralık Değerli Nötrosofik Z Sayı Kümeleri – Alan Eklentisi
RAD	: Hızlı Uygulama Geliştirme (Rapid Application Development)
RUP	: Rasyonel Birleşik Süreç (Rational Unified Process)
SDLC	: Yazılım Geliştirme Yaşam Döngüsü (Software Development Life Cycle)
SFS	: Küresel Bulanık Kümeler (Spherical Fuzzy Sets)
PMI	: Proje Yönetim Enstitüsü (Project Management Institute)
PMBok	: Proje Yönetim Bilgi Tabanı (Project Management Body of Knowledge)
TOPSIS	: İdeal Çözüme Benzerliğe Göre Tercih sıralaması Tekniği (Technique for Order of Preference by Similarity to Ideal Solution)
TRNZN	: Yamuk Nötrosofik Z Sayıları (Trapezoid Neutrosophic Z Numbers)
OASIG	: Organizasyonel Yönler Özel İlgi Grubu (Organizational Aspects Special Interest Group)
WASPAS	: Ağırlıklandırılmış Bütünleşik Toplam Çarpım Değerlendirmesi (Weighted Aggregated Sum Product Assessment)
XP	: Uç Programlama (Extreme Programming)

1. GİRİŞ

1.1. Yazılım Geliştirme Proje Yönetiminin Önemi

Yazılım geliştirme projeleri, bir ihtiyaç doğrultusunda ortaya çıkan gereksinimlerin sistematik bir şekilde ele alınarak aksiyonlar silsilesi ile yazılım kodlarına dönüştürülmesidir. Yazılım projelerinin aslında maliyet düşürücü, verimliliği artırıcı birçok faydası bulunmaktadır. Hayatın içinden örnek vermek gerekirse, iletişim alanında akıllı telefonlarda bulunan uygulamaların insan hayatını nasıl kolaylaştırdığı, COVID-19 pandemi sürecinde teknolojik imkânlar dâhilinde uzaktan eğitim sistemlerinin ne denli kullanım etkinliği sunduğu tartışılmaz bir gerçektir. Hayatımıza teknoloji ve yazılımların yansımaya sayısız örnek verilebilir.

Gelecekte, teknolojinin ilerlemesi, açık kaynak kodlu ücretsiz yazılım geliştirme araçlarının artıyor olması, nesnelerin interneti ile teknolojinin hayatın her alanında aktif olacağı gibi etkenler, yazılım geliştirme projelerinin yönetimini başarılı bir şekilde tamamlamanın sağlayacağı maliyet kazanımını önemli ve öncelikli bir konu haline getirmiştir. Bu nedenle yönetsel anlamda proje başarı dinamiklerini etkileyen faktörlerin incelenerek, geliştirme süreci seçimi ve kaynak ataması hususlarını içeren bu çalışmanın, yazılım projelerinin başarı oranının artırılmasına katkıda bulunacağı düşünülmektedir.

Yazılım geliştirme projelerinde geçmişten günümüze, başarı oranları ile ilgili yapılan çalışmalarda proje başarı oranlarının düşük olduğu görülmektedir. Bu nedenle, yazılım geliştirme faaliyetini yürüten birden fazla fonksiyonel birime sahip büyük ölçekli organizasyonlarda yazılım geliştirme projelerinin başarı oranlarının nasıl artırılabilirliği araştırılmıştır. Birden fazla yazılım geliştirme takımı bulunan, bu takımlara birden fazla yazılım geliştirme süreci üzerinden proje atayabilen organizasyonlarda bu tezde zikredilen çözümün sağlayacağı katkı büyük olacaktır. Özellikle büyük ölçekli yazılım geliştirme ile ilgili ana faaliyetleri yürüten organizasyonların; yazılım geliştirme sürecinin seçimi ve bu seçim sonrasında ilgili proje adımlarının çizelgelenmesi konularında çok boyutlu problemler ile karşılaşması muhtemeldir. Bu problemlerin, yazılım geliştirme projelerindeki başarısızlık oranlarını yükselttiği düşünülmektedir. Yazılım geliştirme projeleri, proje yönetimi prensipleri göz önünde bulundurularak değerlendirildiklerinde; geleneksel/şelale (traditional/waterfall) ya da çevik (agile) yöntemlerden hangisi ile geliştirilmiş olursa olsun, projelerin başarı oranları düşüktür.

Yazılım geliştirme projelerinde görev alan kişiler genellikle, üniversitelerin mühendislik bölümlerinden mezun, iyi eğitim almış, yetkin kişilerdir. İlerleyen başlıklarda değinildiği üzere, son 20-22 yıllık süre içerisinde, nitelikli insan kaynağının görev aldığı yazılım projelerinde başarı oranlarının düşük seyretmesi düşündürücüdür.

Yönetmel kararların iş çıktılarına etkisinin gerek verimlilik gerekse yapılan işin başarısına yönelik etkisi tartışılmazdır. Yönetmel açıdan problemlerin tespit edilmesi akabinde tespate göre uygun çözüm tekniklerinin uygulanması ile yönetmel bakış açısının çözüm odaklı etkisi de yüksektir. Bu bağlamda kullanılabilecek olan yöntem ve teknikler üzerinden, ham verinin işlenmesi sonucu ulaşılan bilginin çözümde kullanılması ile kalıcı çözümlere ulaşılabilir. Ayrıca bu şekilde, gelecekte karşılaşılabilecek sorunların çözümünü de sağlamak mümkündür.

Teknolojinin ilerlemesine paralel olarak akademik dünyadaki gelişmeler ile birbirini besleyen bir döngü de oluşmuştur. Bilgi teknolojileri alanına zamanında yatırım yapan kuruluşlar, yetişmiş insan kaynağının etkinliği ile birlikte problemlerine, teknolojinin desteğini alarak, sistematik çözümlerini yazılım geliştirme projeleri üreterek çözüm aramaktadırlar.

Bilgi teknolojileri alanında, yazılım geliştirme projelerinin başarılı sonuçlanabiliyor olması kuruluşlar için ulaşmak istenilen hedeflerine yaklaştırmacı katalizör olmaktadır. Her yazılım geliştirme projesi, operasyonel işlemlerin bilgisayarlar tarafından yapılması, insan makine ilişkisinde verimlilik artırıcı yeniliklerin yapılması, karar modelleri üzerinden verimlilik artırıcı-enerji azaltıcı katkılar sunması, yeni teknolojilere bütünleşme ile gelecekte olabilecek altyapısal ihtiyaçlara hazırlanılması vb. birçok geliştirme faaliyetlerini bünyesinde barındırır.

20. yüzyılın ortalarından itibaren teknolojinin katlanarak ilerlemesi, açık ve özgür yazılım geliştirme araçlarının artması, Nesnelerin İnterneti ve teknolojinin hayatın her alanında etkili olması maliyet tasarrufu faaliyetlerini önemli kılmaktadır. Yazılım geliştirme projeleri açısından bu önem, bu projelerin başarıyla tamamlanması olarak değerlendirilebilir. Dolayısıyla geliştirme süreci seçimi ve kaynak tahsisi de dâhil olmak üzere proje başarı dinamiklerini yönetimsel anlamda etkileyen faktörlerin incelendiği bu tezde, yazılım geliştirme projelerinin başarı oranının arttırılması amaçlanmaktadır.

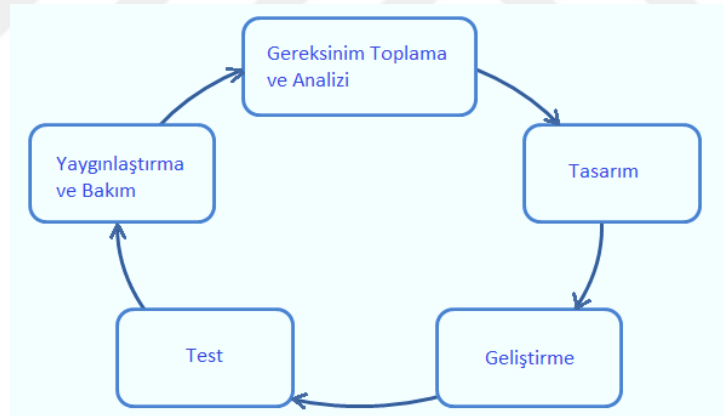
Yazılım geliştirme projelerinin başarılı sonuçlanmasına yönetmel açıdan bakıldığında, geliştirme süreci seçimi üzerinden geleneksel-çevik ya da hibrit modellerin seçimi etkilidir. Bununla birlikte insanın kendi yetkinliklerine uygun görevleri yaparak kazanacağı bireysel motivasyon ile işine kattığı değerin artacağı açıktır.

Proje – geliştirme süreci – takım değerlendirmelerinde, karar verme aşamasında bulanık ÇKKV yöntemlerinin kullanılması açısından tez çalışmasının sonuçları, kusur sayısı ve kusur çözüm süresinde önemli iyileştirmeler sağlanabileceğini göstermiştir. Kusur, yazılım geliştirme dünyasında son kullanıcılara geliştirmeler açılmadan önceki kalite, gereksinimlere uyum, performans gibi kriterler yönünden kontrol sonucu düzeltilmesi istenen bulgu kayıtlarını ifade etmektedir.

Yazılım geliştirme projelerinde önemli olan proje yaşam döngüsü ve bu döngüde dönüşüme uğrayan gereksinimlere dair detaylı açıklama sonraki başlıkta yer almaktadır.

1.2. Yazılım Geliştirme Yaşam Döngüsü ve Gereksinim Yönetimi

SDLC ifadesi, İngilizce olarak “Software Development Life Cycle” kelimelerinin ilk harflerinin kısaltmasından meydana gelmektedir. Türkçe anlamı ise “Yazılım Geliştirme Yaşam Döngüsü” olarak ifade edilmektedir. SDLC, başlangıçtan bitişe mantıksal sıralamayı takip eden bir süreçtir (Dennis ve ark., 2008). Şekil 1.1 basit SDLC akışını ifade etmektedir.



Şekil 1.1. SDLC Adımları (Peker, 2017)

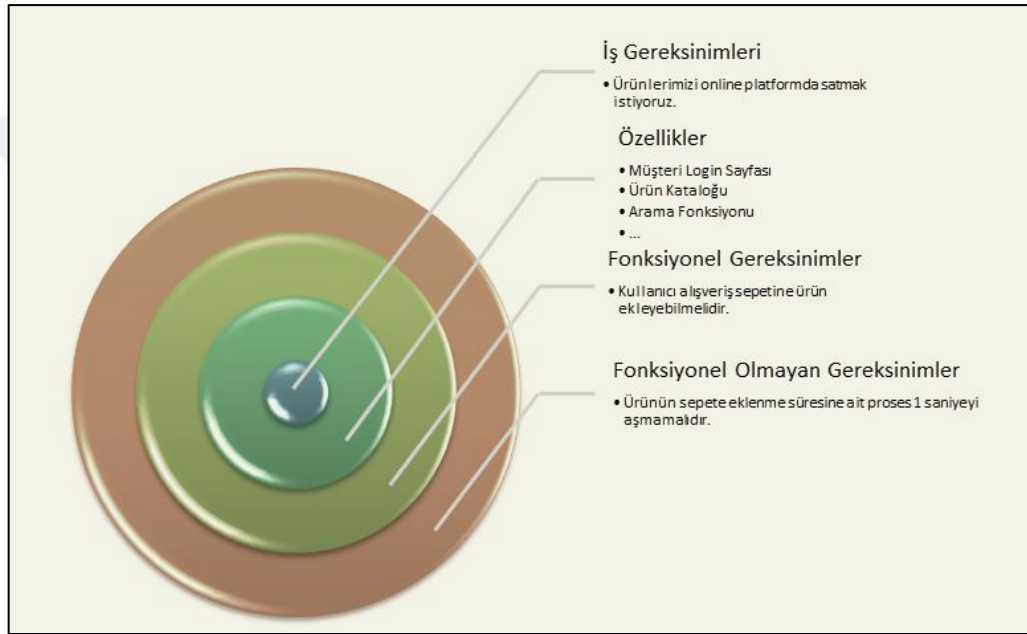
SDLC tanım olarak; başlatma, analiz, tasarım, uygulama ve bakımdan elden çıkarmaya kadar çok adımlı bir süreç aracılığıyla bilgi sistemlerini geliştirme, uygulama ve kullanımdan kaldırma sürecini betimlemektedir (Radack, 2009).

SDLC, baştan sona mantıksal bir yolda ilerleyen bir dizi adım ve alt adımdan oluşan ve aşamalı iyileştirmeyi içerecek şekilde yapılandırılmıştır. Yani, planlama adımının sonuçları, daha da genişletildikleri Analiz adımının bir başlangıç noktası görevini görmektedir. "Her aşama, daha önce yapılan işi geliştirir ve detaylandırır." Bu

kademeli iyileştirme yaklaşımı, doğası gereği, zaman ufku ilerlerken, hatayı azaltmak için talep tahminleri, üretim programları ve kapasite planlarının yönlendirildiği klasik "itme" üretim planlamasına benzemektedir (Polito ve ark., 2001). SDLC sürecinin ana adımlarının açıklamaları aşağıda maddeler halinde verilmektedir;

- **Gereksinim Toplama ve Analizi:** İlk adım olarak, geliştirme projesi ile ilgili olan gereksinimler, bilgi kaynaklarından ya da bakım faaliyetleri neticesinde elde edilen geri besleme kanallarından toplanmaktadır.
 - Gereksinimler yazılım geliştirme projelerinin başlaması için ilk adımı oluşturmaktadır. Gereksinimleri tanımlamak ve projenin uygulanması için gereken süreyi belirleme görevleri bu adımın içinde yer almaktadır (Leau ve ark., 2012). Özellikle sistem tasarımı aşamasında kullanılan girdi işlevini görmektedir.
 - Gereksinimler, yaşam döngüsü çıktılarından beslenebileceği gibi teknolojik veya yasal değişiklikler gibi dışsal faktörlerden de etkilenebilir.
 - Gereksinim tanımlamanın önemi için, özellikle doğru bilginin proje performansındaki kritik rolü nedeniyle matematiksel model geliştirilmesi gibi yaklaşımlar literatürde yer almaktadır (Hickey & Davis, 2003).
 - Büyük veri ve analitiği projelerinin gereksinimlerini ortaya çıkarmak için dört aşamalı bir yöntem Beatty ve Wiegers (2015) tarafından açıklanmıştır. Amaç, kararları olduğu kadar gereksinimleri de net bir şekilde belirlemektir. İş analistlerinin, gereksinimlerin ortaya çıkarılması sürecinde paydaşların alması gereken kararlara dayanarak gereksinimleri belirlemeleri önerilmektedir.
 - Öte yandan, Singh ve Saikia (2016) iyi bir noktaya değinmiş ve gereksinimleri yüksek kaliteli yazılımın temel odağı olarak betimlemişlerdir. Eksiksiz, açık ve doğru ifade edilmiş gereksinimlerin tanımına vurgu yaparak, teslim edilen yazılımdaki hataların, kusurların ve fonksiyonel sorunların ana nedeninin eksik yazılmış gereksinimler olduğunu ifade etmişlerdir. Gereksinimlerin matematiksel ve bilimsel bir yöntem kullanılarak tanımlanması gerekliliğine de çalışmalarında değinmişlerdir.
 - Başka bir çalışmada, iyi gereksinim tanımlamanın; özellikle dış kaynaklar tarafından yapılan projelerdeki başarı oranlarını direk etkilediği ifade edilmektedir (Iqbal ve ark., 2020).

- Benzer bir etkinin iç kaynaklar tarafından geliştirme aşamasında da olabileceği muhtemeldir. Dolayısıyla gereksinimleri bir problem tanımı gibi ele alarak teşhisin doğru konulmasının problem çözümüne etkisi gibi, doğru gereksinim tanımlamanın da proje başarısına etkisi olacağı göz önünde bulundurulmalıdır.
- Gereksinim tanımını Şekil 1.2'deki örnekte görüleceği üzere detaylandırarak ele almak çok daha sistematik bir yöntem sunmaktadır (Moustafaev, 2017).



Şekil 1.2. Gereksinim hiyerarşisi (Moustafaev, 2017)

- Gereksinim analizinin ve tanımlamasının etkililiği üzerine yapılan başka bir çalışmada 5N 1K soruları üzerinden gereksinim tanımlanmasında bir örüntü ile yapılmasının sonuçları araştırılmıştır. Buna göre 17 adet katılımcının anketler üzerinden değerlendirmeleri ışığında gereksinim tanımının belirli bir şablon üzerinden 5N 1K sorularına cevap verilerek oluşturulmasında anlaşılabilirliğin ve test edilebilirliğin ciddi oranda artmış olduğu ifade edilmiştir (Pabuccu ve ark., 2022).

SDLC sürecinin temel anlamda diğer adımları, karar verilecek olan yazılım geliştirme süreci yöntemlerine göre farklılıklar arz edebilmektedir. Kısaca açıklamak gerekirse;

- **Tasarım:** Bu adım, elde edilen gereksinimlerden çıkarılan analiz detaylarına göre yapılacak olan geliştirmenin tasarım aşamasını oluşturmaktadır.
 - Tasarım aşaması algoritma, sistem ve görsel tasarımı adımlarını içerisinde barındırır.
 - Bu aşamada potansiyel sorunlar gün yüzüne çıkmaktadır. Geliştirme aşaması öncesinde belirli bir yol haritası oluşur (Leau ve ark., 2012).
- **Geliştirme:** Bu adımda, tasarımı ve analizi hazır olan geliştirme faaliyetleri, yazılım geliştirme sorumluları tarafından gerçekleştirilir.
 - Yazılım geliştirme projeleri için genellikle en fazla zaman harcanan, asıl olarak faaliyetlerin planlanmasında ve çizelgelenmesinde en büyük belirsizliği içeren adımdır.
 - Geliştirme aşamasında tasarım özellikleri güncellenerek son halini almaktadır (Ruparelia, 2010).
- **Test:** Yazılımın geliştirilmesi sonrasında ilk adımda belirlenen gereksinimlerden yola çıkılarak, gereksinimlerin karşılanıp karşılanmadığının teyit edildiği adımı ifade eder.
 - Bu adımda yazılımın fonksiyonelliğinden, yapısından ve performansından önce statik testinin de yapılması gerekmektedir (Everett ve McLeod, 2007).
- **Yaygınlaştırma ve Bakım:** Test aşamalarını başarılı olarak tamamlayan geliştirmeler, artık yaygınlaştırma (deployment) işleminin yapılması adımına gelmiş olur.
 - Yaygınlaştırma sonrasında karşılaşılan durumlar ile bakım çalışmalarından doğan ek gereksinimler SDLC sürecinin girdileridir.
 - Dolayısıyla bu girdiler ile anlık olarak geliştiricilerce düzeltmeler yapılabilir ya da sonraki aşamada düzeltmek üzere gereksinimler bekletilebilir (Rani, 2017).
 - İlk adımda başlayan yazılım geliştirme görevlerine ait yaşam döngüsü bu adımla tamamlanır.

SDLC süreci boyunca gereksinimlerin doğru tanımlanması ile ne yapılacağı tanımlandıktan sonra, geliştirmelerin nasıl yapılacağına dair temel bilgi ve yaklaşımların öngörüsü için, bulanık ÇKKV hakkında yapılan çalışmalara aşağıda yer verilmiştir.

1.3. Yazılım Geliştirme Projelerinde ÇKKV Önemi

Deming (1998) tarafından yazılan Krizden Çıkış Kitabında, on dört temel ilke içerisinde başarısızlıkların %85'inin yönetim kaynaklı olduğu vurgulanmaktadır. Yazılım geliştirme projelerinin, gelişen teknoloji olanaklarına rağmen ve bilgiye kolay ulaşılabilen bir çağda, başarı oranlarının düşük olması yönetsel anlamda dikkat edilmeyen bazı değişkenlerin proje yaşam döngüsünü etkilediği izlenimi vermektedir. Bu bağlamda, yazılım geliştirme projelerinin geliştirme süreçlerinin ele alınmasında sistematik ve yönetsel bakış açısı ile geliştirme süreci seçimi kararının ÇKKV yöntemleri ile verilerek, projelerin başarısızlık oranlarının azalacağı öngörülmektedir.

Yel ve Engin (2022) tarafından bilgi teknolojileri alanında yapılan, değer akış haritalama analizi ile var olan bir akış üzerinde ne kadar iyileştirme sağlanabileceği benzetim ile desteklenerek tespit edilmiştir. Ayrıca bu çalışmadan yola çıkarak, yönetsel açıdan karar verme adımlarının bilimsel yöntemler üzerinden verilmesi ile projenin başarısına olacak etkisinin neler olabileceği araştırılmak istenmiştir.

Literatür incelendiğinde özellikle projelerin sınıflandırılması üzerine olan kısıtlı çalışmalar içerisinde, yazılım geliştirme projelerinin çevik ya da geleneksel geliştirme sürecine uygunluğu açısından değerlendiren çalışmaların nispeten az olduğu görülmüştür. Bu nedenle, ileri karar verme teknikleri olan bulanık ÇKKV yöntemlerinin bu alanda kullanılması ile belirsizlik içeren projelerin, yürütülecek olan geliştirme süreci bazında değerlendirilmesi bu tezin konusunu oluşturmaktadır.

Projelerin değerlendirmelerinin yapılması ve proje – kişi atamalarının yapılması alanında Yel (2019) çalışması incelendiğinde, sadece ana geliştirme süreçleri (Çevik ve Geleneksel) özelinde bu değerlendirmelerin yetersiz kalacağı düşünülmüştür.

Tez konusu ile ilgili sonraki çalışmalarda, her iki ana geliştirme süreci bazında üçer adet en fazla kullanılan yazılım geliştirme süreçleri temel alınarak, hangi projenin hangi geliştirme süreci ile yürütülmesi gerekliliğinin tespiti de ÇKKV yöntemleri üzerinden gerçekleştirilmiştir. Böylece bu teze ait önemli çıktılardan birisine ulaşılmıştır.

Yel (2019) çalışmasına ait sonuçlar üzerinden, proje sınıflandırmasından sonra bu projelere uygun takımların atanması gerekliliği dikkat çekici bulunmuştur. SDLC içerisinde görev alan ana aktörlerin yetkinliklerinin yine çevik ve geleneksel geliştirme süreci olmak üzere toplamda altı süreç için değerlendirilmesi ile takım üyelerinin projelere atanması analizleri bu tez kapsamında yapılmıştır.

Elde edilen sonuçlardan yola çıkılarak ilgili projenin hangi geliştirme süreci üzerinden yürütülmesi gerekliliği, geliştirme sürecine hangi takımın uygun olduğu bilgileri ile atama modeli uygulanmıştır. Atama sonuçlarının kıyaslanabilmesi için mevcut proje atamaları sonucunda sayısal veriler olduğu için kusur (defect) bazında analizler “eğer öyleyse” (what if) senaryoları üzerinden yapılmıştır. Kıyaslamalar ilgili karar alma aşamalarında bulanık ÇKKV yöntemlerinin kullanılması, aslında sayısallaştırılması zor olan değerlendirme kriterlerini sayısallaştırmıştır. Böylece geliştirme süreci atama yöntemleri açısından alternatifler, matematiksel kıyas zeminine taşınmıştır. Alternatif yöntemlere dair kusur temelli analizler, verimlilik anlamında çarpıcı sonuçlar üretmiştir.

Bu tez, yazılım geliştirme projelerinde karmaşık iş yürütme gibi değişkenler açısından ÇKKV yöntemlerinin uygulanabilirliği ve çevik ile geleneksel geliştirme süreçlerinin proje başarısına etkileri üzerine kurgulanmıştır. Karar verme aşamasında bulanık ÇKKV yöntemlerinin kullanılması, kusur sayısı ve çözüm süresi açısından önemli faydalar sağlanabileceğini göstermiştir. Bu kapsamda yeni tanımlanan NZN-Alan yöntemi, alternatiflerin sıralanmasında doğrusal mesafe yerine alan hesaplamasını kullanan bir uzantı olarak önerilmiştir. NZN-Alan yöntemi ile NZN yöntemi aynı problemde kullanılmış ve NZN-Alan yöntemi kusur sayı ve çözüm süreleri açısından daha iyi sonuçlar vermiştir. Ayrıca hem bulanık ÇKKV yöntemleri hem de bu tezin konusu olan NZN-Alan yöntemi, duyarlılık analizi ile çözüm uzayı genişletilerek aynı probleme uygulanmış ve çözüm etkinliği analiz edilmiştir.

1.4. Araştırma Problemi ve Hedefleri

Yazılım geliştirme projelerinin başarı oranlarının artırılması için proje kriterleri ağırlıklarının, yazılım geliştirme süreçleri için ideal değerlerin belirlenerek, proje takımının değerlendirme kriterleri bazında hangi geliştirme sürecine yatkın oldukları gibi karar noktalarında ÇKKV yöntemlerinin kullanılması gerektiği sektördeki ihtiyaçlar ve literatür incelemeleri ışığında gözlemlenmektedir.

Bu kapsamda gerek yönetim kademeleri, gerekse yazılım sektöründe çalışan kişilerin kolay kıyas edebilmeleri adına dilsel değişkenlerin karar modelinde bir anlam taşıdığı, kıyas ve sıralamada en iyi değer bulunmasında kullanıldığı bulanık (fuzzy) mantık içeren ÇKKV yöntemleri kullanılmıştır.

Yazılım projesi geliştiren büyük organizasyonların geliştirdikleri projelere örnek teşkil etmesi amacıyla 600’den fazla bilgi teknolojileri çalışanı ile katılım bankacılığı prensiplerine uygun olarak bankacılık uygulaması geliştiren Architech Bilgi Sistemleri ve Pazarlama Ticaret A.Ş.’ne ait, 2020 yılında yürütülmüş yedi proje özelinde kusur sayılarına ilişkin veriler kullanılmıştır.

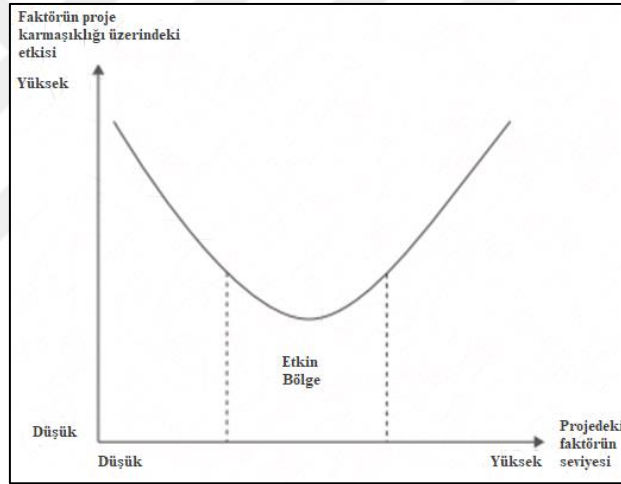
Yazılım geliştirme projelerinin başarısını etkileyen kriterler belirlenerek, bu kriterlerin ağırlıklandırılması için bulanık Analitik Hiyerarşi Süreci (AHS) yöntemi kullanılmıştır. Ayrıca, proje yönetimi çerçevesinde geleneksel ve çevik geliştirme sürecine dair seçim kriterlerinin ve alternatiflerinin değerlendirilmesinde bulanık ÇKKV yöntemlerinden olan NZN, yeni önerilen NZN-Alan, Aralık Değerli Nötrosifik EDAS (Interval Valued Neutrosophic EDAS, IVN-EDAS), Aralık Değerli Nötrosifik WASPAS (Interval Valued Neutrosophic WASPAS, IVN-WASPAS) yöntemleri kullanılmıştır. Tez’e konu olan problem çözümünde kullanılan yöntem aşağıda özetlenmiştir;

1. Yazılım geliştirme projelerinde, projelerin içerdiği ve geliştirme süreci seçimini etkileyen kriter ağırlıklarının bulunması.
2. Ağırlıklar, tez çalışması aşamasında karar verilecek olan bulanık karar verme yöntemi/yöntemleri üzerinden, elektronik ortamda uzman kişilere danışılarak yapılmış olan anketlerin değerlendirilmesi ile bulunmuştur (Khan ve ark., 2019).
 - a. Uzman kişiler Bilgi Teknolojileri alanında 5 yıldan fazla çalışmış olan 5 kişilik alanında uzman olan mühendis kadrosunu ifade etmektedir.
3. Geleneksel ve çevik geliştirme süreçleri üzerinden, en fazla kullanılan çevik süreçler olarak Kanban – Scrum – Extreme Programing(XP), geleneksel süreçler olarak Şelale – V Model – Spiral Model belirlenmiştir.
4. Yazılım projeleri değerlendirme kriterleri üzerinden ideal değerler, yazılım geliştirme süreçleri detayında aynı uzman kişilere iletilen anket üzerinden belirlenmiştir. Böylece geliştirme süreçleri bazında ideal değerler çoklu değerlendiricilerin değerlendirmeleri üzerinden elde edilmiştir.
5. Yazılım Projeleri karar kriterleri ağırlıkları üzerinden her yazılım projesi için projenin hangi yazılım geliştirme sürecine çok daha yakın olduğu belirlenmiştir.
6. Bu aşamada literatürde yer alan altı yazılım geliştirme süreci modelinin kendi aralarında sıralaması, bulanık karar verme yöntemi/yöntemleri ile karar kriterleri üzerinden kullanılabilirlik derecesine göre sıralanmıştır (Gupta ve ark., 2017).

Literatürde pek çok çalışmada bulanık karar verme yöntemleri, sıralama, ağırlıklandırma, en iyi seçeneğin seçimi gibi konularda kullanılmışlardır. Karar verme yöntemleri üzerinden yazılım geliştirme yöntemlerine ait ideal değerlerin belirlenerek, projelerin ve/veya takımların sıralanması sağlanabilir.

Şekil 1.3'te, proje karmaşıklığına etki eden faktörlerin seviyeleri belirli düzeye kadar karmaşıklığı azaltıcı etki yaparken, belirli düzeyden sonra karmaşıklığı daha da artırıcı etkiye sahip olabileceği ifade edilmektedir.

Bu aşamada, karmaşıklığı artırana kadar olan etki alanı, etkin alan, karmaşıklığı artıran alanlarda farklı yazılım geliştirme süreçlerinin veya hibrit yazılım geliştirme süreçlerin uygulanabilirliği tartışılabilir. Bu yaklaşımla karar verme aşamasında kullanılan kriterlerin ideal değerler üzerinden belirli yazılım geliştirme süreçlerine uyarlanabileceği bu tezde önerilmektedir.



Şekil 1.3. Tekil değişkenlerin proje karmaşıklığına etkisi (Maylor ve ark., 2008).

Ayrıca yazılım geliştirme projeleri açısından, Bilgi Teknolojileri takımına ait değerlendirme kriterleriyle bulanık ÇKKV yöntemleri üzerinden her iki yazılım geliştirme süreçlerinde kullanılan 3'er adet yöntemin ideal değerleri belirlenebilir.

Bu açıdan bakıldığında yazılım geliştirme projeleri için bilgi teknolojileri takımının geliştirme süreci bazında bulanık ÇKKV yöntemleri; sıralamalar üzerinden atama değil, geliştirme süreci bazında belirlenen ideal değerler üzerinden atama bakış açısı da verilmiştir.

1.5. Proje Yönetiminde Süreç Atamasının Gerekçesi ve Önemi

İnternetin çok yaygın olarak kullanıldığı, hayatımızın her alanında teknolojik cihazlar ile karşılaştığımız günümüz dünyasında, cihazların mekanik aksamlarına fonksiyonellik katan teknolojik çözümlerde yazılımlar aktif görev almaktadır. İş, yapılandırılmış görevler veya faaliyetlerden oluşan bir bileşimdir. Genel olarak, iş dünyasında görev ve faaliyet terimleri birbirinin yerine kullanılabilir. Görev terimi, insan etkileşiminin olduğu bir iş sürecindeki bir adımı tanımlamak için kullanılır. İnsan etkileşimi yoksa adıma faaliyet denmektedir (Çulha, 2014).

Yüksek teknolojinin kullanımı ile görevleri faaliyetlere dönüştürme sürecinde yazılım geliştirme projeleri önemli bir kilometre taşı oluşturmaktadır. İnsanlığın ilerleyişinin son yıllarda hızlanması, teknolojinin günlük hayatımıza girmesinin bu denli fazla ve baş döndürücü hızda olması, yazılım geliştirme projelerinin başarılı sonuçlanması ile orantılıdır.

Esasen yazılım geliştirme projeleri, emek yoğun görevleri içerir ve esas kaynağı da insandır (Chen ve Zhang, 2012). Yazılım projeleri, insanlar tarafından yapılan görevlerin faaliyetlere dönüştürülmesi aşamasında etkin olarak rol almaktadırlar. Yazılım projelerinin başarılı olarak sonuçlanmasına ait yayınlanan istatistikler incelendiğinde proje başarı oranlarının düşük olduğu görülmektedir.

Yazılım geliştirme projelerinin başarılı olması, faaliyetlerin çoğalması, verimliliğin artması; dolayısıyla maliyetlerin azalması ile hayat standartlarının artması gibi ilişkisel sonuçları da beraberinde getirecektir.

Bu tezde seçilen çalışma alanının şekillenmesi esnasında öncelikle, yazılım geliştirme projelerinin başarı ve başarısızlık oranlarının tespit edilmesi yönünde araştırmalar yapılmıştır. Bu konuda yapılan araştırmalarda geleneksel (waterfall) ile çevik (agile) yazılım geliştirme süreçleri bazında başarı ve başarısızlık oranları kıyaslanmaktadır. Yöntem kıyaslamasına geçmeden 1998 yılında yapılmış bir çalışmada, Çizelge 1.1’de görüleceği üzere, yazılım geliştirme projelerinin başarısızlık oranları farklı kaynaklar üzerinden sunulmaktadır (Ewusi-Mensah, 2003).

Çizelge 1.1. Yazılım projelerinden vazgeçilme derecesine ilişkin özet veriler

Yayınlanmış Çalışmalar	Tüm	Önemli	Kısmi
Standish Grup (1995, 1998)	%28	%46	-
KPMG (Cole 1995)	%10	%28	%24
OASIG (1996)	%40	%25	%80
Ewusi-Mensah ve Przasnyski (1994)	%44	%16	%9

Çizelge 1.1 üzerinden incelemeler yapıldığında ortalama %30 civarında olan projelerden bütünüyle vazgeçme oranı göze çarpmaktadır. Benzer şekilde, 2015 yılında Standish Group tarafından CHAOS isminde yayınlanan raporda çevik ve geleneksel geliştirme süreçleri bakımından başarısızlığa dair bilgiler Çizelge 1.2’te yer almaktadır (Hastie ve Wojewoda, 2015).

Çizelge 1.2. Proje büyüklüğüne ve uygulanan geliştirme süreçlerine göre yazılım projelerinin başarı oranları

Proje Büyüklüğü	Geliştirme Süreci	Başarılı	Sorunlu	Başarısız
Büyük	Çevik	%15	%59	%23
Büyük	Şelale	%3	%55	%42
Orta	Çevik	%27	%62	%11
Orta	Şelale	%7	%68	%25
Küçük	Çevik	%58	%38	%4
Küçük	Şelale	%44	%45	%11

2015 yılındaki yayına göre ortalama başarısız olan projeler %19 civarında iken, 2018 yılında aynı kuruluşun yayınında daha genel bir anlamda %15 oranına kadar gerilemiş olduğu Çizelge 1.3’te görülmektedir (Antony, 2018).

Çizelge 1.3. 2018 yılı Chaos raporuna göre proje başarı oranları

Yöntem	Başarılı	Sorunlu	Başarısız
Çevik (Agile) Süreçler	%42	%50	%8
Geleneksel-Şelale (Waterfall) Süreçler	%26	%53	%21

Yapılan araştırmalarda 22 yıllık sürede başarısızlık oranlarının %30 değerlerinden %15 değerlerine kadar gerilemesi, aslında gelişen teknoloji, iletişim kaynakları ve proje yönetim felsefelerinin yaygınlaşması yazılım projelerinin başarısına olumlu etki ettiği sonucunu göstermektedir. Ayrıca, son 20 yılda daha etkin kullanılan çevik süreçler ve açık kaynak kodlu yazılımların kullanımının artması sonucunda yalın yazılımların geliştirilmesi ile başarısızlık oranının azaldığı da ifade edilebilir.

Ancak, projelerin yönetimindeki başarı oranının düşük olması terazinin bir kefesine konulurken, gerçekten başarılı ve tek seferde doğru sonuca ulaşan projelerin oranının düşük olması da diğer kefeye konularak, Çizelge 1.3’ün yorumlanması daha anlamlı olacaktır. 1998 yılına ait çalışmadaki başarı oranı (100 – “tüm ve önemli oranda vazgeçme oranları ortalamalarının toplamı”) %40, 2015 teki çalışmada (ortalama) %26 ve 2018’deki çalışmada ortalama %34 olması, tezin içeriğindeki yönetsel bakış açısı ile

problem çözümünün uygulanmasını gerekli kılmaktadır. İngiltere’de çıkan bir haberde özel bilgi teknolojileri sektörüne aktarılan kaynak tutarı üzerinden %30 oranında başarısızlık oranına sahip yazılım geliştirme projeleri için aslında boşa harcanan kaynak ile yılda 7000 ilkokul ya da 75 hastane yapılabileceği ifade edilmiştir (Harley, 2007). Bu rakamların büyüklüğü aslında yazılım geliştirme projelerinin başarısızlığının maliyetinin ne denli yüksek olabileceğini de anlatmaktadır.

Bilgi teknolojileri (BT) alanında yürütülen projelerin başarısızlıkları üzerine yapılmış bir çalışmada, çoğu BT projesinin, BT proje yöneticileri tarafından yanlış yönetilmesi, müşterilerin teknik ayrıntılar hakkında bilgi eksikliği ve proje sahibi ile proje ekibi arasındaki iletişim eksikliği gibi çeşitli nedenlerle başarısız olduğu ifade edilmektedir (Korkmaz, 2021).

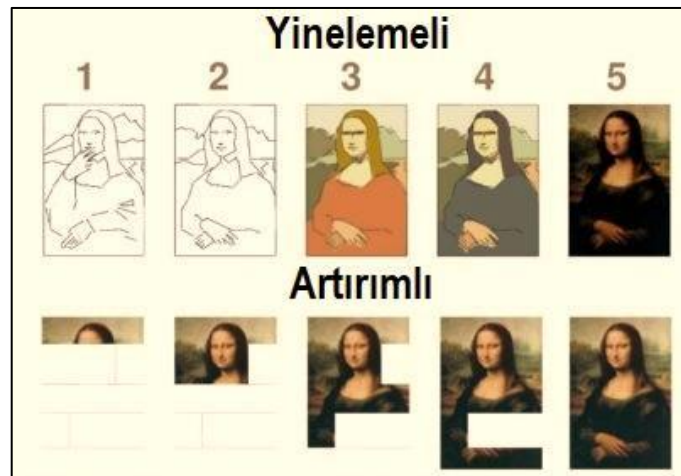
Çalışanlar açısından kendi kriterlerine daha uygun projelerin başarı oranını artırıcı etki yapacak süreç ataması ile ele alınması; işin gerekliliklerine uygun yetkinlikleri taşıyan kişilerin doğru işlere atanması ile iş tatmini ve performansın artacak olması (Heinsman, 2008) bu tezin sektöre ekleyeceği bir fayda olarak düşünülebilir.

2. LİTERATÜR TARAMASI

2.1. Yazılım Geliştirme Proje Yönetimi Metodolojilerine Genel Bakış

Yazılım geliştirme projelerinin yürütme süreçlerinde en temel anlamda iki adet farklı yaklaşım bulunmaktadır. Bunlardan ilki geleneksel geliştirme süreci olarak adlandırılan ve şelale yöntemi olarak da bilinen itme sistemine dayalı olan geliştirme sürecidir. Bir diğeri ise özellikle 2001 yılında çevik manifesto (Fowler ve Highsmith, 2001) ile yaygınlaşması hızlanan ve sadece yazılım dünyasında değil proje yönetimi alanında da uygulamaları olan çevik yöntem sürecidir. Bu yöntem itme sisteminden ziyade gereksinimlerin çekme sistemine dayalı olarak çok küçük aralıklarla proje geliştirme aşamalarına alınmasını içermektedir. Çevik yazılım geliştirme süreçleri, 20. yüzyılın son on yılından bu yana yazılım geliştiricileri tarafından giderek daha fazla kullanılmaktadır (Begel ve Nagappan, 2007). Bu yöntemlerin yanında Project Management Institute (PMI) tarafından yayınlanan Project Management Body of Knowledge (PMBok) (PMI, 2017) içerisinde geleneksel süreçlerden çevik süreçlere geçiş aşamalarında artırımlı (incremental) ve yinelemeli (iterative) olmak üzere iki geliştirme sürecinden de bahsedilmektedir.

Artırımlı geliştirme süreci bütünün küçük parçalara bölünmesi ile adım adım geliştirmelerin kullanıma açılması prensibine dayanmaktadır. Yinelemeli geliştirme süreci ise birikimli geliştirme adı altında daha ziyade taslak çalışma çıktıları üzerinden nihai sonuca ulaşırken proje çıktılarının olgunlaşmasını aşamalar halinde takip eder. Şekil 2.1, bu yaklaşımlara dair anlam karmaşasını özetlemektedir.



Şekil 2.1. Yinelemeli ve Artırımlı geliştirme süreçleri Mona Lisa Örneği (Anonim, 2019)

Şekil 2.1’i destekler nitelikte PMBoK tarafından verilen çizelgede (Çizelge 2.1) çevik ve geleneksel geliştirme süreçlerinin uygulama alanları projenin uygunluk kriterleri ve örnek proje açıklamaları paylaşılmaktadır.

Çizelge 2.1. Proje ihtiyaçlarının, karmaşıklığının ve büyüklüğünün değerlendirilmesi (PMI, 2017)

Geliştirme Süreci	Uygunluk Kriterleri	Örnek Projeler
Çevik	Değişikliklerin kolay ve harcanan zamanın maliyetli olmadığı durumlar	Entelektüel varlık ve araştırmalar bazlı yazılım geliştirme projeleri
Plan Bazlı	Değişikliklerin maliyetli olduğu, zamanın koordineli kullanılmasının önemli olduğu durumlar	Geçmişte benzerleri yapılmış, inşaat ya da fiziksel varlıklar üzerine yapılacak projeler
Yinelemeli (Iterative)	Dinamik gereksinimler ve faaliyetler doğru kabul edilene kadar tekrar edilme durumları	İdeal çözüme ulaşma aşamasına kadar öğrenmenin ve düzeltmenin beklendiği projeler
Artımlı – Birikimli (Incremental)	Sık ve küçük teslimatlar ile birlikte dinamik gereksinimlerin gerektiği durumlar, küçük birikimlerin tesliminin ana amaç olduğu durumlar	Müşterilerin ya da işin kısmi çıktıları erken ve sık olarak görmek istediği projeler
Hibrit	Paydaşların bir yöntem üzerine uzlaşmadıkları ve başka yöntemlerle ilgilendikleri durumlar	Farklı kaynakların ve deneyim seviyelerinin söz konusu olduğu veya yeni yöntem ve tekniklerin araştırıldığı projeler

Çizelge 2.1 incelendiğinde, PMI kuruluşu bakış açısıyla, proje yönetim süreçlerinin sınıflandırılmasında proje yapısı, müşteri ve iş özellikleri ile paydaşların yaklaşımına göre karar verilebilecek tanımlamaların yapıldığı gözlemlenmektedir. Bu yaklaşım tez konusunun içeriği ile büyük ölçüde tutarlılık göstermektedir.

2.2. Proje Sınıflandırma ve Değerlendirme

İngilizce “Categorization” kelimesinin Türkçe karşılığı Türk Dil Kurumu sözlüğünde “sınıflandırma” olarak verilmektedir. Belli, benzer özelliklere sahip bir dizi öge birden fazla kategori içinde bulunabilir. Kategori tanımı birbirini dışlamaz niteliktedir. Diğer taraftan sınıf terimi bir diğerini dışlayıcı bir anlama sahiptir (Archibald, 2004). Tez boyunca “categorization” ifadesi yerine sınıflandırma ifadesi kullanılacaktır. Tez’de kullanılan sınıflandırma ifadesi ile ilgili olarak, dışlayıcı anlamındaki kullanım kast edilmemektedir. Bu nedenle, tez içerisinde proje sınıflandırılması sistematığı tanımında yazılım geliştirme süreci ataması ana adımlardan birisidir.

Yazılım geliştirme projeleri, aslında tek başına benzer yaşam döngüsü aşamalarını içeren ve benzersiz bir süreç yönetimi sürecine sahip olan proje tipi olarak tanımlanabilir. Bu tanımlama içerisinde yazılım için gerekli donanımsal parçaların üretimi kapsam dışındadır. Proje sınıflandırmalarında; proje kapsamı, büyük – küçük projeler, proje

karmaşıklığı, iç ya da dış müşteri için yapılan projeler, müşterinin projeye katkı/katılım derecesi, projelerin risk seviyeleri, mega projeler, tek başına projeler ve destekleyici altyapı projeleri, standart ya da değişken projeler olarak ayrıştırılmıştır (Archibald, 2004).

Organizasyonel ortamın yazılım geliştirme süreci seçiminde kullanılması aşamasında en kapsamlı çalışma Farrell (2008) tarafından yapılmıştır. Bu çalışmada Mintzberg (1989) tarafından tanımlanan organizasyon tipleri ile organizasyonel karakteristiklerin eşleştirmesi yapılmıştır (Çizelge 2.2). Sonrasında bu eşleştirmeden yola çıkılarak yazılım geliştirme süreçleri bazında organizasyon tipleri ile uygunluğu tespit edilmiştir (Çizelge 2.3).

Çizelge 2.2. Organizasyonel tiplerin ve karakteristiklerin eşleştirmesi

Tipler / Karakteristikler	Girişimci	Yenilikçi	Makine	Çeşitlendirilmiş	Profesyonel
Bürokrasi	Az	Az	Yoğun	Az ya da Yoğun	Yoğun
Yapı	Matris	Matris	Fonksiyonel	Fonksiyonel	Fonksiyonel
Karışıklık	Yüksek	Yüksek	Az ya da Yüksek	Az ya da Yüksek	Yüksek
Çevre	Basit	-	Resmi	-	Merkezi Değil
Çaba	Yüksek	Yüksek	Az ya da Yüksek	Az ya da Yüksek	Az ya da Yüksek
Çalışma Biçimi	Özel	Özel	Yüksek Hacim	Özel ya da Yüksek Hacim	Özel
Değişiklik	Yüksek	Yüksek	Düşük	Düşük ya da Yüksek	Düşük ya da Yüksek
Büyüklik	Küçük/Orta	Küçük/Orta	Orta/Büyük	Orta/Büyük	Orta
Esneklik	Yüksek	Yüksek	Düşük	Yüksek	Düşük
Problemler	-Savunmasız -Bağlayıcı -Dengesiz	-Standart Yok -Düşük Komuta Birliği	-Kontrol Takıntısı -İnsan Problemleri -Koordinasyon Problemleri	-Uyumlu Stratejiler veya Süreçler	-Yeni Ürün ve Hizmetlerin Üretilmemesi
Diğer	-Yüksek Görev Duygusu -Duyarlılık	-Takım Çalışması -Çok Orta Kademe Yönetici	-Etkin -Güvenilir -Kesin -Tutarlı	-Merkezi Yapı Altında Otonom Birimler	-Bağımsız Çalışanlar -Profesyonel Standartlar

Çizelge 2.3. Organizasyonel tipler ve geliştirme süreci eşleştirmesi

Geliştirme Süreci	Girişimci	Yenilikçi	Makine	Çeşitlendirilmiş	Profesyonel
Şelale	Hayır	Hayır	Evet	Evet	Evet
Spiral	Evet	Evet	Evet	Evet	Evet
Prototip	Evet	Evet	Hayır	Hayır	Evet
RAD	Evet	Evet	Hayır	Hayır	Evet
Çevik	Evet	Evet	Hayır	Hayır	Hayır
XP	Evet	Evet	Hayır	Hayır	Hayır
Formal Yöntemler	Hayır	Hayır	Evet	Hayır	Hayır
CMM/CMMI	Hayır	Hayır	Evet	Hayır	Evet
ISO	Hayır	Hayır	Evet	Hayır	Evet
Altı Sigma	Hayır	Hayır	Evet	Hayır	Evet
Ad-Hoc	Hayır	Hayır	Hayır	Hayır	Hayır

Proje yöneticilerinin bakış açısından karmaşıklığın yönetimi üzerine bir araştırma yapılmıştır. Bu çalışmada görev, organizasyon, teslimat, paydaşlar ve takım sınıflandırmaları üzerinden bir projenin yönetimini karmaşıklaştıran unsurların tanımlamaları yapılmıştır (Maylor ve ark., 2008).

Karmaşık projeler belirsizlik unsuru taşıyan ve bu unsurun projeye etki etmesi ile ortaya çıkan bir proje türü iken, karışık projeler ise içinde belirsizlik bulunmayan projeler olarak tanımlanabilir (Whitty ve Maylor, 2009).

Bosch-Rekveltd ve ark. (2011), proje karmaşıklığının yönetilmesi hususunda, kategorik olarak yaklaşım, teknik, organizasyonel ve çevresel faktörleri içerisine alan bir yaklaşımla deneysel ve teorik verileri birleştiren bir çalışma yapmışlardır.

Clarke ve O'Connor (2012), yazılım geliştirme projelerine etki eden durumsal faktörlerin Ana Kategori (8) – Kategori (44) – Alt Kategori (170) olmak üzere 3 hiyerarşide tanımlamasını yaparak tüm yönleri ile yazılım geliştirme projelerini etkileyen faktörlerin genel bir listesini vermişlerdir.

Endüstri dallarına göre çevik süreçlerinin başarısı alanında, farklı kültürlerde, farklı ülkelerde ikâmet eden en az 5 yıllık deneyimli, farklı pozisyonlarda 1002 katılımcı üzerinden bir analiz yapılmıştır. Bu analizde, sağlık sektörü ilginç bir şekilde ön plana çıkarak en yüksek istatistiksel doğruluğu içeren ve çevik yöntem ile başarı sağlanan sektör olarak tanımlanmıştır (Serrador ve Pinto, 2015). Ayrıca yazılım geliştirme projelerinin yürütüldüğü yüksek teknoloji ve profesyonel hizmetler kategorilerinde de başarı oranının fazla olduğu tespit edilmiştir. Çizelge 2.4'te bu bilgiler yer almaktadır.

Çizelge 2.4. Endüstriler bazında çevik süreç başarı kıyaslaması regresyon sonuçları

Endüstri	Verimlilik Faktörü	Örnek Adedi	Regresyon P Değeri
İnşaat	4,536	23	0,184
Finansal Hizmetler	4,621	73	0,747
Elektrik-Su-Gaz	4,435	23	0,584
Kamu	4,304	34	0,336
Eğitim	4,867	10	0,084
Diğer	4,484	53	0,0002
Yüksek Teknoloji	4,690	57	0,035
İletişim	5,105	35	0,570
Üretim	4,437	42	0,726
Sağlık	5,042	24	0,017
Profesyonel Hizmetler	4,530	22	0,034
Perakende	4,617	16	0,722

Güven (2021) çalışmasında projeleri, projenin kapsamı, kod satırı sayısı ve süresi üzerinden karmaşık, orta ve yalın olmak üzere üç kriterli sınıflandırma yapıp, bu projelere atanacak olan personellerin projelerin hangi aşamalarında katkı sağlayabileceği tanımlamaları ile benzetim modeli kurarak, atamaları analiz etmiştir. Hangi takımdan hangi sayıda olması gereğini araştırarak, tamamlanacak olan proje sayısının artırılması problemine çözüm aramıştır.

2.3. Çevik ve Geleneksel Geliştirme Süreçleri

Çevik (agile) geliştirme süreci, 2001 yılında yayınlanan Çevik Manifesto ile etkin ve hızlı bir şekilde yaygınlaşmıştır. Nesne tabanlı programlama, açık kaynak kodlu yazılımların yaygınlaşması ve kullanıcıların basit ekran veya uygulamaya yönelmeleri çevik süreçlerin yenilerinin doğmasına, var olanların daha da yaygınlaşmasına neden olmuştur.

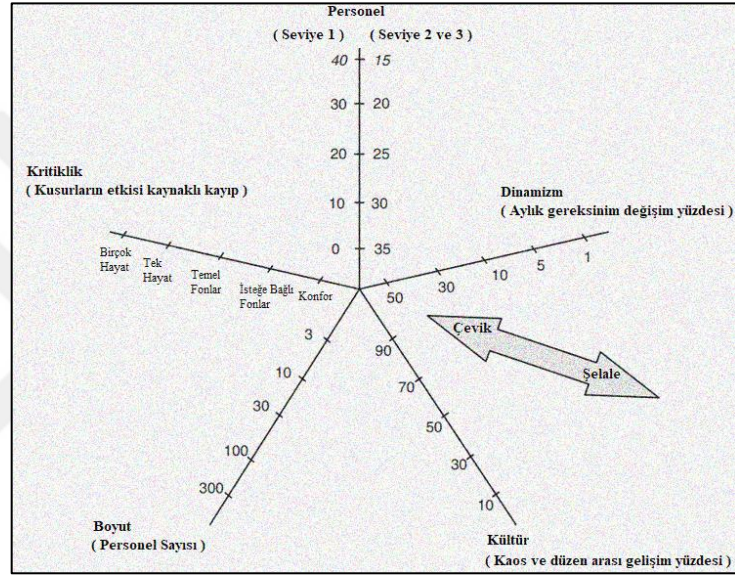
Çevik süreçler, hızlı, değişken, etkili iletişim, farklı bakış açıları ve çözümlemeleri ile yazılım geliştirme sürecini ilerleten yöntemler bütünüdür. Çevik geliştirme süreçlerinin felsefesinden doğan on iki ilkesi vardır (Fowler ve Highsmith, 2001). “Çevik Manifesto” nun ana ilkeleri aşağıdadır;

- En önemli öncelik değerli yazılımın erken ve devamlı teslimini sağlayarak müşterileri memnun etmektir.
- Değişen gereksinimler yazılım sürecinin son aşamalarında bile kabul edilmelidir. Çevik süreçler değişimi müşterinin rekabet avantajı için kullanır.
- Bir yazılım takımında bilgi alışverişinin en verimli ve etkin yöntemi yüz yüze iletişimdir.

- Sadelik, yapılmasına gerek olmayan işlerin mümkün olduğunca arttırılmaması sanatı, olmazsa olmazlardandır.

Çevik yazılım geliştirme süreçleri; yeniden iş yapma, başarısızlık, arızalar ve maliyet aşımı gibi durumlarla, birbirine bağlı bileşen yapıları sistemlerde uygulandığında faydasını gösterebilmektedir (Turk ve ark., 2002).

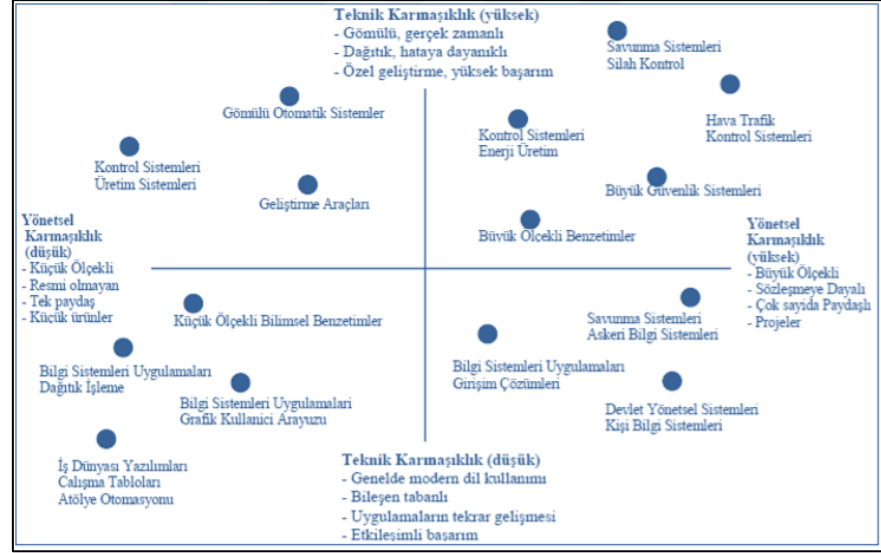
Geleneksel ve çevik yöntemlerin seçimine etki eden faktörler üzerinde sistematik ve yalın yaklaşımların gerekli olduğu aşikârdır. Literatürde bu konudaki çoğu çalışmaya kaynak olarak gösterilen bir grafik, Şekil 2.2’de yer almaktadır.



Şekil 2.2. Geliştirme Süreci seçimine etki eden boyutlar (Boehm ve Turner, 2003)

Şekil 2.2 incelendiğinde farklı boyutlarda yöntem seçimi sayısal tabanlı bir ağ grafiği üzerinde gösterilerek merkeze yakın olan proje değerleri için çevik yöntemlerin kullanılması gerektiği, merkezden uzak olan değerlere sahip projeler için ise geleneksel yöntemlerin seçilmesi gerektiği fikri sunulmaktadır.

2004 yılında yayınlanan “Yazılım Mühendisliği” kitabında Saridoğan (2004); yazılım geliştirme projelerini, teknik karmaşıklık seviyesi ve yönetsel karmaşıklık seviyelerine göre grafik üzerinde göstermiştir (Şekil 2.3). Grafik kullanıcı ara yüzü gibi basit içerikli projeler ile savunma sistemleri projeleri gibi karmaşık ve risk arz eden proje tiplerini bu grafik üzerinde yerleştirerek projelerin teknik ve yönetsel sınıflandırmasını yapmıştır.

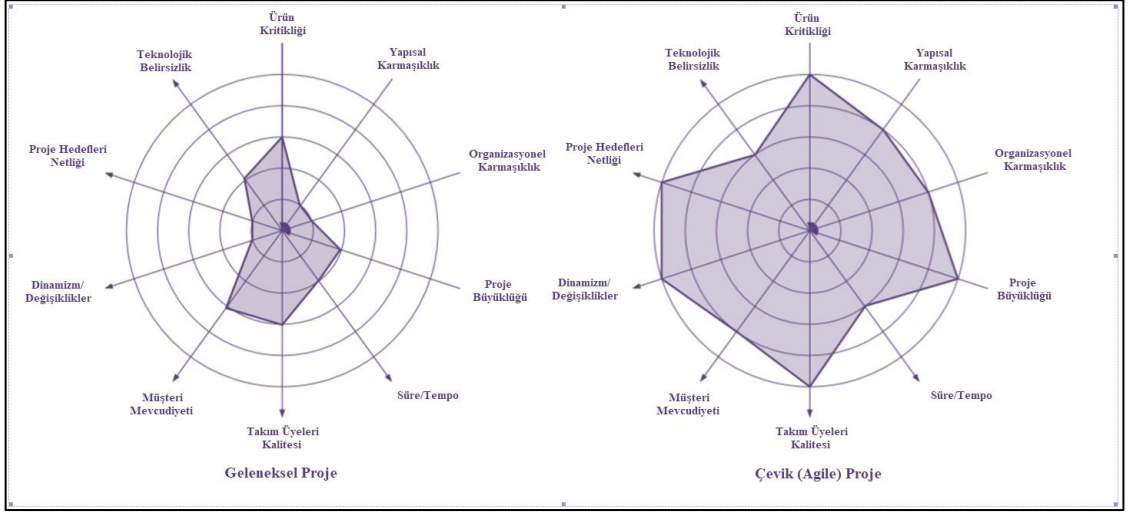


Şekil 2.3. Yazılım geliştirme projeleri sınıflandırma çalışması (Saridoğan, 2004)

Geleneksel (şelale) geliştirme süreci kavramı; süreç bazlı önceki adım sonlanmadan sonraki adımın başlamadığı, önceki adımın çıktısının sonraki adımın girdisi olduğu, akış bazlı sıralı SDLC sürecinin uygulanması olarak özetlenebilir. Önceki aşama tamamlanmadan bir sonraki aşamaya geçilmez. Bu modelde analiz ve tasarım aşaması bir kere tamamlandığında, proje boyunca değişmez (Farrell, 2008).

Fitsilis (2008) tarafından yapılan çalışmada çevik süreçlerin aslında proje yönetimi açısından önemli olan risk yönetimi, maliyet yönetimi, tedarik yönetimi gibi konuları kapsamadığı tespit edilmiştir.

Şpundak (2012) çalışmasında geleneksel ve çevik yöntemler için aslında on farklı kriter ile süreç seçimi için sınıflandırma yapmıştır. Bu çalışmada (Şekil 2.4) merkeze yakın alanı kapsayan projeler için geleneksel yöntemlerin uygulanmasını, merkezden uzak olan projeler için ise çevik yöntemlerin uygulanması gerektiğini ifade etmiştir.

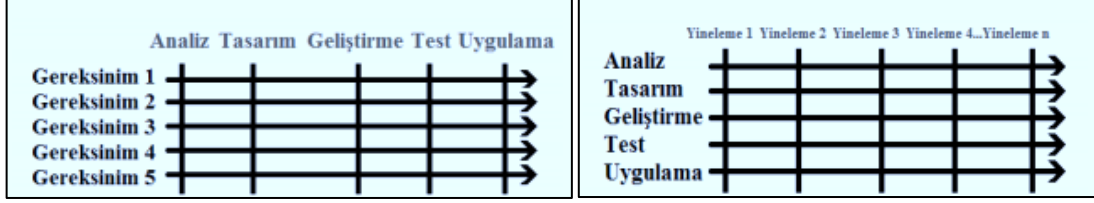


Şekil 2.4. Geleneksel ve çevik geliştirme süreci seçim kriterleri ağ grafiği (Şpundak, 2012)

Proje içeriği özelinde, proje geliştirme süreci seçimi üzerine kesin sonuç veren bir yöntem bulunmamaktadır. Proje bazında geliştirme sürecinin belirlenmesi aşamasında, proje sınıfları ile organizasyonel çevrenin karakteristiklerinin dikkate alınması gerekmektedir (Şpundak, 2014).

Proje yönetimi süreci için tanımlanmış Bütünleşik Yetenek Olgunluk Modeli (CMMI) ile Scrum uygulamalarının eşleştirilmesi sonrasında süreç yaklaşımlarının birbiri ile %46 oranında uyumsuz olduğu sonucuna ulaşılmıştır (Farid ve ark., 2016). Bu tespiti destekler nitelikte çevik yöntemlerin avantajları olduğu gibi dezavantajları da olabilmektedir. Literatürdeki bazı çalışmalarda çevik süreçlerin proje yönetiminin tüm gerekliliklerini karşılamadığı ifade edilmektedir (Salah ve ark., 2017). Üretilcek yazılımın karmaşıklık seviyesi arttıkça ihtiyaç duyulan analiz ve yönetim disiplininin artacağı ifade edilmiştir. Çevik yöntemlerin en erken zamanda müşteri için değer üretmeyi amaçlaması ve eldeki mevcut girdilerle kodlamaya geçilerek yinelemeler içerisinde durumun olgunlaşmasının hedeflenmesi nedeniyle nispeten daha az karmaşık sistemler için daha uygun olduğu ifade edilmiştir (Kayacan, 2017). Abdel-Basset ve ark. (2017)'nin yaptığı çalışmada, proje yaşam döngüsünün en önemli adımı olarak proje seçiminde, nütrosifik ortamda TOPSIS ve DEMATEL yöntemlerini kullanmışlardır. Benzer şekilde, Alava ve ark. (2018), tek değerli nütrosifik sayılar ve AHS yöntemleri ile uygulanabilir bir çerçeve tanımladıklarını iddia ederek bilgi teknolojileri alanında proje seçim problemine çözüm aramışlardır.

Gencer ve Kayacan (2017a) çalışmalarında, proje aşamalarının yazılım süreç seçiminde etkili olduklarına dair yorumlamalarda bulunarak, proje paydaşları bazında yetkinlik tanımlamalarını yapmışlardır.



(a) Şelale modeli süreç akışı

(b) Çevik model süreç akışı

Şekil 2.5. Akış bakımından çevik ve geleneksel geliştirme süreçlerinin karşılaştırılması (Gencer ve Kayacan, 2017a)

Geleneksel geliştirme süreçleri SDLC adımlarını sırası ile işleterek gereksinim bazında bir ilerleme sağlarken, çevik geliştirme süreçleri gereksinimleri parçalar halinde SDLC sürecine tabi tutarak kısa zamanda çalışan program parçalarının teslimatına yönelik bir yaklaşım sergilemektedir. Şekil 2.5 değerlendirildiğinde çevik geliştirme süreçleri yalın üretim / hücre tipi üretim tekniklerine benzerlik gösteriyor olsa bile, çevik geliştirme süreçleri ve yalın uygulamalar birbirinin alt kolu olarak değil tamamlayıcısı olarak ele alınmalıdır (de Oliveira Lacerda ve ark., 2020).

Çevik geliştirme süreçlerinin büyük ölçekli organizasyonlarda, büyük ölçekli projelere uygulanmasına yönelik bir çalışmada; iletişim, kod kalitesi, bütünleşme sorunu, planlama, önceliklendirme sorunları anahtar faktör olarak ele alınmıştır. Bu kapsamda, büyük ölçekli projelerin farklı takımlar tarafından birbirinden bağımsız parçalarının ele alınarak çevik geliştirme sürecinin uygulanabilirliğini ifade edilmiştir. Bununla birlikte, küçük ve orta büyüklükteki projelerde çevik geliştirme süreçlerinin daha etkili olduğunu da ifade edilmiştir (Lebdeh ve ark., 2020).

Tez kapsamında yapılan çalışmaya ÇKKV yöntemlerinin bilgi teknolojileri projelerinde kullanımına benzer, gereksinim analizi, çalışanların değerlendirilmesi, proje tipleri, organizasyonel yapı vb. kriterlerinin SDLC metodoloji seçimi için kıyaslandığı bir çalışmada, karar verme modelleri kullanılmıştır. Süreç modelleri kriterlerinin bilgi olarak çizelgeler aracılığı ile ilişkilerinin tanımlaması yapılmıştır. Çalışma sonucunda ise değerlendirilen kriterlere en iyi cevap veren model olarak spiral modelin (yaygın bir şelale geliştirme süreci) uygulanabilir olduğu ifade edilmiştir (Adanna ve Nonyelum, 2020).

Çevik yöntemlerin olumsuz etkileri ile ilgili olarak, Gheorghe ve ark. (2020) çalışmalarında, çevik geliştirme süreçlerinin müşteri ile iletişimi ve iş birliği kurmanın anahtar faktör olduğunu vurgularken, sprint (koşu) bazında yapılan geliştirmelerde özellikle zaman baskısının yazılım geliştiricileri üzerinde olumsuz bir etki yarattığından da söz etmektedirler. Bu zaman baskısının fazla mesai ve geç saatlere kadar iş yerinde zaman geçirme ile sonuçlanabildiğini ifade etmişlerdir.

Özdemir (2022) çalışmasında ise doğru ve başarılı çıktılara ulaşabilmek için doğru proje geliştirme sürecinin seçilmesi gerekliliğine vurgu yapmıştır. Çalışmanın kapsamında, çevik ve geleneksel süreçler bazında projelerin sınıflandırılmaları ve akabinde projelerin başarılı sonuçlanıp sonuçlanmaması ile ilgili sürecin sahip olduğu üyelik fonksiyonlarının çıktıları kıyaslanmıştır. Kıyaslamaları bulanık kavramlar üzerinden ulaşıkları sonuçlara göre yapmıştır.

BT projelerinin değerlendirilmesini zihinsel ergonomi açısından ele alan Akça (2022), çevik ya da geleneksel proje yönetim süreçlerinin proje içindeki çalışanların rolleri bazında bilişsel yük olarak proje yöneticileri ve analistler üzerinde anlamlı etkileri olduğunu tespit etmiştir. Benzer şekilde geliştirme süreçlerinin cinsiyet açısından bilişsel yük kriterine göre farklılığın olmadığını ifade ederek, çevik projelerin geleneksel süreçlere göre daha az bilişsel yük oluşturduğunu beyan etmiştir.

2.4. Yazılım Geliştirme Süreci Seçimine Dair Çalışmalar

Büyüközkan ve ark. (2004), doğru yazılım geliştirme stratejisinin maliyet ve kalite bakımından dengeli bir şekilde seçimine odaklanmışlardır. Bu çalışmada ekonomi ve kalite olmak üzere iki ana strateji kriteri ile kalite kriterinin altında ise tasarım, performans, uyum alt kriterleri üzerinden değerlendirme yapmışlardır.

Çevik yöntemlerin proje karakteristiğine göre seçiminin yapıldığı bir çalışmada projeler, gereksinim, büyüklük, geliştirme zaman aralığı, karmaşıklık ve risk değerleri bakımından 0 - 5 arasındaki ölçekle değerlendirilmiştir. Bu değerlendirmeler projelerin hangi çevik modele uygun olduğunu yedi alt kriterli başka bir seçim matrisi ile kıyaslanarak yapılmıştır (Mnkandla, 2008; Mnkandla ve Dwolatzky, 2007).

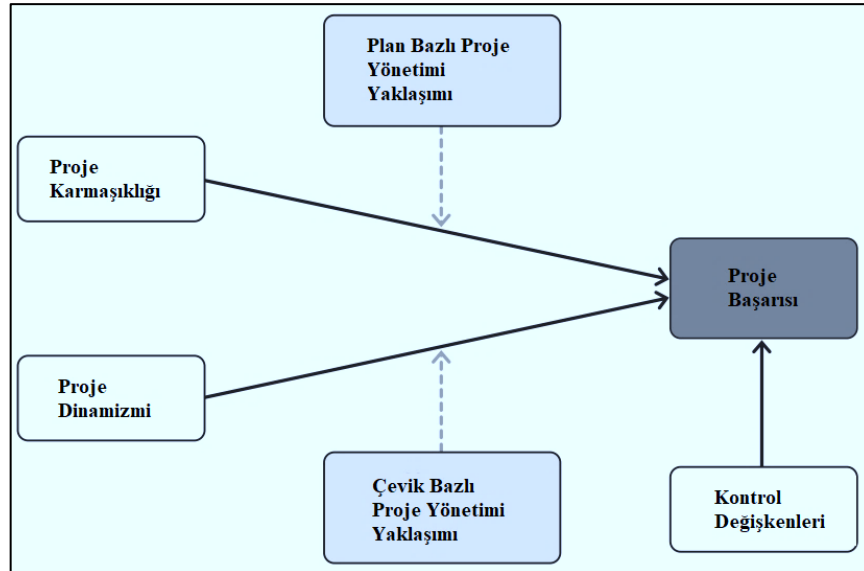
Khan ve Beg (2013), farklı yazılım geliştirme projeleri için risk değerlendirme bazlı doğru SDLC modelinin seçilmesi için pragmatik bir çözüm sunmuşlardır. Projeye en uygun model seçimi amaçlanarak geleneksel ve çevik metodolojilerin karakteristikleri yedi kriter (Çizelge 2.5) üzerinden değerlendirilmiştir. Aynı çalışmada uygulayıcılar

bazında yapılan bir ankette, proje başarısına yönelik en yüksek riskin doğru döngü sürecinin seçimi olduğunu sonucuna varılmıştır.

Çizelge 2.5. Yazılım geliştirme süreci seçim kriterleri

Sıra No	Değerlendirme Kriteri	Geleneksel Süreç			Çevik Süreç	
		V-Modeli	Yinelemeli - Artırmalı	Şelale	XP (Extreme Programming)	RUP (Rational Unified Process)
1	İş Kritikliği	Yüksek	Orta	Düşük	Düşük	Yüksek
2	Müşteri Katılımı	Düşük	Orta	Düşük	Yüksek	Düşük
3	Gereksinimlerin Netliği	Yüksek	Düşük	Yüksek	Düşük	Orta
4	Gereksinimlerin Değişkenliği	Düşük	Yüksek	Düşük	Yüksek	Orta
5	Son Kullanıcıların Mevcudiyeti	Orta	Yüksek	Düşük	Orta	Yüksek
6	Proje Büyüklüğü	Büyük	Büyük	Küçük	Orta	Büyük
7	İş ve Teknik ile ilgili Karmaşıklık	Düşük, Orta	Yüksek, Yüksek	Düşük, Düşük	Orta, Yüksek	Yüksek, Orta

Khan ve ark. (2014), yazılım geliştirme süreç grubu bazında AHS ile seçim çalışmasını, gereksinimler, maliyet ve zaman, gereksinim değişkenliği ve sistem karmaşıklığı açısından ele alarak SDLC süreç seçimi çalışması yapmışlardır.



Şekil 2.6. Geleneksel süreçler ile çevik süreçlerin kıyaslanması (Butler ve ark., 2020)

Çevik yazılım geliştirme modelleri için Butler ve ark. (2020) çalışmalarında, çevik metodolojilerinin başarılı olması anlamında, proje karmaşıklığı ile proje dinamizmi kriterleri açısından inceleme yapmışlardır (Şekil 2.6). İnceleme kapsamında geleneksel modeller ile çevik modeller kıyas edilmiş, yazılım geliştirme uzmanlarına gönderilen anket çalışmaları ile dinamizm kriterinin, çevik yöntemler ile daha iyi yönetilerek proje başarısındaki negatif etkilerin azaldığı tespit edilmiştir. Ayrıca, Vijayasarathy ve Butler (2015) geniş ve çeşitli bir katılımcı grubuna bir anket yapmışlardır. Bu çalışmada organizasyon, proje ve takım faktörleri kullanılmıştır. Böylece çevik, geleneksel, yinelemeli ve hibrit yöntemlerin organizasyonel, proje ve takım özellikleri özetlenmiştir.

2.5. Proje Başarısını Etkileyen Faktörler

Li ve Eberlein (2007), proje başarısı ya da yöntem seçimi için olmasa da proje başarısını doğrudan etkileyen, gereksinim tanımlama aşamasında kullanılabilecek yöntemlerin seçiminini proje kriterleri üzerinden yapmışlardır. Bu çalışma kapsamında yazarlar, aşağıdaki kriterleri karar aşamasında kullanmışlardır;

- Proje Büyüklüğü
- Proje Karmaşıklığı
- Gereksinim Değişkenliği
- Proje Kategorisi (Güvenlik Kritik Sistem, Güvenlik Kritik Olmayan sistem vb.)
- Güvenlik Kritiklik Seviyesi
- Zaman Kısıtları
- Maliyet Kısıtları

Liu ve Zhai (2007) çalışmalarında, projelerin sınıflandırılmasını risk değerlendirmesi açısından ele almışlardır. Çalışma içerisinde hangi kriterin risk değerlendirmesi açısından daha önemli olduğu sıralamasına ulaşmışlardır. Kriterlerin sıralaması aşağıda yer almaktadır;

- Takım Riski
- Organizasyonun Çevresel Riski
- Talep Riski
- Planlama ve Kontrol Riski
- Kullanıcı Riski
- Karmaşıklık Riski

Yazılım projelerinde sınıflandırma çalışmaları bakımından yer bulan bir diğer konu ise yazılım projeleri çaba tahminleme çalışmalarıdır. Bu bağlamda, Al Khalidi ve ark. (2012) çalışması dikkat çekmektedir. Proje çabasının tahmininde aşağıdaki kriterleri kullanmışlardır;

- Ürün Faktörleri
- Platform Faktörleri
- Personel Faktörleri
- Proje Faktörleri

Tahminleme konusunda Khatibi ve Bardsiri (2015) çalışmalarında, Uluslararası Yazılım Ölçüt Standartları Grubu (International Software Benchmarking Standard Group, ISBSG) tarafından toplanan yazılım projeleri bilgilerini kendi modellerinin doğruluğunu kıyaslamak için kullanmışlardır. İlgili kuruluşun topladığı ve elinde olan standart veriler arasında aşağıdaki 3 kriter, proje sınıflandırması için de önem arz etmektedir;

- Geliştirme Tipi
- Organizasyon Tipi
- Geliştirme Platformu

Destekleyici nitelikte, Ceylan ve Gürsev (2020) çalışmalarında, Scrum, Kanban ve Şelale yazılım geliştirme süreçleri bazında 4 farklı projenin özelliklerini ve başarı kriterlerini analiz etmişlerdir.

2.6. Proje Çizelgeleme ve Atamalar

Chang ve ark. (2008), literatürde yer alan yazılım projeleri ile ilgili çalışmalarında, çizelgeleme hususunda insan merkezli görev çizelgelemesi alanına gerekli önem verilmediğini belirterek, genetik algoritma merkezli yaklaşımlarını bu probleme uyarlamışlar ve gerçek durumları en iyi genetik algoritmanın temsil edebildiğini ifade etmişlerdir.

Luna ve ark. (2011) büyük firmaların, yazılım projelerinde büyük ölçeklerdeki projeler ile uğraştıkları, bu problemin ise yeni algoritmalar ile çözüme ulaştırılabileceğini öne sürerek, dört adet çok amaçlı algoritmayı yazılım geliştirme projeleri çizelgeleme problemine uygulamışlardır. Bu algoritmalar genetik algoritma, evrim stratejisi, ateş böceği algoritması, diferansiyel evrim algoritmalarıdır. En iyi çözümü de evrim stratejisinin verdiğini ifade etmektedirler.

Chen ve Zhang (2012) yazılım geliştirme projelerinin görev çizelgelemesi ile birlikte insan kaynağının tahsisini de gerekli kıldığını vurgulayarak, yazılım proje planlamasına yönelik esnek ve etkili bir model geliştirmek için, olay tabanlı çizelgeleyici (EBS) ve karınca kolonisi optimizasyonu (ACO) algoritmasını kullanmışlardır.

Xiao ve ark. (2013), yazılım projelerinin çizelgelenmesi problemini, karmaşık öncelik sıralaması, fazla görev sayısı ve insan kaynağının olduğu NP (non-deterministik polynomial) – zor olarak tanımlamışlardır. Bu projelerin çizelgelemesinde karınca kolonisi üzerine sezgisel bir yöntem geliştirmişlerdir. Karınca kolonisi stratejisi-yazılım projeleri çizelgeleme problemi (ACS-SPSP) ismini verdikleri yöntem ile genetik algoritmayı, 30 rastgele örnek üzerinden incelemişler ve ACS-SPSP yönteminin daha iyi oranları ve tutarlılığı sergilediğini ifade etmişlerdir.

Minku ve ark. (2013), yaptıkları çalışmada evrimsel algoritma kullanarak ve çalışanların farklı görevlere olan bağlılıklarından doğan fazla mesai gereksinimini dikkate alırken, normalleştirme yaparak çözümlerinin etkili sonuçlar verdiğini ve popülasyondaki normalleştirmenin yeni tasarımlarında anahtar faktör olduğunu ifade etmişlerdir.

Shen ve ark. (2015), yazılım projelerinin çizelgelemesinde, literatürdeki çoğu çalışmanın statik ve deterministik senaryolar üzerine yoğunlaştığını ifade ederek, kesinti olması durumunda bu planların performans düşüşüne hatta uygun olmayan çözüm olabileceğini ifade etmektedirler. Bu durumun çözümü için çok amaçlı evrimsel algoritma tabanlı önleyici yeniden planlama yöntemini önermişlerdir. 21 adet problem üzerine yaptıkları deneysel sonuçlarla yöntemlerinin etkili olduğunu ifade etmektedirler.

Ellappan ve Ashwini (2017) olay tabanlı planlayıcı ile dinamik personel atama sistemini kullanarak programlama konusunda personel atama problemine çözüm aramışlardır. Makine öğrenmesini de modellerine dâhil ederek karınca kolonisi optimizasyonu sezgiseli ile personeli görevlere atama problemine farklı bakış açısı getirmişlerdir. Shen ve ark. (2018) mevcut çalışmaların yazılım geliştirme planlama optimizasyonunda yalnızca sınırlı sayıda insan faktörünü içerdiklerini ifade ederek, yazılım geliştirme proje çizelgelemesinin dinamik optimizasyon problemi olduğunu vurgulamaktadırlar. İnsan faktörünün insanın artan yeteneklerinin de önceki görevlerinden bağımsız olarak, yeni projelerde önem arz ettiğini söylemişlerdir. Çok amaçlı iki aşamalı memetik algoritma ile önleyici ve dinamik olarak çizelgeleme problemine yaklaşmışlardır.

2.7. Nötrosofik Z Sayı Kümeleri ve Aralık Değerli Yöntemler

İlk olarak Zadeh (2011) tarafından bulanık sayı çifti olarak tanımlanan Z sayıları literatüre kazandırılmıştır. Nötrosofik Z Sayıları bilgi, belirsizlik ve hata bilgilerinden yola çıkarak karar verme aşamasında insan değerlendirme sistemindeki belirsizlik ve hata kavramlarını da matematiksel model içerisine katmıştır.

Chi ve Liu (2013) çalışmalarında aralıklı nötrosofik kümeler ile TOPSIS yöntemini birlikte kullanmışlardır. Ayrıca bu çalışmada hesaplama adımlarını ve işlem sıralarını da ayrıntılı olarak tanımlamışlardır.

Tavakkoli-Moghaddam ve ark. (2015)'ın çalışmalarında öklid mesafeli grup karar problemlerinde kriter ağırlıklarını belirlemek için aralık değerli kararsız bulanık TOPSIS yöntemi kullanılmıştır. Gerçek dünyadaki nesnelerin doğası gereği kararsız ve belirsiz olduğu belirtilmektedir. Ayrıca kararsızlığın ve belirsizliğin bilir kişi yargılarının doğasında olduğu belirtilmektedir.

Benzer Şekilde Gitinavard ve ark. (2016), kriterlerin optimal ağırlıklarını bulmak için aralık değerli kararsız bulanık (Interval Valued Hesitant Fuzzy, IVHF) Hamming mesafe ölçümüne dayalı genişletilmiş bir maksimum sapma yöntemi önermiştir. Alternatifleri sıralamak için Hamming mesafe ölçümü ile göreceli yakınlık katsayısını hesaplamışlardır. Tedarikçi seçim probleminde önerilen yöntemin kıyasladıkları çalışmaya göre daha iyi sonuçlar verdiğini bulmuşlardır.

Gitinavard (2019) adayları ve kriterleri değerlendirmek için iki farklı kararsız bulanık ölçeği kullanmıştır. Değerlendirmelerde, değerlendiricilerin risk algılarının kötümserlik ile iyimserlik arasında orta düzeyde olduğu varsayılmaktadır. Uzmanlar arasında bilgi paylaşımını ifade eden kolektif bilgelik kavramından esinlenerek Kararsız Bulanık Kolektif Bilgelik Ağırlıklandırma (Hesitant Fuzzy Collective Wisdom Weighting, HFCWW) yaklaşımını önermiştir. Karar vericilerin ağırlıkları bu yöntem kullanılarak belirlenmiştir. Bu amaçla 3 karar verici ve 5 alternatif stratejik proje 10 kriter üzerinden değerlendirilmiştir. Önerilen yaklaşımda 5 farklı yaklaşımla karşılaştırılmasında standart sapma göstergesinden daha iyi sonuçlar verdiği dikkat çekmektedir. Ancak önerdikleri yöntemin sonuçları, karşılaştırmalarda kullanılan standart sapma için 0-1 aralığında normalleştirme yapılmadan kullanılmıştır.

Çok kriterli grup karar verme (MCGDM) çözümünde uygulanabilecek, IVHF-COPRAS yöntemini temel alan yeni bir yöntem (IVHF-MCGA) Gitinavard ve ark. (2017) tarafından önerilmiştir. Ayrıca grup karar verme sürecinde her karar vericinin

ağırlığını tahmin etmek için yeni bir aralık değerli kararsız bulanık uzlaşma çözümü (IVHF-CS) yöntemi tanımlanmıştır. Önerilen yaklaşım ile bulanık grup TOPSIS yöntemi arasındaki karşılaştırmalı bir analiz, alternatiflere ve kriter değişikliklerine uygunluk, karar verme sürecindeki çeviklik, karar verici ağırlıklarının etkisi, ilk ve sonuncunun etkisi olmak üzere dört karşılaştırmalı değişkene dayalı olarak gerçekleştirilmiştir. Sonuçların sıralaması karşılaştırılan yöntemle aynı olduğundan önerilen yöntemin etkili olduğu görülmüştür.

IVHFE-ELECTRE yöntemi Gitinavard ve ark. (2018) tarafından sunulmuştur. Önerilen yöntemi doğrulamak için otomotiv endüstrisinde örnek olay çalışması yapılmıştır. Aday tedarikçiler çevresel yeterliliklerine göre değerlendirilmektedir. Yöntemin duyarlılığını ve geçerliliğini araştırmak için bir duyarlılık analizi, yazarlarca eklenmiştir. Sonuç olarak, değerlendiricilerin ağırlıklarının farklılaştırılmasının problemin sonucuna etkisinin güçlü olduğu, kriterlerin ağırlıklarının değişiminin ise duyarlı olduğu sonucuna varmışlardır.

Bolturk ve Kahraman (2018) yaptıkları çalışmada nütrosifik kümelerin alt sınıfı olan aralık değerli nütrosifik kümeleri baz alarak, AHS yöntemini genişleterek kosinüs benzerlik ölçümlerine dayanan aralık değerli nütrosifik AHS (IVN-AHS) yöntemini tanımlamışlardır. İkili karşılaştırma matrislerinde bu yöntemin objektif puanlama prosedürünü sağladığını öne sürerek, enerji alternatifi seçiminde bir uygulama ile çalışmalarını tamamlamışlardır.

Aralıklı nütrosifik Z sayılarına benzer olarak, Kutlu Gündoğdu ve Kahraman (2019), çalışmalarında genelleştirilmiş 3 boyutlu küresel bulanık kümeleri (SFS) tanımlamışlardır. Bu tanımlamalarını küresel bulanık uzaklıklara dayandırmışlardır. Sonrasında aralık değerli küresel bulanık ağırlıklı aritmetik ortalama ve aralık değerli bulanık geometrik ortalama hesaplamaları üzerinden TOPSIS kullanılarak 3D yazıcılarda çok kriterli seçim problemini çözmüşlerdir.

Jiang ve ark. (2019) tarafından yapılan bir çalışmada, gerçek uygulama alanlarında yaygın kullanılan tek değerli özel nütrosifik küme üzerinde aritmetik ortalamayı kullanarak, TOPSIS yöntemi ile pozitif ideal çözüm ve negatif ideal çözüm üzerinden değerlendirme yapmışlardır.

Bir başka örnek olarak, Vafadarnikjoo ve ark. (2020) yaptıkları çalışmada En İyi-Kötü Yöntemi (BWM), yöntemini nütrosifik genişletmesini en iyinin diğerlerine oranı (best-to-others) ve diğerlerinin en kötüye oranı (others-to-worst) değişkenlerini ekleyerek gerçekleştirmişlerdir.

Du ve ark. (2021) tarafından yapılan bir çalışmada NZN tanımlamalarına, aritmetik ortalama ile geometrik ortalama formülleri bir problem üzerinde uyarlanarak sonuçlar kıyaslanmıştır. Bu kıyaslamada normal NZN için tek bir değere odaklanıldığını güvenilirlik değerinin göz ardı edildiğini ifade etmişlerdir. Güvenilirlik değerinin kullanımının sonuçların bilgi olarak ifade derecesini ve güvenilirliğini artırmış olduğunu ifade ve alternatiflerin sıralamasının da değiştiğini tespit etmişlerdir.

Ye (2021), çalışmasında benzerlik ölçüsünün karar vermedeki önemine değinmiştir. NZN kümesinin benzerlik ölçüsünün olmamasından yola çıkarak, NZN kümeleri arasında uzaklık ve benzerlik ölçülerini kullanmış ve ağırlıklı genelleştirilmiş uzaklığı ile kosinüs ve kotanjant benzerlik ölçümlerini kıyaslamıştır.

Yong ve ark. (2021) çalışmalarında, yamuk nütrosofik sayıları ile nütrosofik Z sayılarını birleştirerek TrNZN (Trapezoid NZN), bu birleştirme sonuçlarının benzerlik sıralamasını yaparken ağırlıklı aritmetik ortalamalar ile ağırlıklı geometrik ortalamaları kullanmışlardır.

Taş ve Karataş (2021), yazılım sektöründe nitelikli personel seçimi probleminde, bilişim alanında hizmet veren firmalar için başarı ve sürdürülebilirlik açısından en doğru ve en uygun personel belirlenmesinin önemine değinmiş, nütrosofik AHS ile kriter ağırlıklarını belirlemiş ve nütrosofik TOPSIS ile de adayların sıralamasını yapmışlardır.

2.8. Bulanık AHS ve Nütrosofik Z Sayıları

Abdel-Basset ve ark. (2017) bir pozisyona başvuran adaylar arasından en iyisini seçmek için AHS yöntemini nütrosofik ortama aktararak üçgen nütrosofik sayılarla ikili karşılaştırmalarda kullanıcıların yanılgılarını azaltarak daha etkili sonuçlar elde ettiklerini belirtmişlerdir.

Abdel-Basset ve ark. (2018) bir firmada yapılan gerçek bir çalışmanın çıktılarını, AHS yöntemini nütrosofik alana taşıyarak model önerisi sunan SWOT analizini karşılaştırarak kendi modellerini doğruladıklarını belirtmişlerdir.

Kutlu Gündoğdu ve Kahraman (2020), yaptıkları çalışmada yenilenebilir enerji yer seçimi aşamasında klasik AHS yöntemini küresel bulanık AHS yöntemine genişleterek geçerlilik ve kabul edilebilirliği arttırmayı amaçlamışlardır. Burada üç boyutlu üyelik fonksiyonlarının daha bilgilendirici olduğu ve net bir şekilde ifade edilebileceği vurgulanmaktadır.

Ponce Ruiz ve ark. (2021), bilgi sistemlerinin iş süreçlerini basitleştirdiğini, maliyetleri azalttığını ve işlemler için daha güvenli olduğunu belirtmişlerdir. Bu bağlamda bilgi sistemlerinin stratejik planlamasında nütrosifik istatistikleri kullanmışlardır.

Molina Chalacan ve ark. (2021), BT hizmetlerinde doğru önceliklendirme ile yazılım geliştiricilerin ve/veya teknisyenlerin çalışma sürelerini optimize etmenin önemini vurgulamışlardır. Çalışmalarında Saaty'nin AHS yöntemini nütrosifik bir bakış açısıyla ağırlıklandırmışlardır.

2.9. Mevcut Model ile Diğerlerinin Eleştirel Analizi ve Motivasyon Noktaları

Karar verme problemlerinde bulanık ÇKKV yöntemlerinin kullanımı literatürde çokça yer almaktadır. Seçilen spesifik problem özelinde problemin alt başlıkları olan proje değerlendirme – geliştirme süreci değerlendirme – takım yetkinliklerini değerlendirme ve bu değerlendirmeler üzerinden de atama modeli ile atamaları kıyaslayan bir çalışma bu tezde birleştirilmiştir.

Bu çalışmanın ortaya çıkmasında bize ilham veren ve tetikleyen motivasyon kaynakları aşağıda sıralanmıştır;

- Özellikle yazılım geliştirme projelerinde gereksinimler üzerinde 5N1K modelini kullanan BOACube adlı bir çerçeve üzerinde çalışırken tespit edilen gereksinim tanımlarının proje başarısı ve anlaşılabilirliği üzerindeki yüksek etkisi.
- Literatür taramasına göre projelerin değerlendirilmesinde kullanılan kriterler, proje süreç yönetimi yönteminin seçimi gibi spesifik bir konuya göre dar bir kapsamda kullanılmaktadır.
- Yine literatür taramasında nütrosifik ve ÇKKV yöntemlerinin sıralama hesaplamalarında “mesafe” terimini temel aldığı görülmektedir.
- Yazılım geliştirme projelerinin başarı oranları, çevik veya geleneksel süreçlere ve proje boyutlarına göre düşüktür.
- Yel ve Engin (2022) bilgi teknolojileri alanında yönetsel bakış açısı ve metodolojik değişikliklerle gerçekleştirdikleri çalışmalarında yüksek verimlilik artışı elde etmişlerdir.
- Çizelge 2.6'daki eleştirilerde ve Çizelge 2.7'de içerik açısından değerlendirilen çevik – geleneksel yöntemlere ilişkin literatürdeki örneklerde,

- Geliştirme süreci seçiminde projelere bağlı olarak değerlendirmelerin yapılması.
- Daha sonra geliştirme süreci detayları temelinde böyle bir çalışmanın yapılması.
- Takımların değerlendirilmesi, atamaların yapılması, sonuçların karşılaştırılması ve verimliliğin hesaplanması.
- Tüm bu ögelerde bulanık ÇKKV yöntemlerinin bir kombinasyonunun bulunmaması.

dikkat çekmektedir.

Çizelge 2.6. Çevik ve geleneksel yöntemlere ilişkin literatür eleştirileri

Çalışma - Yıl	Detaylı Açıklama
Boehm ve Turner (2003)	Çalışmalarında çevik ve geleneksel metodolojileri alt kriterler bazında değerlendirmişlerdir. Radar şemasında beş farklı alanda proje özelliklerine göre hangi yöntemin seçilmesi gerektiğini görsel olarak ifade etmişlerdir. Çevik yaklaşımın risk yönetimi eksikliğini dengeleme çerçevesini devreye alıp gidermeye çalışarak değerli önerilerde bulunmuşlardır. Ancak geleneksel ve çevik metodolojilerin kırımını temel alan bir çalışma yapmadıkları gibi, proje ekibi görevlendirmeleri yoluyla da metodolojilerinin uygunluğunu ifade edemedikleri düşünülmektedir.
Begel ve Nagappan (2007)	Çevik metodolojilerin geliştirme, test etme ve yönetimde yer alan çalışanlar tarafından değerlendirilmesine dayanan bir çalışmadır. Bir firma çalışanları üzerinde yapılan ankete göre en popüler çevik yaklaşımın Scrum olduğunu belirlemişlerdir. Bu çalışmada gerçekleştirilen anket, çevik metodolojiye odaklanarak yalnızca çeviklik değerlendirilmiştir. Popüler SCRUM metodolojisi ve firmada kullanılan diğer metodolojiler üzerinden değerlendirme kriterleri bazında karşılaştırmalar yapmışlardır. Ancak çalışanların daha iyi moral, daha fazla verimlilik, daha iyi önceliklendirme, daha fazla kalite gibi kriterlerde düşük puanlar verme eğiliminde oldukları gözlemlenmiştir. Aynı zamanda çevik metodolojilerin büyük projelere uyarlanamaması ve çok fazla toplantı yapılması çalışanlar tarafından sorun olarak görülmüştür.
Mnkandla ve Dwolatzky (2007) / Mnkandla (2008)	Kapsamlı Çevik Metodolojiler (Generic Agile Methodologies, GAM) modeli olarak projeleri ve süreç yönetimi yöntemlerini matrisler üzerinden değerlendirerek, projelere atanacak takımlarda hem sosyal bağlanma hem de teknik odaklı olmak üzere iki aşamalı atamalar yapmışlardır.
Fitsilis (2008)	Yazılım geliştirme süreçleri kapsamında PMBoK ile Çevik Proje Yönetimini karşılaştırmıştır. Karşılaştırmayı süreç adımları detayında yapmıştır. Çevik metodolojilerin kapsam yönetimi, insan yönetimi, kalite yönetimi gibi alanlarda iyi olduğu ancak risk yönetimi, maliyet yönetimi ve satın alma yönetimi alanlarını karşılamadığı belirtmiştir.
Khan ve Beg (2013)	Her proje için farklı bir SDLC modeli seçilmesi gerektiğini vurgulamışlardır. Tek bir yaklaşımın tüm projelere uygulanamayacağını belirtmişlerdir. Yanlış SDLC modeli seçiminin yazılım geliştirme projeleri için çok kritik olduğu sonucuna varmışlardır. Karar destek matrisi ile en uygun SDLC modelinin seçimi için değerli bir sonuç ortaya çıkarmışlardır.
Şpundak (2014)	Proje yönetimi yaklaşımı ve metodolojisi ayrı ayrı ele alınarak iki farklı proje yönetimi metodolojisinin birleştirilebilirliği araştırılmıştır. Çevik ve geleneksel yaklaşımları proje ve organizasyon özellikleri bazında karşılaştırılmıştır. Ancak hibrit yaklaşım üzerinden takım atama ve değerlendirme gibi yaklaşımları esas alan herhangi bir çalışma yapılmamıştır.
Khan ve ark. (2014)	AHS ile SDLC modeli seçiminde yazılım projelerini ihtiyaç ve kriterler bazında değerlendirmişlerdir. Çevik - hızlı ve geleneksel yöntemleri seçmeye odaklanmışlardır.
Vijayarathay ve Butler (2015)	Organizasyonel projeler ve takımlar üzerinde yapılan anketin sonuçlarını özetlemişlerdir. Organizasyonel değerlendirme kapsamında, çok çalışanı olan büyük organizasyonların çoğunlukla geleneksel yöntemlerle proje yürüttüğü, az personele sahip organizasyonlarda çevik ve hibrit metodolojilerin kullanımına ilişkin bilgiler bulmuşlardır. Proje değerlendirme kapsamında çoğunlukla süreç yönetimi metodolojisinin kritik projelerde kullanımını değerlendirmişlerdir. Kritik projelerde geleneksel yönetimin kullanılmasının ağırlık taşıdığını belirtmişlerdir. Takım değerlendirmesinde, tek takımlı projelerde çoğunlukla çevik ve görsel modellerin kullanıldığı, 4 ve daha fazla ekibin olduğu projelerde ise geleneksel

	yöntemlerin kullanıldığı sonucuna varmışlardır. Böylece projelerin organizasyonel - proje ve takım bazlı özelliklerine göre çevik - geleneksel - yinelemeli ve hibrit metodolojilerin kullanımına ilişkin anket sonuçlarını yayınlamışlardır.
Farid ve ark. (2016)	Çalışma, CMMI ve Scrum metodolojilerinin CMMI orijinal yönetim süreç alanları üzerinden birbirleriyle ne derece örtüştüğünün matematiksel bir karşılaştırmasını içermektedir. Projenin özelliklerine göre uygunluk için bir eşleştirme matrisi kullanılmıştır. Scrum geliştirme süreci tedarikçi sözleşme yönetimi, risk yönetimi, bütünsel proje yönetimi ve nicel proje yönetimi süreç alanlarını kapsamadığını belirtmişlerdir. Scrum geliştirme sürecinin eksikliklerini gidermek için yeni bir model önermişlerdir. Bazı projelerin CMMI'ya, bazılarının ise Scrum sürecine uygun olabileceği bilinen bir gerçektir. Aslında süreç yönetimi metodolojisine farklı özelliklere göre odaklanmak daha etkili sonuçlar doğurabilirdi.
Salah ve ark. (2017)	Çalışma Scrum ve PMBoK aracılığıyla çok değerli sonuçlara sahip hibrit bir model önerilmiştir. İki farklı dinamik üzerinde çalışan yöntemin birleştirilmesi, esas olarak iki farklı metodolojinin ve proje türlerinin yönetiminde sorun yaşamasından kaynaklanmaktadır. Bu nedenle daha yönetsel ve proje odaklı bakış açısıyla hibrit bir modelin oluşturulmasının daha anlamlı olacağı düşünülmektedir.
Gencer ve Kayacan (2017a)	Çevik ve geleneksel metodolojileri farklı özellikler üzerinden karşılaştırarak, iki süreç yönetimi metodolojisinin farklılıkları görselleştirilmiştir. Süreç yönetimi metodolojileri açısından güncel olmasa da karşılaştırma kriterleri açısından detaylı ve anlamlı sonuçlara ulaşmışlardır.
Butler ve ark. (2020)	Proje dinamik ve karmaşıklığına ilişkin anketin sonuçlarını paylaşmışlardır. Projelerin dinamik yükseğe başarı için çevik metodolojilerin uygulanmasını önermişlerdir. Hangi yaklaşımın proje çıktılarını dayandırılması gerektiğini dinamizm ve karmaşıklık üzerinden ilişkilendirmişlerdir. Ancak süreç yönetimi metodolojilerine ayrıntılı olarak odaklanmamışlardır.
Adanna ve Nonyelum (2020)	Takım için süreç yönetimi metodolojisi bazında değerlendirmeler yapmışlardır. Aynı zamanda kullanıcı katılımına ilişkin değerlendirmeler yapmış ve proje türü, risk değerlendirmesi gibi yönetsel kriterleri eklemişlerdir. Bulanık arayüz değerlendirmesi ile hangi süreç geliştirme yönteminin kriterlere göre seçilmesi gerektiğine değinerek geniş bir kriter ağı üzerinden hangi süreç yönetim metodolojisinin hangi kriterlere cevap vereceğini tanımlamışlardır.
Gheorghe ve ark. (2020)	Çevik Manifestoda yer alan 4 ana prensip üzerinden on iki alt prensibi detaylandırmışlardır. Google Akademik'te 2001 öncesi ve sonrası çeviklik üzerine yapılan çalışmaların sayısı ile çevikliğin önemine vurgu yapmışlardır. Çevik yöntem ile ilgili aramalarda kullanılan 10 anahtar kelimenin kullanım sayılarını derlemişlerdir. Projede çalışan kişi sayısına ve inovasyon içeriğine göre süreç seçimi yapılabileceğini belirtmişlerdir.

Çizelge 2.7. İncelenen çevik ve geleneksel yöntem literatürünün özellik karşılaştırması

Çalışma - Yıl	Süreç Yönetimi Metodolojisine Odaklanma Detayı	Proje Bütçesi veya Başarısına İlişkin Yorumlar	Çevik-Geleneksel Metodoloji Kıyaslama	Süreç Yönetimi Metodolojisinin Proje Özellikleri Yoluyla Ele Alınması	Takım Değerlendirmesi	Proje - Takım Görevi	ÇKKV Yöntemlerinin Kullanımı
Boehm ve Turner (2003)	Hayır	Hayır	Evet	Evet	Hayır	Hayır	Hayır
Begel ve Nagappan (2007)	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır
Mnkandla ve Dwolatzky (2007) / Mnkandla (2008)	Evet	Evet	Evet	Evet	Hayır	Hayır	Hayır
Fitsilis (2008)	Evet	Hayır	Evet	Hayır	Hayır	Hayır	Hayır
Khan ve Beg (2013)	Evet	Hayır	Evet	Evet	Hayır	Hayır	Hayır
Şpundak (2014)	Hayır	Evet	Evet	Evet	Hayır	Hayır	Hayır
Khan ve ark. (2014)	Evet	Hayır	Evet	Hayır	Hayır	Hayır	Evet
Vijayasathay ve Butler (2015)	Evet	Hayır	Hayır	Evet	Evet	Hayır	Hayır
Farid ve ark. (2016)	Evet	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır

Salah ve ark. (2017)	Evet	Evet	Evet	Evet	Hayır	Hayır	Hayır
Gencer ve Kayacan (2017a)	Hayır	Evet	Evet	Evet	Hayır	Hayır	Hayır
Butler ve ark. (2020)	Hayır	Evet	Evet	Hayır	Hayır	Hayır	Hayır
Adanna ve Nonyelum (2020)	Evet	Hayır	Evet	Evet	Evet	Hayır	Hayır
Gheorghe ve ark. (2020)	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır
Önerilen Yöntem (NZN-Alan)	Evet	Evet	Evet	Evet	Evet	Evet	Evet

Çizelge 2.7'de görüldüğü gibi yönetsel yöntem değışikliđinin etkinliđinin ölçölmesi problemine bütünleşik bir yaklaşımla çözüm aranması gerekmektedir. Bu ihtiyaç doğrultusunda çalışmamızın katkıları aşağıda özetlenmiştir;

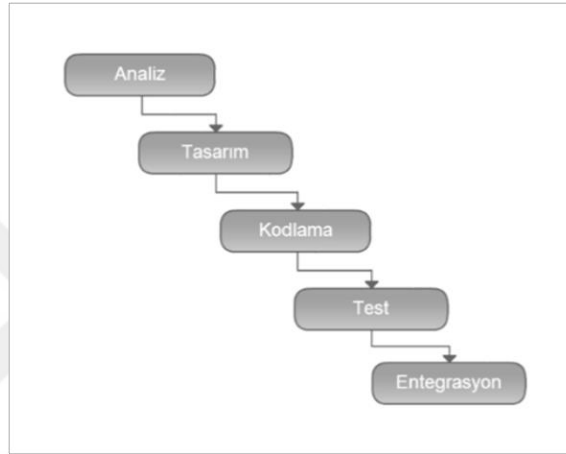
- Öncelikle yazılım geliştirme projelerinin yürütölmesinde yönetsel etki bulanık ÇKKV yöntemleriyle araştırılmıştır.
- Projeler, geliştirme süreçleri, takımlar değeriendirilerek takım atamaları ve atamalara ilişkin mevcut durum karşılaştırmalarına ilişkin sonuçlar elde edilmiştir.
- Bulanık ÇKKV yöntemlerinden NZN yöntemi genişletilerek, diđer bulanık ÇKKV yöntemlerinde olduđu gibi alternatiflerin "mesafe" ile sıralanması yerine "alan" hesabı üzerinden sıralama yapılmaktadır.
- Önerilen NZN-Alan yönteminin geçerliliđine ilişkin veri sonuçları, duyarlılık analizleri ve karşılaştırma analizleri ile elde edilmiştir.
- Karar vericileri olan her büyüklükteki organizasyon için geçerli yeni bir yöntemle proje – geliştirme süreci – takım atama şablonu önerilmiştir.

3. YAZILIM GELİŞTİRME SÜREÇLERİ

Bu bölümde araştırma sorusu üzerinden istenen çıktılarına ulaşmak için kullanılacak olan geliştirme süreçleri, çevik ve geleneksel olmak üzere iki ayrı başlık altında açıklanmaktadır.

3.1. Geleneksel Proje Geliştirme Süreçleri

3.1.1. Şelale süreci

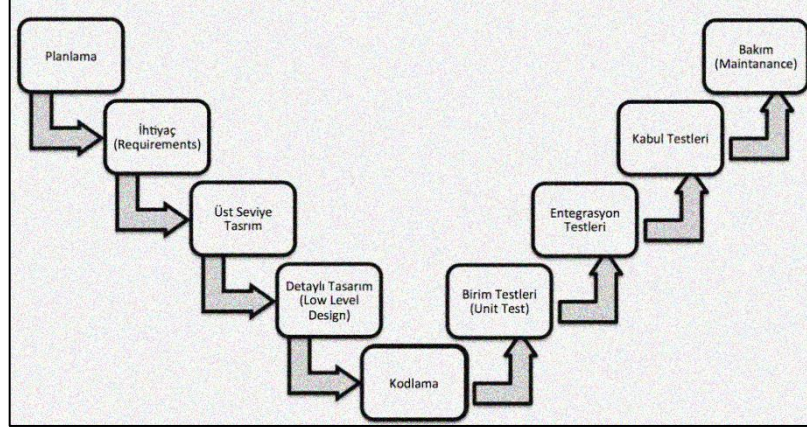


Şekil 3.1. Şelale modeli (Karayel, 2023)

Birçok çalışmada geleneksel ya da şelale geliştirme süreci olarak tanımlanan bu model, aşağı doğru akan yapısı ile tasarımın önceden netleştirilmesi gereken ve bilgi teknolojileri projelerinde kullanılan yöntemi ifade eder. Model içerisinde geri bildirim, SDLC aşamaları arasında sınırlıdır. Şelale modelinin yazılım geliştirme dünyasına getirdiği ilk özellik, aşamalar arası geri bildirimin kısıtlanması ile yeniden çalışma maliyetlerini kısıtlamasıdır. İkincisi ise SDLC sürecine prototip oluşturma yaklaşımını kazandırmasıdır (Boehm, 1988).

Şelale geliştirme süreci isminden ve Şekil 3.1’de yer alan örnekten yola çıkarak, şelale dizilimine benzeyen, sıralı tasarım sürecini içinde barındıran proje yönetim sürecidir (Sinha ve Das, 2021). Bir sonraki aşamanın başlaması için içinde bulunan aşamanın tamamlanması ve içinde bulunan aşamanın tamamlanması için de gereksinim tanımlamalarına uygun test senaryolarının doğrulanması gerekmektedir. Tek yönlü ürün olgunlaşması ve bilgi akışı söz konusudur.

3.1.2. V-Model süreci



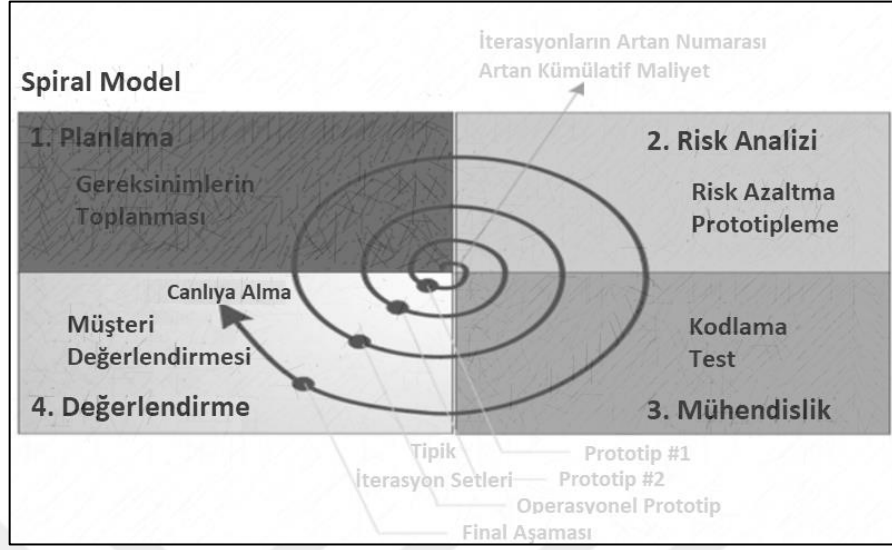
Şekil 3.2. V modeli (Seker, 2015)

Şelale geliştirme sürecinde olduğu gibi birbirini izleyen süreç adımlarının “V” şeklinde diziliminden oluşan modeldir (Şekil 3.2). Sonraki aşamanın başlaması için önceki aşamanın tamamlanması gerekmektedir. Test aşamalarına şelale geliştirme sürecinden daha fazla önem verilmektedir (Munassar ve Govardhan, 2010).

V’nin anlamı kabul (verification) ve doğrulama (validation) ifadelerinden gelmektedir. Bu sürecin, gereksinimlerin net olduğu küçük ve orta ölçekli projelerde uygulanması daha etkili olacaktır (Sarker ve ark., 2015). Kabul kriteri, gereksinimlerin müşteri tarafından kabulü olarak nitelendirilmektedir. Doğrulama adımları, üst seviye tasarım, detaylı tasarım ve kodlama aşamalarına karşılık gelen entegrasyon testleri ile birim testlerini ifade etmektedir.

Üst seviye tasarımdan, detay tasarımların oluşturulmasına kadar her aşamada kendi seviyesine karşılık gelen test senaryoları yürütüldüğünden şelale geliştirme sürecine nazaran daha başarılı sonuçlar alınabileceği ifade edilebilir.

3.1.3. Spiral model süreci



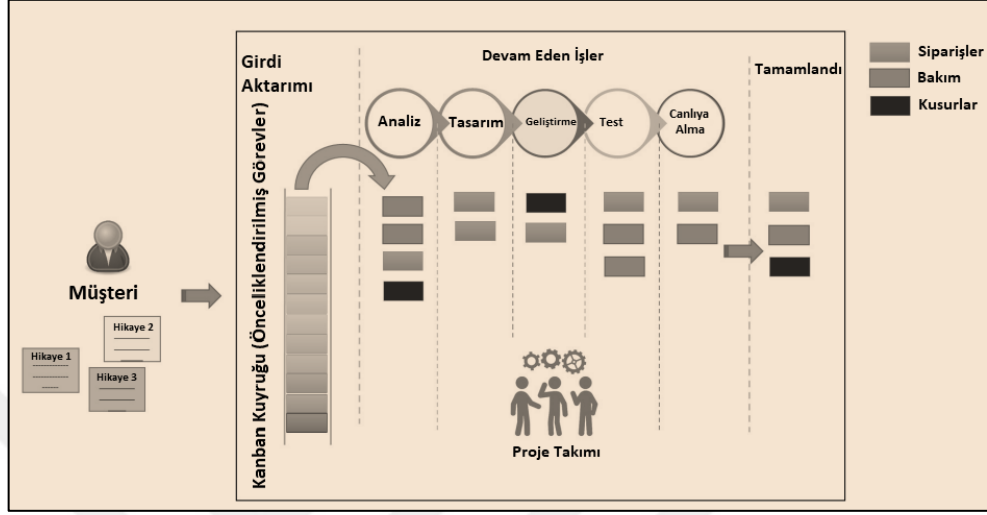
Şekil 3.3. Spiral model (Ahmad ve ark., 2016)

Bu süreç, büyük ölçekli kamu yazılım projelerinde şelale geliştirme sürecinin deneyimlenmesinden sonra geliştirilmiştir (Boehm, 1988). Spiral model, aşağıdan yukarıya, yukarıdan aşağıya elde edilen değerleri geliştirmek için tasarım ve ilk modellemenin farklı bileşenlerini birleştirmeye çalışan yazılım geliştirme sürecidir (Ahmad ve ark., 2016). Şekil 3.3'te gösterilen geliştirme sürecinin dört ana aşaması aşağıda özetlenmiştir (Highsmith ve Cockburn, 2001).

- **Planlama:** Bu aşamada gereksinimler toplanmakta ve sonraki aşamalar için somut amaç ve hedefler belirlenmektedir.
- **Risk Analizi:** Bu aşamada riski azaltmak için ilk modeller hazırlanmaktadır. Müşteri ile etkin iletişim kurularak büyük riskler tanımlanmaktadır. Tanımlanan bu riskler etkileri bakımından değerlendirilmeye tabi tutulmaktadır. Risklerin azaltılması için veriler toplanarak analiz edilmektedir.
- **Mühendislik:** Bu aşamada geliştirme maddeleri kodlanırken, planlama aşamasındaki amaç ve hedefler doğrultusunda testler yapılarak doğrulama aşaması gerçekleştirilmektedir.
- **Değerlendirme:** Geliştirilen model gözden geçirilerek sonraki spiral tur için yeniden planlama ve gereksinim toplama işlemleri yapılmaktadır. Böylece SDLC döngüsü de tamamlanmış olur.

3.2. Çevik Proje Geliştirme Süreçleri

3.2.1. Kanban süreci



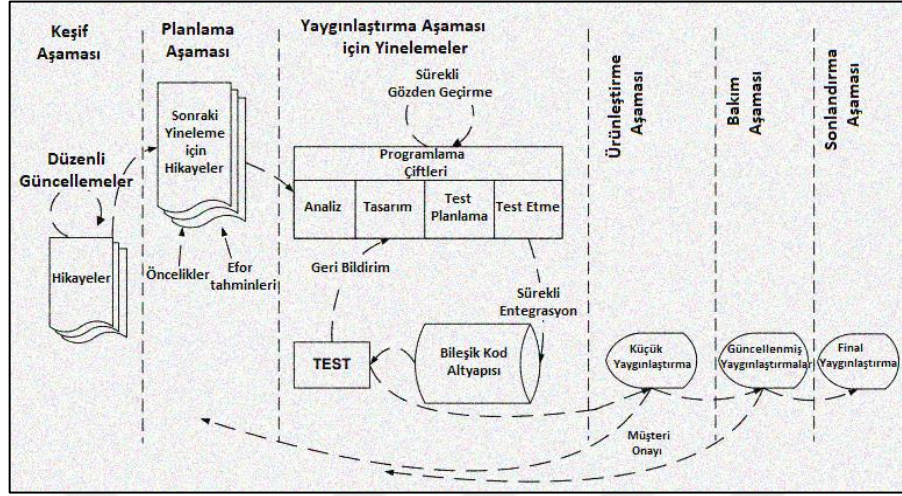
Şekil 3.4. Kanban süreci (Zayat ve Senvar, 2020)

Kanban geliştirme süreci yazılım geliştirme projelerine tam zamanında (Just-In-Time JIT) teslimatların yapılmasını amaçlayan proje geliştirme sürecidir. Doğru işin doğru zamanda yapılmasına odaklanmaktadır (Lei ve ark., 2017). Kanban, çalışanların mevcut üretim sorunlarını ve gelecek görevler hakkındaki farkındalığını netleştirmeye çalışmaktadır (Ikonen ve ark., 2011).

Kanban ile çalışmaya başlamak için, bir pano iş akışındaki aşamaları temsil edecek şekilde sütunlara bölünür. Ardından yapılacak işlere ait tanımları içeren kartlar, içinde oldukları aşamaları da temsil edecek şekilde bir sütundan diğerine alınarak ilerleme takip edilir (Zayat ve Senvar, 2020).

Kanban geliştirme sürecinin temel ilkeleri, görevlerin önceliklendirilmesi ve akış yönetimine yardımcı olacak şekilde işin görselleştirilmesi, yürütülen işlerin sınırlandırılmasıdır. Projede yapılacak olan görevler görselleştirilerek (Şekil 3.4) bütünsel bakış açısı ile anlaşılabilirlik artmaktadır (Ahmad ve ark., 2013).

3.2.3. Extreme Programming (XP) süreci



Şekil 3.6. XP Yaşam Döngüsü (Abrahamsson ve ark., 2017)

Basitçe söylemek gerekirse, XP, müşterinin belirttiği gereksinimlerin kodlara dönüştürülmesi ve bu kodların SDLC adımları arasındaki hareketlerinde gereksinimlere uygunlukları açısından test edilmesidir. XP bir ürünün zamanında müşteriye teslim edilmesine yönelik tasarlandığından, öngörülemez hiçbir yenilik veya özellik içinde bulunan süreç adımına dâhil edilmemektedir (Şekil 3.6). XP'nin arka planındaki fikir, geliştirme sürecinde daha sonra değişikliklere ihtiyaç duyulursa ve müşteri bunları geliştirme ekibine bildirirse bile, geliştiricilerin içinde bulunan yineleme adımıyla değişimler hakkında endişelenmemesidir. Kullanıcı gereksinimleri genellikle net ve belirlenmiş olmaktan ziyade muğlak ve değişken olabildiği için değişikliklerin yönetilememesi, geliştirici takım açısından kolayca karmaşaya dönüşebilmektedir (Erickson ve ark., 2005). XP geliştirme süreci bu karmaşanın gerçekleşmemesi için önlemleri sistemli şekilde sunmaktadır.

XP geliştirme süreci kısaca şu aşamalardan meydana gelmektedir; Birinci aşama olarak sistem mimarisi ile gereksinimlerin incelendiği “Keşif Aşaması” yer almaktadır. İkinci aşama olarak iş değeri üretebilecek “Ne yapılabilir?” sorusu ile “Sonraki yinelemede ne yapılmalı?” sorularına cevap aranan “Planlama Aşaması” gelmektedir. “Yaygınlaştırma aşaması için yinelemeler” aşaması, tasarım, kodlama, test ve bütünleşme gibi adımları bünyesinde barındırır. Akabinde geliştirmelerin küçük parçalar halinde yaygınlaştırıldığı “Ürünleştirme Aşaması” gerçekleştirilir. “Bakım Aşaması”nda, geliştirilen sistem koşturularak devam ederken sistemin olgunlaştırılması için gerekli küçük

geliřtirmeler yapılır. Son aşama olarak müşteri memnuniyeti sağlanmışsa, geliştirilecek özellik kalmamışsa ya da müşteri gereksinimleri verimli, ulaşılabilir veya yapılabilir olmadığı için projenin sonlandırılması gerekiyorsa, proje kapanışı için birkaç sayfa doküman hazırlanarak proje sonlandırılır. Bu aşama “Sonlandırma Aşaması” olarak isimlendirilir (Anwer ve ark., 2017).

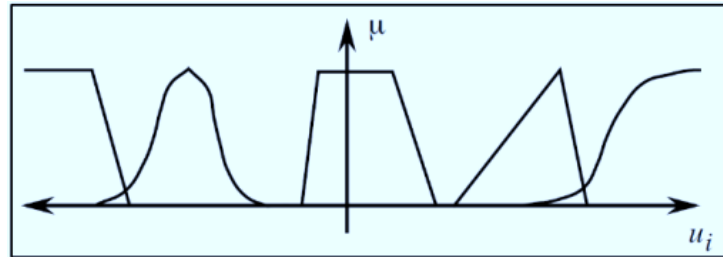


4. BULANIKLIK ÇOK KRİTERLİ KARAR VERME TEKNİKLERİ

Klasik küme teorisinde bir bileşen, bir kümeye üyedir “1” ya da değildir “0” şeklinde ifade edilmektedir. Zadeh bu kuramın üzerine, bir bileşenin “0” ile “1” arasında değer alabileceği üyelik fonksiyonu ile ifade edilen bulanıklık kavramını geliştirmiştir. Kendi cümleleri ile ifade edilecek olursak, “kesin olmayan şekilde tanımlanmış sınıfların insan düşüncesinde, soyutlamada, iletişimde önemli bir rol oynadığı” gerçeğine değinmiştir (Zadeh, 1965). Böylece Zadeh (2008), bulanık mantığın bulanık olmadığına vurgu yaparak, anlamsal ve öznel belirsizliği yönetebilecek olan bulanık küme teorisi ve bulanık mantığın kapısını aralamıştır.

Sınıflandırmalardan yola çıkarak sözel kavramların matematiksel ifadelerinde üyelik fonksiyonlarının kullanılması, süreklilik içeren üyelik fonksiyonları ile her değerlendirmenin “0 – 1” aralığında bir karşılığının olması gibi etkenler gerçek hayatta karşılaşılan uygulama ve mühendislik problemlerinde bulanık mantığın kullanılmasına olanak sağlamıştır. Bulanık mantık işlemleri bulanıklaştırma ve durulaştırma aşamalarını içeren iki adımdan oluşmaktadır.

Bulanıklaştırma aşamasında kullanılan üyelik fonksiyonu ile ilgili literatürde birçok yaklaşım bulunmaktadır. Bu yaklaşımlar arasında hangisinin daha etkin olduğu ya da birbirine karşı olan üstünlüklerinden söz etmek çok anlamlı değildir. Çünkü ele alınan problemdeki veri kümelerinin uygunluğu üzerinde bu değerlendirme öznel içerdiği için üstünlükten bahsetmek yersiz olacaktır. Üyelik fonksiyonları olarak sıklıkla üçgen, yamuk, çan eğrisi, Gauss ve işaret fonksiyonları kullanılmaktadır. Şekil 4.1’de bu fonksiyonlara ait görsel yer almaktadır.



Şekil 4.1. Üyelik fonksiyonları (Elmas, 2003)

Üyelik fonksiyonlarına ilişkin bir derleme çalışmasında (Çizelge 4.1), fonksiyonların tiplerine göre detay açıklamalara yer verilmiştir (Liu ve ark., 2020).

Çizelge 4.1. Üyelik fonksiyonlarının kıyaslaması

Bulanık Küme	Ne Zaman	Ne	Nasıl	Karmaşıklık
Üçgen Bulanık Sayı	Fikirler cevapları içerir: kısmen evet ve kısmen hayır.	Kesin üyelik ile belirsizlik net bir sayı olarak tanımlanır.	Üst ve alt sınırları ve orta noktayı tanımlanır.	Basit
Yamuk Bulanık Sayı			Üst ve alt sınırları ve iki orta nokta tanımlanır.	
Tip 2 Bulanık Küme: Yamuk Aralıklar	Fikirler oldukça belirsiz cevaplar içerir.	Kesin olmayan üyeliğe sahip net bir sayının belirsizliği tanımlanır.	Sırasıyla üst ve alt yamuk bulanık sayıların üst ve alt sınırları ve iki orta noktası tanımlanır.	Çok karmaşık
Sezgisel Bulanık Küme	Fikirler cevapları içerir: evet, hayır ve emin değilim.	Kesin üyelik ve kesin üye olmama durumuyla net bir sayının belirsizliği açıklanır.	Aidiyet ve ait olmama dereceleri tanımlanır.	Karmaşık

Durulaştırma aşamasında bulanık küme değerinin kesin bir değere ulaştırılması sağlanmaktadır. Bu adımda en çok ağırlık merkezi, kapsam analiz yöntemi, sınırların ortalaması, iyimserlik endeksi, bulanık entropi gibi yöntemlerden bahsedilebilir (Liu ve ark., 2020).

Literatür incelemeleri sonucunda bulanık kümeler ile ilgili olarak, bulanık küme değerinin kesin bir değere ulaştırılması ile aslında problemin doğasında olan belirsizliğin karar modeline eklenmesinin kısmi olduğu ifade edilebilir. Bu durumdan yola çıkarak daha güçlü bir teoriye olan ihtiyaçla hibrit kümeler üzerinden belirsizliğin karar modeline dâhil edilmesi araştırılmıştır. Böylece nütrosofi ismi ile anılan bir matematiksel model karar verme problemlerinde kullanılmak üzere literatürde yerini almıştır.

Bu açıklamalar rehberliğinde, literatürde en çok kullanılan ve tezin olgunlaşması aşamasında kullanılmış olan yöntemler arasından bulanık AHS, bulanık TOPSIS, bulanık EDAS, bulanık WASPAS ve NZN yöntemlerine dair başlıklar aşağıda yer almaktadır.

4.1. Bulanık AHS

Bulanık AHS yöntemi, literatürdeki birçok çalışmaya benzer şekilde, karar kriterlerinin ağırlıklandırılmasında kullanılmaktadır (Heidary Dahooie ve ark., 2018). Bu bağlamda Buckley (1985) tarafından kullanılan geometrik ortalama yöntemi birçok çalışmada çok kişili değerlendirmelerin birleştirilmesinde esas alınmaktadır. Yöntemin kullanım ve hesaplama sıralaması aşağıda adım adım yer almaktadır (Ulutaş ve ark., 2018).

- **Adım 1 :** Grup kararlarının kriterlere göre karşılıklarının bulunması,

- **Adım 2 :** İkili karşılaştırma matrisinin oluşturulması,
- **Adım 3 :** Geometrik ortalama denklemi ile kriterlere ait bulanık geometrik ortalamasının ve bulanık ağırlıkların (w_j) bulunması,
- **Adım 4 :** Bulanık ağırlıkların durulaştırılması (Denklem 4.1) ($l =$ düşük değer, $m =$ orta değer, $u =$ yüksek değer).

$$w_j = \frac{l(w_j)+m(w_j)+u(w_j)}{3} w_j = \frac{l(w_j)+m(w_j)+u(w_j)}{3}, j = 1,2,...,m \quad (4.1)$$

- **Adım 5 :** Bulunan değerlerin normalleştirilmesi (Denklem 4.2).

$$w_j^* = \frac{w_j}{\sum_{j=1}^m w_j}, j = 1,2,...,m \quad (4.2)$$

4.2. Bulanık TOPSIS

Literatürde bulanık AHS ile birlikte ÇKKV problemlerinde alternatiflerin sıralanmasında sıkça kullanılan yöntemlerden bir diğeri de bulanık TOPSIS yöntemidir. Özellikle alternatiflerin pozitif ideale yakınlığı ile negatif ideale uzaklığı üzerinden sıralama yaptığı için güçlü bir sıralama yöntemi olduğu söylenebilir.

Chen (2000), bir yazılım firması için sistem analizi yapan mühendis seçiminde bulanık TOPSIS'i kullanmıştır. Aynı çalışmadan alınan ve yöntem hakkında genel bir bilgi veren hesaplama adımları aşağıda yer almaktadır;

- **Adım 1 :** Alternatiflerin dilsel ölçekler üzerinden değerlendirilmesi.
- **Adım 2 :** Bulanık karar matrisinin normalleştirilmesi (Denklem 4.3) (A fayda kriterini, B ise maliyet kriterini ifade etmektedir).

$$r_{ij} = \left(\frac{a_{ij}}{c_j^*}, \frac{b_{ij}}{c_j^*}, \frac{c_{ij}}{c_j^*} \right), j \in A : r_{ij} = \left(\frac{a_j^-}{c_{ij}}, \frac{a_j^-}{c_{ij}}, \frac{a_j^-}{c_{ij}} \right), j \in B \quad (4.3)$$

$$c_j^* = \max c_{ij}, IF j \in A : a_j^- = \min a_{ij}, IF j \in B \quad (4.4)$$

- **Adım 3 :** Normalleştirilmiş karar matrisinin ağırlıklandırılması (ağırlıklarla çarpılması) (Denklem 4.5).

$$v_{ij} = r_{ij} \times w_j \quad (4.5)$$

- **Adım 4 :** Bulanık pozitif ve bulanık negatif ideal çözümlerin hesaplanması (Denklem 4.6).

$$v_j^* = (1, 1, 1, \dots) : v_j^- = (0, 0, 0, \dots) \quad (4.6)$$

- **Adım 5 :** Yakınlık oranının hesaplanması (Denklem 4.7).

$$d_j^* = \sum_{i=1}^n d(v_{ij}, v_j^*) : d_j^- = \sum_{i=1}^n d(v_{ij}, v_j^-) : i=1, 2, \dots, m \quad (4.7)$$

$$CC_i = \frac{d_i^-}{d_i^* + d_i^-}$$

- **Adım 6 :** Alternatiflerin yakınlık oranına göre sıralanması.

4.3. Bulanık EDAS

Bulanık EDAS yöntemi ilk olarak Keshavarz Ghorabae ve ark. (2015) tarafından envanter sınıflandırmasında kullanılmıştır. Bulanık EDAS yöntemi yeni olmasına rağmen birçok çalışmada kullanılarak geçerliliğini ve kıyaslamalardaki etkililiğini kanıtlamıştır.

Aşağıdaki adımlar, Stević ve ark. (2019) tarafından tanımlanmış olan 8 temel hesaplama adımını ifade etmektedir;

- **Adım 1:** Ortalama karar matrisinin oluşturulması (Denklem 4.9),

$$X = [\check{x}_{kl}]_{n \times m}, \text{ 'k' alternatifleri 'l' ise kriterleri ifade etmektedir.} \quad (4.8)$$

$$\check{x}_{kl} = \frac{1}{a} \oplus_{i=1}^a \check{x}_{kl}^i, (i=1, 2, \dots, a : \text{karar verici}) \quad (4.9)$$

- **Adım 2:** Kriter ağırlık matrisinin oluşturulması (Denklem 4.10),

$$\tilde{w}_l = \frac{1}{a} \oplus_{i=1}^a \tilde{x}_l^i, \text{ 'w' kriter ağırlıklarını ifade etmektedir.} \quad (4.10)$$

- **Adım 3:** Ortalama çözüm matrisinin oluşturulması (Denklem 4.11),

$$\tilde{av}_l = \frac{1}{n} \oplus_{k=1}^n \tilde{x}_{kl}, \text{ 'av' ortalama çözüm değerini ifade etmektedir.} \quad (4.11)$$

- **Adım 4:** Fayda veya maliyet kriterlerini dikkate alarak, ortalamadan pozitif uzaklık (PDA) (Denklem 4.12, 4.14) ve ortalamadan negatif uzaklık (NDA) (Denklem 4.13, 4.15) değerlerinden oluşan matrislerin belirlenmesi;

$$PDA = [\tilde{pda}_{kl}]_{n \times m}, \quad (4.12)$$

$$NDA = [\tilde{n da}_{kl}]_{n \times m}, \quad (4.13)$$

$$\tilde{pda}_{kl} = \begin{cases} \frac{\psi(\tilde{x}_{kl} - \tilde{av}_l)}{k(\tilde{av}_l)}, & \text{eğer } k \in \text{Fayda}; \\ \frac{\psi(\tilde{av}_l - \tilde{x}_{kl})}{k(\tilde{av}_l)}, & \text{eğer } k \in \text{Maliyet}; \end{cases} \quad (4.14)$$

$$\tilde{n da}_{kl} = \begin{cases} \frac{\psi(\tilde{x}_{kl} - \tilde{av}_l)}{k(\tilde{av}_l)}, & \text{eğer } k \in \text{Maliyet}; \\ \frac{\psi(\tilde{av}_l - \tilde{x}_{kl})}{k(\tilde{av}_l)}, & \text{eğer } k \in \text{Fayda}; \end{cases} \quad (4.15)$$

- **Adım 5:** Tüm alternatifler için pozitif (sp_k) (Denklem 4.16) ve negatif (sn_k) (Denklem 4.17) mesafelerinin ağırlıklı toplamalarının hesaplanması;

$$sp_k = \oplus_{l=1}^m (\tilde{w}_l \otimes \tilde{pda}_{kl}) \quad (4.16)$$

$$sn_k = \oplus_{l=1}^m (\tilde{w}_l \otimes \tilde{n da}_{kl}) \quad (4.17)$$

- **Adım 6:** Tüm alternatifler için pozitif mesafe (Denklem 4.18) ve negatif mesafe (Denklem 4.19) normalleştirilmiş değerlerinin hesaplanması;

$$\tilde{ns p}_k = \frac{sp_k}{\max_k(k(\tilde{sp}_k))}, \quad (4.18)$$

$$\tilde{ns n}_k = 1 - \frac{sn_k}{\max_k(k(\tilde{sn}_k))} \quad (4.19)$$

- **Adım 7:** Tüm alternatifler için değerlendirme puanının ($\tilde{a}s_k$) (Denklem 4.20) hesaplanması.

$$\tilde{a}s_k = \frac{1}{2}(\tilde{n}\tilde{s}p_k \otimes \tilde{n}\tilde{s}n_k) \quad (4.20)$$

- **Adım 8:** Tüm alternatiflerin değerlendirme puanına ($\tilde{a}s_k$) göre azalan sırada sıralanması ve en iyi aday çözümün belirlenmesi.

4.4. Bulanık WASPAS

WASPAS yöntemi ilk olarak Zavadskas ve ark. (2012) tarafından kullanılmıştır. Model, ağırlıklı toplama modeli ile ağırlıklı çarpım modellerinin birleşimidir. Model kendi içerisinde farklı değerlere ait ağırlıklı toplama ve çarpım sonuçlarını birleştirdiği için sıralamalarda kullanılan bir diğer yöntem olarak literatürde ağırlık kazanmıştır.

Bu çalışmada WASPAS modeli olarak bulanık tabanlı WASPAS yöntemi kullanılmıştır. Hesaplama detayları, Turskis ve ark. (2015) tarafından yapılan çalışmaya dayanmaktadır. Hesaplama adımları aşağıda kısaca açıklanmaktadır;

- **Adım 1:** Bulanık karar matrisinin hazırlanması (Denklem 4.21),

$$X = [\tilde{x}_{kl}]_{n \times m}, \text{ 'k' alternatifleri 'l' ise kriterleri ifade etmektedir.} \quad (4.21)$$

- **Adım 2:** Matrisin tüm elemanları için başlangıç değerlerinden normalleştirilmiş karar matrisinin oluşturulması (Denklem 4.22),

$$\tilde{\tilde{x}}_{kl} = \begin{cases} \frac{\tilde{x}_{kl}}{\max_k \tilde{x}_{kl}}, & \text{eğer } \max_k \tilde{x}_{kl} \text{ tercih edilirse,} \\ \frac{\min_k \tilde{x}_{kl}}{\tilde{x}_{kl}}, & \text{eğer } \min_k \tilde{x}_{kl} \text{ tercih edilirse,} \end{cases} \quad (4.22)$$

- **Adım 3a:** Ağırlıklı Toplama Yöntemi için ağırlıklı normalleştirilmiş bulanık karar matrisinin hesaplanması ($\tilde{\tilde{X}}_q = [\tilde{\tilde{x}}_{kl}]_{n \times m}$),

- **Adım 3b:** Ağırlıklı Çarpım Yöntemi için ağırlıklı normalleştirilmiş bulanık karar matrisinin hesaplanması ($\tilde{X}_p = [\tilde{x}_{kl}]_{n \times m}$),
- **Adım 4:** 3a’da ($\tilde{Q}_k$) ve 3b ($\tilde{P}_k$)’de hesaplanan (Denklem 4.23, 4.24) matrisler için optimallik fonksiyon değerlerinin hesaplanması (Denklem 4.25, 4.26);

$$\tilde{Q}_k = \sum_{k=1}^m \tilde{x}_{kl}, k=1,2,\dots,m \quad (4.23)$$

$$\tilde{P}_k = \prod_{l=1}^n \tilde{x}_{kl}, k=1,2,\dots,n \quad (4.24)$$

$$Q_k = \frac{1}{3}(Q_{k\alpha} + Q_{k\beta} + Q_{k\gamma}), \quad (4.25)$$

$$P_k = \frac{1}{3}(P_{k\alpha} + P_{k\beta} + P_{k\gamma}), \quad (4.26)$$

- **Adım 5:** Alternatiflerin hepsini kapsayacak şekilde bulanık WASPAS için optimallik fonksiyon değerlerinden türetilen sabit (Denklem 4.27) kullanılarak, birleştirilmiş fayda fonksiyonunun hesaplanması (Denklem 4.28),

$$\lambda = \frac{\sum_{k=1}^m P_k}{\sum_{k=1}^m Q_k + \sum_{k=1}^m P_k}, \quad (4.27)$$

$$K_k = \lambda \sum_{l=1}^m Q_k + (1 - \lambda) \sum_{l=1}^m P_k, \lambda = 0, \dots, 1; \quad 0 \leq K_k \leq 1, \quad (4.28)$$

- **Adım 6:** Alternatifler fayda fonksiyon değerine göre sıralanır ve maksimum (K_k) değeri olan birisi seçilir.

4.5. Nötrosifik Z Sayı Kümeleri (NZN)

Ye (2021), çalışmasında belirsiz değer “U” bulanık değerini “N” ve bulanık değer “U” üzerinden güvenilirlik ölçüsü olan “M” değerini alarak Z sayılarını nötrosifik Z sayıları olarak genişletmiştir. Tanımlamada mevcut yöntemlerin hata payından yoksun olduğu ifade edilmiştir. Daha sonra karar verme aşamasına benzerlik ölçümünde iki nokta arasındaki uzaklık denklemi olarak kosinüs ve kotanjant denklemlerini tanımlamış ve uygulama üzerinde farklılıklarını gözlemlemiştir. Bu çalışmada mesafe hesaplamaları için Ye (2014) tarafından yapılan çalışmadan yararlanılmıştır.

NZN karar metodolojisinin kullanılmasının ve ölçeklendirilmiş dilsel değişkenlere dayalı hesaplamaların avantajları açıktır. NZN yöntemi, içerdiği Doğruluk (T) – Belirsizlik (I) – Yanlışlık (F) değişkenleri ile değerlendirmede kullanılan dilsel

değişkenlere karşılık gelen bulanık ölçekler açısından güçlü bir temsil yeteneğine sahiptir. Diğer bulanık yöntemler ile karşılaştırıldığında kendi hesaplamalarında güvenilirlik hesaplamaları ve bulanık değer – belirsiz değer gibi değişkenleri kullanması nedeniyle bu çalışmada problemin çözümü için uygun olduğu düşünülmektedir. Seikh ve Mandal (2023) aralık değerli bulanık kümeden (Interval Valued Fuzzy Set, IVFS) türetilen aralık değerli Fermatean bulanık küme (Interval Valued Fuzzy Fermatean Set, IVFFS) yönteminin, karmaşık problemlerin modellenmesi ve çözümü için daha uygun olduğunu belirtmişlerdir. Yazılım geliştirme projelerinin doğru takımlar aracılığıyla, doğru süreç yönetimi metodolojisiyle geliştirilmesi sorununun da karmaşık olduğu aşikârdır. Mandal ve Seikh (2023), başka bir çalışmada aralık değerli küresel bulanık (Interval Valued Spherical Fuzzy, IVSF) kümeler için kesin olmayan bilgileri daha iyi karşıladığını belirtmiştir. Bu bakımdan T-I-F için üç boyut üzerinden NZN hesaplamalarının kesin olmayan bilgileri karşılama açısından yine güçlü olduğu söylenebilir. Ye (2021)'nin çalışmasında önerdiği yöntemin mevcut nütrosifik karar verme yöntemlerine göre daha gelişmiş olduğunu ve karar vericilerden alınan bilgilerin güvenilirliğini artırdığını da vurgulamıştır. Broumi ve ark. (2019), üçgen ve yamuk sayıların gerçek dünya uygulamalarında daha basit hesaplama adımlarına sahip olduğunu belirtmişler, üçgen ve yamuk nütrosifik sayıları kullanmışlardır. Ek olarak, Bolturk ve Kahraman (2018) nütrosifik sayıların bir arada kullanıldığı bulanık yöntemleri derleyerek özellikle 2013 – 2018 yılları arasında yıllar bazında değerlendirmeler yapmışlar ve nütrosifik sayılar üzerine yapılan çalışmaların sayısının 2016 ve 2017 yıllarında arttığını belirtmişlerdir.

Bu bağlamda, NZN'nin en yaygın kullanılan bulanık ÇKKV yöntemlerinden bulanık EDAS ve bulanık WASPAS ile karşılaştırılmasını içeren Yel ve Baysal (2023) çalışmasında, NZN'nin, bulanık EDAS ve bulanık WASPAS'tan daha iyi sonuçlar verdiğini ifade etmişlerdir. Aralık değerli Nütrosifik Z sayıları, belirsizliği ortadan kaldırması, karmaşık sorunları çözmesi ve karar vericilerin esnek bir değerlendirme yapmasına olanak sağlaması nedeniyle tercih edilmektedir. Ayrıca Tavakkoli-Moghaddam ve ark. (2015) aralık değerli üyelik derecelerinin değerlendirme hatalarını azaltabileceğini bulmuşlardır.

Yukarıda belirtilen kaynaklar ve bilgiler kapsamında, nütrosifik Z sayılarının karar verme problemlerinde kullanılması ve ölçeklenmiş dilsel değişkenler üzerinden hesaplamalara tabi tutulmasının faydaları açıkça görülmektedir. Bu çalışmada Ye (2021), tarafından yapılan çalışmaya dayalı olarak, nütrosifik Z sayılarına genişletilmiş Z sayıları

olan hem belirsizliğin bulanık değerleri “N”, “U” hem de güvenilirlik değeri “M” Z sayıları, hesaplamada kullanılmıştır. Ye'nin (2021) çalışmasında tanımladığı sıra-uzaklık hesaplama formülü aşağıda verilmiştir;

- **Adım 1:** Nötrosofik Z sayıları kümesi aşağıdaki tanıma göre oluşturulmuştur. Burada doğruluk T, belirsizlik I ve yanlışlık F ile temsil edilir (Denklem 4.29).

$$S_z = \{ \langle u, T(N, M)(u), I(N, M)(u), F(N, M)(u) \rangle | u \in U \}, \quad (4.29)$$

- **Adım 2:** Nötrosofik Z sayılarının kriterler bazında değerlendirilmesi ve pozitif ideal çözüme uzaklığı aşağıdaki denklemler ile kriter ağırlıkları toplanarak hesaplanır ($v_k \in [0,1]$, $\sum_{k=1}^n v_k = 1$) (Denklem 4.30, 4.31).

$$MD_{wp}(S_{z1}, S_{z2}) = 1 - D_{wp}(S_{z1}, S_{z2}), \quad (4.30)$$

$$MD_{wp}(S_{z1}, S_{z2}) = 1 - \frac{1}{2} \left\{ \left[\frac{1}{3} \sum_{k=1}^n v_k (|TN_{1k} - TN_{2k}|^p + |IN_{1k} - IN_{2k}|^p + |FN_{1k} - FN_{2k}|^p) \right]^{1/p} + \left[\frac{1}{3} \sum_{k=1}^n v_k (|TM_{1k} - TM_{2k}|^p + |IM_{1k} - IM_{2k}|^p + |FM_{1k} - FM_{2k}|^p) \right]^{1/p} \right\}, \quad (4.31)$$

5. ÖNERİLEN YÖNTEMİN TANIMI

5.1. NZN-Alan Yönteminin Ayrıntıları

(4.30) ve (4.31) denklemlerinden yola çıkarak (5.1) ve (5.2) denklemlerini ayırtırdığımızda uzaklık değerlerinin bulunması yerine alan değerlerinin bulunmasını sağlayacak denklemi (5.3) elde etmiş oluruz.

Uzaklık ölçümü:

$$\left[\frac{1}{3} \sum_{k=1}^n v_k (|TN_{1k} - TN_{2k}|^p + |IN_{1k} - IN_{2k}|^p + |FN_{1k} - FN_{2k}|^p) \right]^{1/p}, \quad (5.1)$$

Güvenilirlik ölçümü:

$$\left[\frac{1}{3} \sum_{k=1}^n v_k (|TM_{1k} - TM_{2k}|^p + |IM_{1k} - IM_{2k}|^p + |FM_{1k} - FM_{2k}|^p) \right]^{1/p}, \quad (5.2)$$

$$\text{Elips Alanı(EA)} = \pi AB = \pi \frac{(5.1)}{2} \frac{(5.2)}{2}, \pi = 3.14, \quad (5.3)$$

Denklem (5.3) incelendiğinde, nütrosifik Z sayılarının belirsizlik ve güvenilirlik değerlerini tek boyutlu uzaklık değerine indirgeyerek sıralama yapmak yerine, elips alanının belirlenmesiyle elde edilecek ideal pozitif çözümün, aşağıdaki açıklama ve gerekçelerle daha uygun sonuçlar vereceği düşünülmüştür. Tüm hesaplamalarda “p” tamsayı değeri 1 olarak alınmıştır.

Farklı anlamlara sahip ancak birbirini destekleyen iki veri grubu üzerinden karar verildiğinde, bu iki değer karşılaştırılarak çarpım-bölüm gibi değerlendirmelerle sıralanması birçok karar verme yönteminde kullanılmaktadır. Örneğin TOPSIS yönteminde sonuç, bir alternatifin negatif ideale uzaklığı ve pozitif ideale yakınlığı karşılaştırılarak elde edilmektedir. Benzer şekilde VIKOR yönteminde çarpma ve/veya bölme ile sıralamada kullanılacak veriler hesaplanmaktadır. Bulanık “U” değeri ile NZN yönteminde kullanılan pozitif ideal değer arasındaki fark ve bu farkın güvenilirlik değerinin aritmetik ortalaması ile sıralamalara ulaşmak çok yönlü değerlendirme değişkenleri (doğruluk, belirsizlik, yanlışlık) açısından eksik olabilir. Bu varsayıma göre, NZN-Alan yönteminde elips alan hesaplamasının kullanılmasına karar verilmiştir.

Karar verme sürecinde, gerçek değer farkı ve güvenilirlik gibi iki farklı değişken kullanımının yanı sıra, NZN ve NZN-Alan yöntemlerine dayalı deneysel sonuçların

doğru olacağı öngörüsüyle elips alan hesaplamaları yapılmıştır. Elips alan hesaplamaları, iki farklı değişkenin birbirleri üzerindeki baskınlıkları ve birbirleri üzerindeki etkileri dikkate alınarak yapılmıştır. Denklem (5.1)'in belirsizlik değişkeni olduğu ve bu belirsiz değişkenin ideal “N” değerinden uzaklığı hesaplanmaktadır. Denklem (5.2)'de ise güvenilirlik değerinin ideal “M” değerinden uzaklığı hesaplanmaktadır. NZN yöntemi için uzaklıkların değerlerinin aritmetik ortalaması alınıp "1" sayısından çıkarılarak ve en yüksek değere sahip alternatif birinci sıraya yerleştirilir.

Bu çalışma ile iki farklı anlama gelen ve birbirine bağımlı olan denklemlerin N ve M değerlerine göre uyarlanması ile elde edilen sonuçlarda farklılık olup olmadığı araştırılmıştır. Ayrıca üç farklı yaklaşımla alan tabanlı hesaplamaların etkinliğinin belirlenmesi amaçlanmıştır.

Birinci Yaklaşım: Elips Alan Modeli

$$Alan_{elips} = \left\{ \begin{array}{l} (5.1), (5.2) = 0 \\ (5.2), (5.1) = 0 \\ \pi \frac{(5.1)}{2} \frac{(5.2)}{2}, (5.1) > 0 \text{ ve } (5.2) > 0 \end{array} \right\} \pi = 3.14, \quad (5.4)$$

Elips alan modeli ve denklem (5.4) ile ilgili açıklamalar;

- Güvenilirlik değerinin uzaklığı sıfır ise, sıralamada belirsizlik değerinin uzaklığı kullanılmalıdır.
- Belirsizlik değerinin uzaklığı sıfır ise sıralamada güvenilirlik değerinin uzaklığı kullanılmalıdır.
- Her iki değer de sıfırdan büyükse;
 - Birbirinin etki alanları üzerinden belirsizlik mesafesi veya güvenilirlik mesafesi sıfıra yaklaştıkça bağımlı değişken oldukları için birbirleri üzerindeki etki dereceleri azalan bir yapıda olacaktır.
 - Bu nedenle, azalan etkinin sonuçlara yansımaları elips alanı ile ifade edildiğinde gerçeği daha iyi yansıtacaktır.

İkinci Yaklaşım: Dikdörtgen Alan Modeli

$$Alan_{dikdörtgen} = \left\{ \begin{array}{l} (5.1), \text{ eğer } (5.2) = 0 \text{ ise,} \\ (5.2), \text{ eğer } (5.1) = 0 \text{ ise,} \\ (5.1)(5.2), \text{ eğer } (5.1) > 0 \text{ ve } (5.2) > 0 \text{ ise} \end{array} \right\}, \quad (5.5)$$

Dikdörtgen alan modeli ve denklem (5.5) ile ilgili açıklamalar;

- Güvenilirlik değerinin uzaklığı sıfır ise, sıralamada belirsizlik değerinin uzaklığı kullanılmalıdır.
- Belirsizlik değerinin uzaklığı sıfır ise, sıralamada güvenilirlik değerinin uzaklığı kullanılmalıdır.
- Her iki değer de sıfırdan büyük ise sıralamalarda kullanılacak hesaplama dikdörtgen alan hesaplama formülü olmalıdır.

Üçüncü Yaklaşım: Hibrit Alan Modeli

$$Alan_{hibrit} = \left\{ \begin{array}{l} (5.1), \text{ eğer } (5.2) = 0 \text{ ise,} \\ (5.2), \text{ eğer } (5.1) = 0 \text{ ise,} \\ (5.1)(5.2), \text{ eğer } (5.1) > 0 \text{ ve } (5.2) > 0 \text{ ve } (5.1) > (5.2) \text{ ise,} \\ \pi \frac{(5.1)(5.2)}{2}, \text{ eğer } (5.1) > 0 \text{ ve } (5.2) > 0 \text{ ve } (5.1) \leq (5.2) \text{ ise} \end{array} \right\}, \pi = 3.14, \quad (5.6)$$

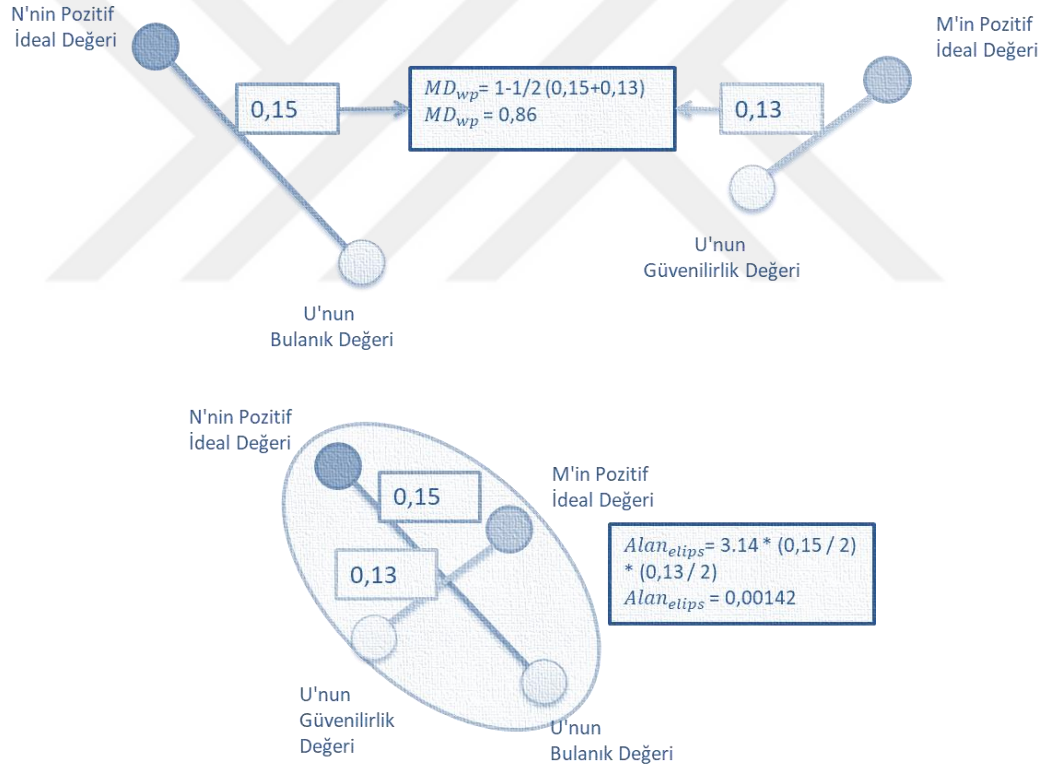
Hibrit alan modeli ve denklem (5.6)'ya ilişkin açıklamalar;

- Güvenilirlik değerinin uzaklığı sıfır ise, sıralamada belirsizlik değerinin uzaklığı kullanılmalıdır.
- Belirsizlik değerinin uzaklığı sıfır ise sıralamada güvenilirlik değerinin uzaklığı kullanılmalıdır.
- Denklem 5.6'da yer alan hesaplama, güvenilirlik mesafesinin belirsizlikten küçük olduğu durumlarda belirsizlik mesafe değeri ile güvenilirlik mesafe değeri üzerinden dikdörtgen yaklaşımının kullanılmasını önermektedir. Tam tersi durumda ise elips alan hesabı üzerinden alan değeri hesaplanmalıdır.

$$Alan = \left\{ \begin{array}{l} (5.1)^2, \frac{d(5.1)}{d(5.1)} = 2(5.1), \text{ eğer } (5.1) > (5.2) \text{ ise,} \\ \pi \frac{(5.1)(5.2)}{2}, \frac{d(5.1)}{d(5.1)} = \pi(5.2), \text{ eğer } (5.1) \leq (5.2) \text{ ise} \end{array} \right\}, \pi = 3.14, \quad (5.7)$$

- Denklem (5.7) incelendiğinde, belirsizliğin kendine göre türevi alınarak hesaba katılması, yeni bir alan hesabı ortaya koymaktadır. Bu durumda güvenilirliğin etkisi, belirsizliğin etkisinin yerini almaktadır.
- Her iki değer de sıfırdan büyük ve güvenilirliğe olan uzaklık belirsizliğe olan uzaklığa eşit veya büyükse, baskın değişken güvenilirlik mesafesidir. Güvenilirliğin etkisinin belirsizliğin uç noktalarında azaldığı varsayılır. Uç noktalarda azalana güvenilirlik etkisiyle sıralamalarda elips alan denkleminin kullanılması gerekir.

Bu açıklamalardan hareketle, Gündoğdu ve Kahraman (2019) çalışmasında bahsedilen geometrik anlamda yöntemlerin karşılaştırma grafiğinden esinlenerek Şekil 5.1'deki $Alan_{elips}$ için önerilen hesaplama mantığı açıklanmıştır.



Şekil 5.1. Önerilen alan hesaplaması ve mantığının açıklaması

Şekil 5.1 incelendiğinde belirsizlik ve güvenilirlik açısından ideal çözüme yakınlığın sıralanmasında tek boyutlu çözüm yerine belirsizlik ve güvenilirlik değerlerinin çarpımı için oluşturulan alan değerinin kullanılması daha kapsayıcı ve tutarlı sonuçların üretilebilmesi bakımından faydalı olmuştur. Kısaca ideal pozitifte yakınlık

derecesine bağılı olarak güvenilirlik değeri nin kapsadığı alanın sıralama kriteri olarak kullanılmasının daha anlamlı sonuçlar verdiğı düşünölmektedir.

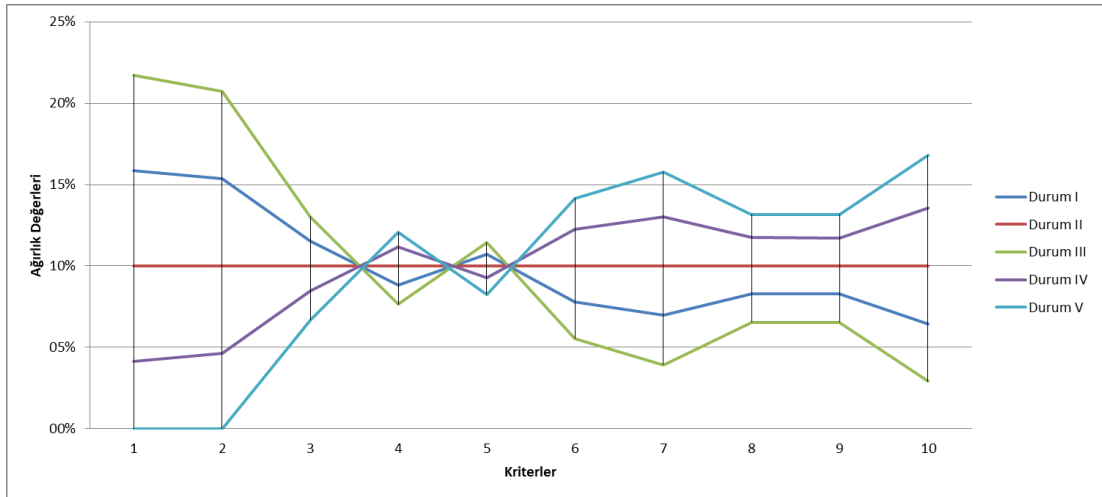
5.2. NZN-Alan Yöntemini Doğrulamak için Türetilen Durum Tanımları

Proje değeriendirme ve takımların yetkinliklerini değeriendirme kriterleri Yel ve ark. (2021) çalışmasından alınmıştır. NZN ve NZN-Alan yöntemlerinin kriter ağırlıkları üzerindeki karar etkinlikleri, gerçekte ilgili projeleri geliştiren takımlardan elde edilen ekiplerin bir kiři-gün çabasına karşılık kaç adet kusur ürettiğine göre belirlenmiştir.

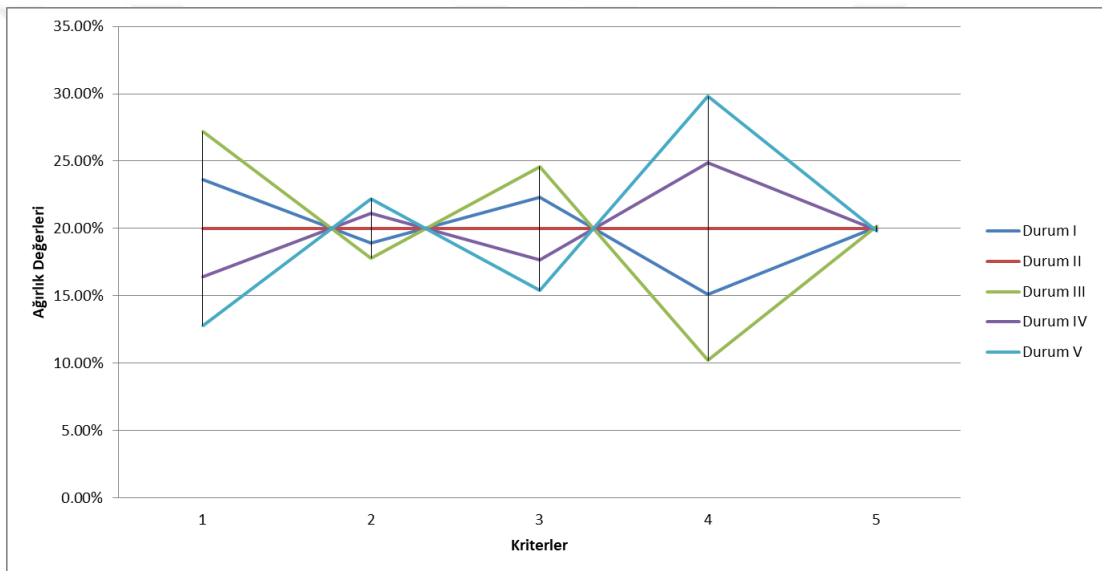
Ayrıca kriter ağırlıkları Yel ve ark. (2021) hesaplamaları, çalışmada belirlenen ağırlıklar üzerinden değil, aşğıdaki senaryolar üzerinden ek dört durum tanımı ile tekrarlanmıştır. Senaryoların belirlenmesi aşamasında, Baysal ve ark. (2015) tarafından yapılan çalışma temel alınmıştır;

- **Durum I:** Yel ve ark. (2021) çalışmasında belirlenen ağırlıklar esas alınmıştır.
- **Durum II:** Aşğıdaki denklemd e tüm kriterlerin eşit ağırlıkta olduğı varsayılır.
 - $100/(\text{Kriter Sayısı}), 100/10=10$ (Proje Değeriendirme Kriter Ağırlıkları),
 $100/5=20$ (Takım Yetkinlik Değeriendirme Ağırlıkları)
- **Durum III:** Durum I'deki kriter ağırlıklarının Durum II'deki ağırlıklara göre simetrisi alınır.
- **Durum IV:** Bu seçenekte, Durum I'deki ağırlıklar ile Durum II'deki ağırlıklar arasındaki fark, Durum II'ye iki ile çarpılarak eklenir.
- **Durum V:** Durum IV'teki ağırlıklar Durum II'deki ağırlıklara dayanmaktadır ve simetri değeri ler ağırlık olarak belirlenir.
 - Bu durumda Proje Değeriendirme Kriterlerinden "0"ın altındaki simetrik değeri ler "0" olarak alınmış ve diğeri kriterlere eşit olarak dağıtılmıştır.

Şekil 5.2 ve Şekil 5.3'te, yukarıdaki beş durum için ağırlıkların çizgi grafikleri bulunmaktadır. Eşit ağırlıklara sahip Durum II'nin diğeri durumlar için temel değeri olarak belirlendiğı ve simetri eksenini görevi görerek simetrik yansımaların üretiminde kullanıldığı görölmektedir.



Şekil 5.2. Proje Değerlendirme Kriterlerinin Durum Bazlı Ağırlıkları



Şekil 5.3. Takım Değerlendirme Kriterlerinin durum bazlı ağırlıkları

Proje ekibi atamalarının sonuçları, NZN ve NZN-Alan yöntemleri beş farklı durum üzerinden denenerek incelenmiştir.

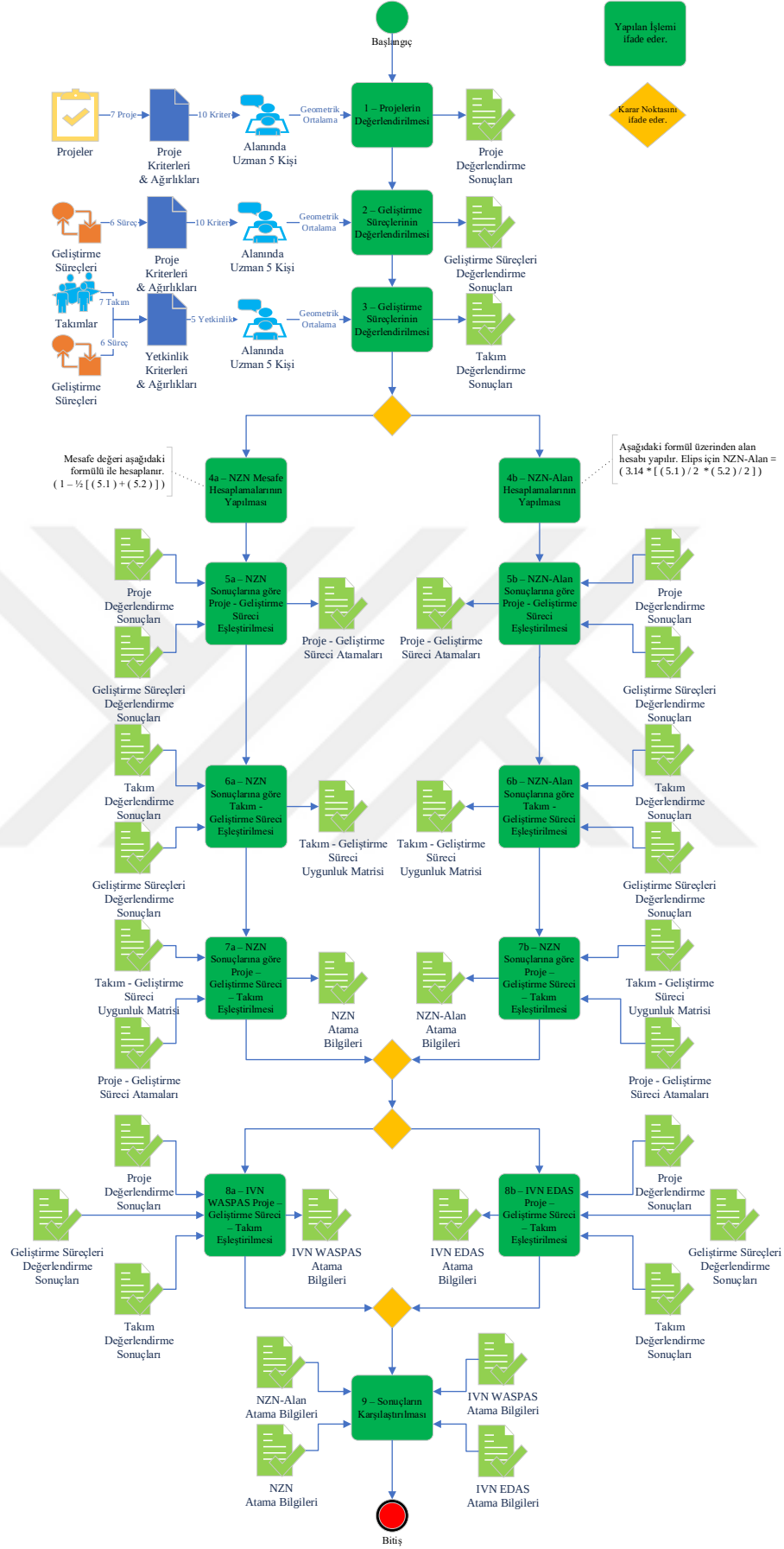
5.3. Bulanık ÇKKV'nin Yazılım Geliştirme Proje Yönetimine Uygulanması

Nötrosofik kümeler, bulanık kümeler ile sezgisel bulanık kümelerin etkileşime girdiği ve geliştirildiği kümeler olarak düşünülebilir. NZN'de T (Doğruluk), I (Belirsizlik) ve F (Yanlışlık) değişkenleri de dilsel değişkenler üzerinden ifade edilerek karar aşamasında kullanılabilecek bir yöntem olarak geliştirilmiştir (Çetiner Karataş, 2019).

Diğer yandan nütrosifik küme, “e” evrenindeki nütrosifik kümenin bir doğruluk, bir belirsizlik ve bir yanlışlık üyeliği ile tanımlanır. Bu tanım aynı zamanda modelde doğruluk, belirsizlik ve yanlışlık ile nütrosifik kümeleri kullanarak karar vericilerin daha etkin kararlar almasının yolunu açmaktadır (Smarandache, 1998). Tezde kullanılan nütrosifik sayı kümeleri yönteminin adı tam olarak Aralık Değerli Nütrosifik Z Sayıları'dır. Yukarıda belirtilen değişken aralıklı üç üyelik kümeli gerçek hayat modelinde (T-I-F), ölçeğin alt ve üst sınır değerlerini de içermektedir (Taş ve Karataş, 2021). Çizelge 5.1 ve devamındaki Şekil 5.4'te NZN ve NZN-Alan yöntemlerinin proje değerlendirmeye uyarlanması, geliştirme yönteminin belirlenmesi ve ilgili takıma projelerin atanmasında izlenen adımlar ayrıntılı olarak verilmiştir. Şekil 5.4 adım adım veri ilişkileri üzerinden problem çözümüne dair izlenen yolu özetlemektedir. Bu çizelge ve şekilde yer alan alanında uzman kişiler, bilgi teknolojileri alanında en az 5 yıllık tecrübesi olan ve aktif olarak halen bu sektörde çalışmaya devam eden mühendislerden oluşmaktadır.

Çizelge 5.1. NZN ve NZN-Alan yöntemlerinin problem çözme adımları

NZN	NZN-Alan
Alternatif projeler, proje karar kriterlerine göre beş farklı uzman tarafından Bolturk ve Kahraman (2018) 'de tanımlanan ölçeğe göre değerlendirilir.	
Geliştirme Süreçleri, her birisi için ayrı ayrı proje değerlendirme kriterleri üzerinden beş farklı uzman tarafından aynı ölçeğe göre değerlendirilir.	
Alternatif takımlar, Takım Yetkinlikleri değerlendirme kriterlerine göre beş farklı uzman tarafından aynı ölçekte değerlendirilir.	
Yazılım Geliştirme Yöntemleri, her yöntem için ayrı ayrı takım yetkinlik değerlendirme kriterleri üzerinden beş farklı uzman tarafından aynı ölçeğe göre değerlendirilir.	
Değerlendiricilerin değerlendirmelerinin geometrik ortalamaları alınarak alternatiflerin T, I ve F değerleri bulunur.	
Bulunan değerlere göre genelleştirilmiş mesafeye dayalı benzerlik ölçüsü denkleminde göre sıralama yapılır. (1-1/2[Denklem(5.1) + Denklem(5.2)])	Bulunan değerlere göre TN, IN, FN değerlerinin aralıklarının ağırlıklı ortalama değeri alınır. TM, IM ve FM değerlerinin ağırlıklı ortalama değeri alınır.
	Daha sonra N için bulunan değer ile M için bulunan değer aritmetik ortalaması yerine üç farklı yaklaşımla alan hesabına göre sıralama yapılır.
Projelerin sıralama sonuçları ile Geliştirme Süreci sıralama sonuçları karşılaştırılarak hangi projenin hangi yazılım geliştirme süreci ile geliştirilmesi gerektiği ataması yapılır.	
Takımların değerlendirmeleri ile geliştirme süreçlerinin ideal değerleri birbirine yakınlık açısından farkları alınarak hesaplanır.	
Projelerin eşleştirildiği geliştirme sürecine hangi takımların yatkın olduğuna göre proje ve takım eşleştirmesi verilmektedir. Bu eğilimi takımların kendi değerlendirmelerinde ilgili geliştirme sürecinin ideal değerlerine olan uzaklıkları belirler.	
Yukarıdaki işlemler beş farklı durum üzerinde çalıştırılarak karşılaştırma çizelgesi elde edilir.	
Devamında NZN, NZN-Alan, Aralık Değerli EDAS ve Aralık Değerli WASPAS yöntemleri aynı probleme uygulanarak sonuçları kıyaslanır. Böylece NZN-Alan yönteminin etkililiği üzerine yorum yapılabilecek bilgiler elde edilmiştir.	



Şekil 5.4. NZN ve NZN-Alan yöntemlerinin problem çözme diyagramı

5.4. Problem Tanımı ve Kriter Değerlendirme Ayrıntıları

Problem kısaca, bir yazılım kuruluşunda geliştirilmek üzere oluşturulan yedi adet projenin bir organizasyonel birim içinde aktif görev alan ve iki kişiden oluşan yedi adet takıma atanması olarak tanımlanabilir. İlk olarak tez bünyesinde tanımlanan toplam altı adet olan yazılım geliştirme süreci değerlendiricilerce değerlendirilmiş ve ideal değerler seçilen kriterler üzerinden elde edilmiştir. Sonra, projelerin hangi geliştirme sürecine uygun olduğu değerlendiricilerce yapılan değerlendirmeler ile elde edilmiştir. Daha sonra, iki kişiden oluşan takımların yetkinlik değerlendirmeleri yapılarak hangi geliştirme sürecine yakın olduklarına dair hesaplamalar yapılmıştır.

Yedi proje 2020 yılının ikinci altı aylık döneminde geliştirilen projeler arasından seçilmiş, takım atamaları da gerçek atamalar üzerinden ele alınarak kusur analizleri yapılmıştır. Bu başlık altında yer alan değerlendirmelere ait bilgiler Alan_{elips} yaklaşımına göre paylaşılmıştır. Diğer yaklaşımların sonuçları karşılaştırmaların yapıldığı çizelgelerde detaylı yer almaktadır. Projeler geliştirilirken takımların performansının bir göstergesi olan kusur oluşum bilgisi karşılaştırma amacıyla kullanılmıştır. Bu bilgilere göre takımların bir kişi-günlük çabasına karşılık üretilmiş kusur sayıları ve bu kusurların çözüm süreleri Çizelge 5.2’de verilmiştir.

Çizelge 5.2. Takımların kusur çözüm bilgileri

Takımlar	Kusur/Kişi Gün	Çözüm Zamanı (Gün)/Kusur
T1	0.410	2.841
T2	0.356	3.029
T3	0.634	4.001
T4	0.272	2.846
T5	0.000	0.000
T6	0.389	3.454
T7	0.255	2.426

Problemin çözümü madde madde aşağıda açıklanmıştır;

1. Projelerin değerlendirme kriterleri ve kriterlerin ağırlıkları Yel ve ark. (2021) tarafından yapılan çalışmadan alınmıştır. Bu değerler Çizelge 5.3’de gösterilmiştir.

Çizelge 5.3. Proje değerlendirme kriterleri ve ağırlıkları

Kriterler	Ağırlıkları
Gerçek Zamanlı Çalışma	%15.90
Hata Duyarlılığı	%15.40
Yüksek İşlem Kapasitesi	%11.50
Özel Geliştirme	%8.80
Proje Büyüklüğü	%10.70
Resmi / Sözleşme Bağlılığı	%7.80
Paydaş Sayısı	%7.00
Bürokrasi	%8.30
Değişime Direnç	%8.30
Organizasyon Büyüklüğü	%6.50

2. Takım değerlendirmeleri için yetkinlik kriterleri ve kriterlerin ağırlıkları da Yel ve ark. (2021) çalışmasından alınmıştır. Bu değerler Çizelge 5.4'te gösterilmektedir.

Çizelge 5.4. Takım Yetkinlikleri değerlendirme kriterleri ve ağırlıkları

Kriterler	Ağırlıkları
Tecrübe	%23.60
Alan Bilgisi	%18.90
İletişim	%22.30
Takım Çalışmasına Uyum	%15.10
Analitik Düşünme	%20.10

3. Proje kriterleri ve takım değerlendirme kriterlerinin uzmanlar tarafından değerlendirilmesinde Bolturk ve Kahraman (2018) tarafından tanımlanmış ölçek kullanılmıştır (Çizelge 5.5).

Çizelge 5.5. NZN Ölçek Kümesi

Sıra Numarası	Dilsel Terimler	Nötrosifik Küme					
		TL	TU	IL	IU	FL	FU
1	Eşit önem	0.5	0.5	0.5	0.5	0.5	0.5
2	Çok daha fazla önem	0.5	0.6	0.35	0.45	0.4	0.5
3	Orta önem	0.55	0.65	0.3	0.4	0.35	0.45
4	Orta derecede daha fazla önem	0.6	0.7	0.25	0.35	0.3	0.4
5	Güçlü önem	0.65	0.75	0.2	0.3	0.25	0.35
6	Çok daha önem	0.7	0.8	0.15	0.25	0.2	0.3
7	Çok güçlü önem	0.75	0.85	0.1	0.2	0.15	0.25
8	Çok daha fazla önem	0.8	0.9	0.05	0.1	0.1	0.2
9	Aşırı önem	0.9	0.95	0	0.05	0.05	0.15
10	Son derece yüksek önem	0.95	1	0	0	0	0.1
11	Kesinlikle daha fazla önem	1	1	0	0	0	0

4. NZN için proje ve takım kriterleri ağırlıkları ve ölçek bilgileri belirlendikten sonra geleneksel ve çevik geliştirme süreçleri için aynı kriterler bağlamında bir değerlendirme yapılmıştır. Değerlendirmeye konu olan yazılım geliştirme yöntemleri, Gencer ve Kayacan (2017b) çalışmasında yapılan derlemeden yola çıkılarak Türkiye’de en yaygın kullanılan çevik yöntemlerden ve en bilinen geleneksel yöntemlerden seçilmiştir. NZN ölçeği temel alınarak, geliştirme süreçlerinin projeler özelinde değerlendirme sonuçları ile takım yetkinliklerine ait değerlendirme sonuçları sırasıyla Çizelge 5.6 ve Çizelge 5.7’de örnek olarak verilmiştir. Bu çizelgelerdeki değerler, Çizelge 5.5’e göre dilsel değişkenlerin sıra numaralarını ifade etmektedir ve “Karar Verici 1”’in değerlendirmeleridir.

Çizelge 5.6. Proje Değerlendirme kriterleri ve Geliştirme Süreci değerlendirme bilgileri

	Kanban	Scrum	XP	Şelale	V Model	Spiral Model
Gerçek Zamanlı Çalışma	1	1	2	7	7	7
Hata Duyarlılığı	2	2	2	5	11	11
Yüksek İşlem Kapasitesi	3	6	3	4	9	4
Özel Geliştirme	2	3	2	1	1	1
Proje Büyüklüğü	1	1	1	7	7	7
Resmi / Sözleşme Bağlılığı	3	1	3	6	6	11
Paydaş Sayısı	2	2	5	6	9	6
Bürokrasi	3	4	3	7	10	7
Değişime Direnç	2	1	2	4	2	2
Organizasyon Büyüklüğü	3	3	3	7	7	11

Çizelge 5.7. Takım Yetkinliği Değerlendirme kriterleri ile Geliştirme Süreci değerlendirme bilgileri

	Kanban	Scrum	XP	Şelale	V Model	Spiral Model
Tecrübe	3	2	1	10	2	10
Alan Bilgisi	2	4	1	10	7	10
İletişim	11	10	11	1	1	1
Takım Çalışmasına Uyum	3	5	2	2	6	2
Analitik Düşünme	1	11	1	1	1	1

5. Yazılım projeleri geliştirme süreçleri üzerinde değerlendirmeler yapıldıktan sonra projeler ve takım yetkinlikleri değerlendirilmiştir. Örnek sonuçlar “Karar Verici 1” için sırasıyla Çizelge 5.8 ve Çizelge 5.9’da sunulmaktadır.

Çizelge 5.8. NZN Ölçeğinde proje değerlendirme kriterlerinin değerlendirme sonuçları

Projeler	Pr-1	Pr-2	Pr-3	Pr-4	Pr-5	Pr-6	Pr-7
Gerçek Zamanlı Çalışma	6	4	7	2	3	6	11
Hata Duyarlılığı	1	4	4	6	10	9	1
Yüksek İşlem Kapasitesi	1	3	9	4	9	3	9
Özel Geliştirme	5	10	2	5	1	7	2
Proje Büyüklüğü	3	6	10	3	10	6	7
Resmi / Sözleşme Bağlılığı	2	11	1	9	2	3	4
Paydaş Sayısı	9	1	2	4	9	5	1
Bürokrasi	3	6	1	1	10	5	6
Değişime Direnç	7	10	5	6	4	10	4
Organizasyon Büyüklüğü	3	10	11	11	5	5	7

Çizelge 5.9. Takım yetkinlik kriterlerinin NZN ölçeğine ilişkin değerlendirme sonuçları

Takımlar	T1	T2	T3	T4	T5	T6	T7
Tecrübe	3	5	3	2	2	6	3
Alan Bilgisi	10	2	9	1	5	6	3
İletişim	7	4	2	7	11	7	1
Takım Çalışmasına Uyum	1	11	10	3	3	9	3
Analitik Düşünme	7	6	4	11	2	7	11

6. Proje değerlendirme kriterlerinin NZN ölçeğinde değerlendirilmesi ile NZN için denklem (4.31), NZN-Alan_{elips} için denklem (5.3) kullanılarak hesaplamalar yapılmıştır. NZN için hesaplamanın maksimumunun atamalar için en iyi olduğu, NZN-Alan_{elips} için minimum değer en iyisi olduğu belirtilmelidir. NZN için mesafe ve NZN-Alan_{elips} için alan bilgilerini içeren

hesaplama sonuçları Durum I için Çizelge 5.10'da verilmiştir. Çizelge 5.10, projelerin hangi yazılım geliştirme yöntemiyle geliştirileceğinin atamasını içermektedir. Atamalar, Çizelge 5.10'da koyu karakterlerle gösterilmiştir.

- Amaç fonksiyonu, projelerin takımlara seçilen geliştirme süreci ile atandığında, takımların ilgili geliştirme sürecine uyumsuzluklarının minimize edilmesi olarak tanımlanmıştır.
- İlk kısıt, her projenin en fazla 1 takıma atanması şeklinde tanımlanmıştır.
- İkinci kısıt ise her takıma en fazla 1 projenin atanması şeklinde tanımlanmıştır.

Çizelge 5.10. NZN ve NZN-*Alan_{elips}* sıralama sonuçları

	Pr-1: XP	Pr-2: XP	Pr-3: Kanban	Pr-4: Kanban	Pr-5: Scrum	Pr-6: Şelale	Pr-7: Kanban
NZN							
T1	0.883	0.883	0.841	0.841	0.854	0.851	0.841
T2	0.908	0.908	0.883	0.883	0.899	0.808	0.883
T3	0.854	0.854	0.821	0.821	0.833	0.771	0.821
T4	0.869	0.869	0.910	0.910	0.908	0.836	0.910
T5	0.957	0.957	0.924	0.924	0.926	0.910	0.924
T6	0.849	0.849	0.847	0.847	0.841	0.893	0.847
T7	0.887	0.887	0.898	0.898	0.907	0.812	0.898
	Pr-1: XP	Pr-2: XP	Pr-3: Kanban	Pr-4: Kanban	Pr-5: Scrum	Pr-6: Şelale	Pr-7: Kanban
NZN-<i>Alan_{elips}</i>							
T1	0.011	0.011	0.020	0.020	0.017	0.018	0.020
T2	0.007	0.007	0.011	0.011	0.008	0.029	0.011
T3	0.017	0.017	0.025	0.025	0.022	0.041	0.025
T4	0.013	0.013	0.006	0.006	0.007	0.021	0.006
T5	0.001	0.001	0.005	0.005	0.004	0.006	0.005
T6	0.018	0.018	0.018	0.018	0.019	0.008	0.018

7. Atamalar, fayda ve maliyet kriterleri üzerinden Excel Çözücü aracılığıyla yapılmıştır. Yedi takıma yedi proje atanacağından, bu tür bir problem, zor olmayan bir karar matrisi olduğundan Excel tarafından en iyi şekilde çözülebilir. Çizelge 5.11, Durum I için proje atamalarının NZN ve NZN-*Alan_{elips}* yöntemleriyle takım atama örnek sonuçlarını göstermektedir.

NZN ve NZN-Alan yöntemleri, beş farklı durum ve üç yaklaşım üzerinden analiz edilerek proje ekibi atamalarının sonuçları elde edilmiştir. Hem geliştirme süreçlerinin hem projelerin hem de takımların değerlendirmeleri alanında uzman ve en az beş yıllık deneyime sahip mühendisler tarafından yapılmıştır. Bu değerlendirmeler temelinde ağırlıklar farklılaştırılarak beş farklı durum için deneysel sonuçlar elde edilmiştir.

Çizelge 5.13, kusur sayısına özel "Gerçek Değerler - NZN" - "Gerçek Değerler - NZN- $Alan_{elips}$ " - "NZN – NZN- $Alan_{elips}$ " karşılaştırmalarını içerir. Benzer şekilde Çizelge 5.14'te $Alan_{dikdörtgen}$ yaklaşımını temsil eder ve Çizelge 5.15 $Alan_{hibrit}$ sonuçlarını temsil etmektedir. Başarı oranlarını içeren sütunlar (Yöntem Sonucu–Gerçek Değer)/(Gerçek Değer) denklemi üzerinden hesaplanmıştır. İki farklı yöntemin kıyaslanmasında (İlk Yöntem Sonucu–İkinci Yöntem Sonucu)/(İlk Yöntem Sonucu) denklemi kullanılmıştır.

Çizelge 5.13. $Alan_{elips}$ için tüm Durum atamalarına göre kusur sayısı sonuçları

Kusur Sayısı	Gerçek Değer	NZN	NZN-Alan	Gerçek - NZN Oranı	Gerçek - NZN Alan Oranı	NZN - NZN Alan Oranı
Durum - I	162	173.57	146.02	%-7	%10	%16
Durum - II	162	170.46	130.43	%-5	%19	%23
Durum - III	162	136.92	133.20	%15	%18	%3
Durum - IV	162	138.61	130.43	%14	%19	%6
Durum - V	162	169.81	129.21	%-5	%20	%24

Çizelge 5.14. $Alan_{dikdörtgen}$ için tüm Durum atamalarına göre kusur sayısı sonuçları

Kusur Sayısı	Gerçek Değer	NZN	NZN-Alan	Gerçek - NZN Oranı	Gerçek - NZN Alan Oranı	NZN - NZN Alan Oranı
Durum - I	162	173.57	146.02	%-7	%10	%16
Durum - II	162	170.46	130.43	%-5	%19	%23
Durum - III	162	136.92	136.72	%15	%15	%0
Durum - IV	162	138.61	130.43	%14	%19	%6
Durum - V	162	169.81	157.78	%-5	%3	%7

Çizelge 5.15. $Alan_{hibrit}$ için tüm Durum atamalarına göre kusur sayısı sonuçları

Kusur Sayısı	Gerçek Değer	NZN	NZN-Alan	Gerçek - NZN Oranı	Gerçek - NZN Alan Oranı	NZN - NZN Alan Oranı
Durum - I	162	173.57	148.46	%-7	%8	%14
Durum - II	162	170.46	132.92	%-5	%18	%22
Durum - III	162	136.92	126.59	%15	%22	%8
Durum - IV	162	138.61	131.18	%14	%19	%5
Durum - V	162	169.81	145.75	%-5	%10	%14

Çizelge 5.13, 5.14 ve 5.15 incelendiğinde, alan bazlı sıralama yöntemlerinin tüm durumlar için NZN yöntemine göre daha iyi sonuçlar verdiği görülmektedir. Öte yandan, $Alan_{hibrit}$ sorunumuza ilgili alan bazlı hesaplamalar arasında en iyi (126.59) çözümü vermiştir. $Alan_{dikdörtgen}$ yaklaşımı ile ilgili olarak, NZN'nin en iyi sonucu olan Durum III ile aynı çıktıyı üretmiş, ancak diğer durumlarda daha iyi sonuçlar verdiği görülmüştür.

NZN ve NZN-Alan karşılaştırması, yalnızca kusur sayısı açısından değil, aynı zamanda kusur çözümlerinin çözüm süresi açısından da yapılmıştır. Üç yaklaşım için ($Alan_{elips}$, $Alan_{dikdörtgen}$ ve $Alan_{hibrit}$) sırasıyla Çizelge 5.16, 5.17 ve 5.18'de bu karşılaştırmaların bir özeti verilmiştir. Başarı oranları, kusur sayılarında açıklandığı gibi hesaplanmış ve çizelgelere eklenmiştir.

Çizelge 5.16. $Alan_{elips}$ için tüm Durum'lar bazında kusur Kişi-Gün çözüm sonuçları

Kusur Çözüm Süreleri	Gerçek	NZN	NZN-Alan	Gerçek - NZN Oranı	Gerçek - NZN Alan Oranı	NZN - NZN Alan Oranı
Durum - I	523	576.54	447.83	%-10	%14	%22
Durum - II	523	572.96	418.24	%-10	%20	%27
Durum - III	523	453.47	410.26	%13	%22	%10
Durum - IV	523	443.03	418.24	%15	%20	%6
Durum - V	523	565	401.48	%-8	%23	%29

Çizelge 5.17. $Alan_{dikdörtgen}$ için tüm Durum'lar bazında kusur Kişi-Gün çözüm sonuçları

Kusur Çözüm Süreleri	Gerçek	NZN	NZN-Alan	Gerçek - NZN Oranı	Gerçek - NZN Alan Oranı	NZN - NZN Alan Oranı
Durum - I	523	576.54	447.83	%-10	%14	%22
Durum - II	523	572.96	418.24	%-10	%20	%27
Durum - III	523	453.47	426.29	%13	%18	%6
Durum - IV	523	443.03	418.24	%15	%20	%6
Durum - V	523	565	506.45	%-8	%3	%10

Çizelge 5.18. $Alan_{hibrit}$ için tüm Durum'lar bazında kusur Kişi-Gün çözüm sonuçları

Kusur Çözüm Süreleri	Gerçek	NZN	NZN-Alan	Gerçek - NZN Oranı	Gerçek - NZN Alan Oranı	NZN - NZN Alan Oranı
Durum - I	523	576.54	454.79	%-10	%13	%21
Durum - II	523	572.96	422.24	%-10	%19	%26
Durum - III	523	453.47	390.29	%13	%25	%14
Durum - IV	523	443.03	425.31	%15	%19	%4
Durum - V	523	565	451.53	%-8	%14	%20

Kusurların çözüm sürelerine ilişkin beş durum için üç farklı yaklaşımın sonuçları incelendiğinde, sayı ile ilgili sonuçlara benzer şekilde en iyi sonucu $Alan_{hibrit}$ yaklaşımının (390.29) verdiği görülmektedir. Ancak genel olarak, ortalama değerler üzerinden tüm durumları en iyi yöneten yaklaşımın $Alan_{elips}$ olduğu açıktır.

Tüm sonuçlar üzerinden bir değerlendirme yapıldığında NZN yönteminin çözüm için ürettiği durum bazında çıktıların ortalamalarının gerçek değerler üzerinden kusur sayıları bazında %3, çözüm süresi bazında ise %0 olduğu görülmektedir. Diğer taraftan, $Alan_{elips}$ için sayı bazında ortalama iyileştirme %17, çözüm süresi ise beş durumun ortalaması üzerinden %20 olarak gerçekleşmiştir. Benzer şekilde, bu değerler $Alan_{dikdörtgen}$ için %13 - %15 ve $Alan_{hibrit}$ için %15 - %18'dir.

6. DOĞRULAMA VE DURUM SONUÇLARI

6.1. NZN-*Alan_{elips}* için Tüm Durumların Duyarlılık Analizi

Duyarlılık analizi için değişken proje değerlendirme kriter ağırlıkları ve değişken takım değerlendirme kriter ağırlıkları açısından 3 farklı alan yaklaşımının hem kusur sayısı hem de kusurların çözüm süresi değişkenlerinin incelenmesi uygun olacaktır. Şekil 5.2 ve 5.3'teki durum çizgi grafikleri incelendiğinde, Durum II'ye (tüm kriterlerin aynı ağırlığa sahip olduğu) göre simetrik olan ve uç değerleri içeren Durum III ve Durum V'i karşılaştırmak daha anlamlı olacaktır. Aşağıdaki açıklamalarda, en iyi sonuçları veren NZN-*Alan_{elips}* için yorumlar yapılmıştır.

Buna göre, Durum I olarak tanımlanan kriter ağırlıkları üzerinden türetilen Durum III ve Durum V için elde edilen kusur sayılarının gerçek değerler ile kıyas sonuçları çarpıcı olmuştur. NZN-*Alan_{elips}* yöntemi, kusur sayısı çıktısına göre simetrik olan Durum III ve Durum V üzerinden değişken ağırlıklarından az etkilenecek daha istikrarlı sonuçları üretebilmiştir (min: %18 (Durum III) - maks: %20 (Durum V)). Öte yandan NZN yönteminin duyarlılığı aynı durumlar için çok daha yüksek seyretmiş (min: -%5 (Durum II ve Durum V) - maksimum: %15 (Durum III)) ve NZN-Alan kadar iyi sonuçlara da ulaşamamıştır.

Benzer şekilde kusur çözümleme sürelerini incelediğimizde NZN-*Alan_{elips}* gerçek verilere göre daha istikrarlı ve tutarlı sonuçlar üretmiştir (%22 (Durum III) - %23 (Durum V)). NZN yönteminin oranı rakamlara dayalı analize paralel olarak daha yüksektir (-%8 (Durum V) - %13 (Durum III)). Bu durumda değişken kriter ağırlıkları bulunsada NZN-*Alan_{elips}*'in daha tutarlı ve daha yakın sonuçlar ürettiği ve NZN yönteminin kriter ağırlıklarına göre hassasiyet zayıflığını giderdiği ifade edilebilir.

Bundan sonraki konu başlıklarında önerilen modelin, yaygın kullanılan aralık değerli diğer bulanık ÇKKV yöntemleri ile çözümleri ve kıyaslamaları yer almaktadır.

6.2. Problemin Aralık Değerli WASPAS ile Çözümü

Aralık değerli nütrosifik (IVN) WASPAS yönteminin aynı probleme aynı değerlendirme verileri üzerinden uygulanmasıyla elde edilen sonuçlar bu başlık altında yer almaktadır. Bu çalışmada IVN-WASPAS yöntemi Ayyildiz ve Taskin (2022)'da tanımlanan hesaplama adımları uygulanmıştır. Hesaplama adımları aşağıda yer almaktadır;

- **Adım 1:** Alternatiflerin değerlendirme matrisi $(X^{\sim})_{IVN}$ sayılarına dönüştürülmüştür. Alternatiflerin değerlendirme sonuçları, geliştirme süreçleri değerlendirmeleri (GSD) üzerinden mutlak değerler olarak çıkarılmıştır. Böylece her GSD için ayrı bir karar matrisi elde edilmiştir (Denklem 6.1).

$$X^{\sim} = [\tilde{X}^{\sim}_{11} \tilde{X}^{\sim}_{12} \dots \tilde{X}^{\sim}_{1n} : \dots \vdots : \tilde{X}^{\sim}_{m1} \tilde{X}^{\sim}_{m2} \dots \tilde{X}^{\sim}_{mn}] \quad (6.1)$$

- **Adım 2:** Her alternatif için ağırlıklı toplam değerler $(Q^{\sim}1)$ hesaplanmıştır (Denklem 6.2).

$$Q^{\sim}1 = \sum_{j=1}^n \tilde{X}^{\sim}_{ij} w_j, \quad (6.2)$$

- **Adım 3:** Her alternatif için ağırlıklı çarpım değerleri $(Q^{\sim}2)$ hesaplanmıştır (Denklem 6.3).

$$Q^{\sim}2 = \prod_{j=1}^n (\tilde{X}^{\sim}_{ij})^{w_j} \quad (6.3)$$

- **Adım 4:** $Q^{\sim}1$ sonuçları ve $Q^{\sim}2$ sonuçları birleştirilmiştir (Denklem 6.4).

$$Q^{\sim}i = \lambda Q^{\sim}1 + (1 - \lambda) Q^{\sim}2 \quad (6.4)$$

- **Adım 5:** IVN sayılar denklem 6.5 yardımıyla nütrosifik olmayan sayılara dönüştürülmüş ve azalan düzende sıralanmıştır (Denklem 6.5) (Demircan ve Özcan, 2021).

$$\mathcal{K}(A) = \frac{T^L + T^U + 2 - F^L - F^U + T^L \times T^U + \sqrt{(1-F^L)(1-F^U)}}{6} \times \left(1 - \frac{I^L + I^U}{2} - \sqrt{I^L I^U}\right) \quad (6.5)$$

Hesaplama adımları tamamlandıktan sonra aynı problemin çözülmesiyle elde edilen sonuçların karşılaştırılması Çizelge 6.1’de verilmiştir.

Çizelge 6.1. IVN WASPAS için tüm Durum atamalarına göre kusur kriterleri karşılaştırması

Tüm Durumlar	Gerçek Kusur Sayısı	IVN WASPAS	Gerçek Kusur Çözüm Süresi	IVN WASPAS
Durum I	162	157.02	523	500.35
Durum II	162	126.09	523	396.18
Durum III	162	137.67	523	456.21
Durum IV	162	166.76	523	555.24
Durum V	162	159.93	523	504.57

Çizelge 6.1 incelendiğinde IVN WASPAS yönteminin aynı probleme uyarlanması ile elde edilen sonuçlar görülmektedir. Bu sonuçlara göre IVN WASPAS 5 farklı durumun sadece 1 tanesinde gerçek değerlerden daha kötü sonuca ulaşmış, 126.09 gibi en iyi kusur sayısı değerini de hesaplayabilmiştir. Durum I ve II için gerçek değere kusur sayısı bakımından çok yaklaşmıştır. Bununla birlikte yine Durum I ve II için kusur çözüm süresinde gerçek değerle aralarında anlamlı ve pozitif yönde farklılıklar oluşmuştur.

6.3. Problemin Aralık Değerli EDAS ile Çözümü

IVN WASPAS karşılaştırmasına benzer şekilde IVN EDAS yöntemi de aynı sorunu çözmek için kullanılmıştır. IVN EDAS yöntemindeki hesaplama adımları da Demircan ve Özcan (2021) çalışmasına dayanılarak yapılmıştır.

Sonuç değerlerinin nütrosifik olmayan sayılara dönüştürülmesinde denklem 6.5 dayanak olarak kullanılmıştır.

Hesaplama adımları aşağıda özetlenmiştir;

- **Adım 1:** Değerlendirme sonuçlarına göre IVN EDAS karar matrisi (D_j) hazırlanmıştır.
- **Adım 2:** Ortalama karar matrisi, her geliştirme süreci için değerlendiricilerin değerlendirme sonuçlarından elde edilmiştir. Bu nedenle her geliştirme süreci için farklı ortalama karar matrisi elde edilmiştir.
- **Adım 3:** Her bir geliştirme süreci için uzman değerlendirmeleri sonucunda elde edilen ağırlıklar ortalama karar matrisi ile birleştirilmiştir.
- **Adım 4:** $PDA_{m \times n}$ (Denklem 6.6) ve $NDA_{m \times n}$ (Denklem 6.7), n ’inci alternatifi ortalama çözümüne olan pozitif ve negatif mesafenin m ’inci kriteri cinsinden

performans değerini sırasıyla gösterir. Denklemler fayda (B) ve maliyet (C) için gösterilmiştir.

$$pda_{mn} = \begin{cases} \frac{Z(x_{mn} \ominus AV_n)}{\mathcal{K}(av_n)}, & \text{eğer } m \in B \\ \frac{Z(AV_n \ominus x_{mn})}{\mathcal{K}(av_n)}, & \text{eğer } m \in C \end{cases} \quad (6.6)$$

$$nda_{mn} = \begin{cases} \frac{Z(AV_n \ominus x_{mn})}{\mathcal{K}(av_n)}, & \text{eğer } m \in B \\ \frac{Z(x_{mn} \ominus AV_n)}{\mathcal{K}(av_n)}, & \text{eğer } m \in C \end{cases} \quad (6.7)$$

- **Adım 5:** Pozitif mesafe ortalaması (positive distance from average, PDA, Denklem 6.8) ve negatif mesafe ortalaması (negative distance from average, NDA, Denklem 6.9) değerleri hesaplanmıştır.

$$PDA = [pda_{mn}]_{m \times n} \quad (6.8)$$

$$NDA = [nda_{mn}]_{m \times n} \quad (6.9)$$

- **Adım 6:** Tüm alternatifler için PDA (Denklem 6.10) ve NDA'ya (Denklem 6.11) ait ağırlıklı toplam değerleri hesaplanmıştır.

$$sp_n = \sum_{j=1}^l (w_j \otimes pda_{mn}) \quad (6.10)$$

$$np_n = \sum_{j=1}^l (w_j \otimes nda_{mn}) \quad (6.11)$$

- **Adım 7:** sp_n (Denklem 6.12) ve np_n (Denklem 6.13) değerleri tüm alternatifler dikkate alınarak normalleştirilmiştir.

$$nsp_n = \frac{sp_n}{\max(\mathcal{K}(sp_n))} \quad (6.12)$$

$$nsn_n = 1 - \frac{sn_n}{\max(\mathcal{K}(sn_n))} \quad (6.13)$$

- **Adım 8:** Tüm alternatifler için değerlendirme puanı (as_n) hesaplanmış (Denklem 6.14) ve alternatifler azalan değerler yönünde sıralanmıştır.

$$as_n = \frac{1}{2} (nsp_n \otimes nsn_n) \quad (6.14)$$

IVN EDAS yönteminin problem çözümü sonrasında elde edilen değerlendirme kriterlerine ilişkin bilgiler Çizelge 6.2'de yer almaktadır.

Çizelge 6.2. IVN EDAS için tüm durum atamalarına göre kusur kriterleri karşılaştırması

Tüm Durumlar	Gerçek Kusur Sayısı	IVN EDAS	Gerçek Kusur Çözüm Süresi	IVN EDAS
Durum I	162	136.22	523	445.70
Durum II	162	163.11	523	538.17
Durum III	162	132.57	523	424.45
Durum IV	162	135.90	523	421.29
Durum V	162	149.04	523	468.99

Çizelge 6.2 incelendiğinde IVN EDAS, sadece Durum II için gerçek kusur sayısı değerlerini geçmiştir. 132.57 gibi çok düşük kusur sayısına ulaşabilmiştir. Genel anlamda IVN EDAS'ın IVN WASPAS'a göre çıktılar bakımından 5 farklı durum için daha tutarlı olduğu ve birbirine yakın sonuçları ürettiği ifade edilebilir.

6.4. NZN-Alan ile Aralık Değerli WASPAS ve EDAS Kıyaslaması

Aralık değerli ÇKKV yöntemleri, sıralama aşamasında alternatifleri "mesafe" ölçüsüne göre karşılaştırmaktadırlar. Bu tezde iki farklı anlamda kullanılan belirsizlik ve güvenilirlik değerlerinin aritmetik operatörler üzerinden uzaklıklarını hesaplamak yerine "alan" hesaplaması ile sıralama yapılmıştır. Literatürde sıklıkla kullanılan aralık değerli WASPAS ve EDAS yöntemlerinin NZN-Alan yöntemi ile karşılaştırılması bu başlıkta ele alınmaktadır. Sonuçlar Çizelge 6.3 ve 6.4'te yer almaktadır.

Çizelge 6.3. NZN-Alan en iyi sonuçları için tüm durum atamalarına göre kusur sayısı karşılaştırması

Kusur Sayısı	Gerçek Değer	WASPAS	EDAS	NZN-Alan
Durum I	162	157.02	136.22	146.02
Durum II	162	126.09	163.11	130.43
Durum III	162	137.67	132.57	126.59
Durum IV	162	166.76	135.90	130.43
Durum V	162	159.93	149.04	129.21

Çizelge 6.4. NZN-Alan en iyi sonuçları için tüm durum atamalarına göre kusur çözümü kişi-gün süresi karşılaştırması

Kusur Çözüm Süresi	Gerçek Değer	WASPAS	EDAS	NZN-Alan
Durum I	523	500.35	445.70	447.83
Durum II	523	396.18	538.17	418.24
Durum III	523	456.21	424.45	390.29
Durum IV	523	555.24	421.29	418.24
Durum V	523	504.57	468.99	401.48

Çizelge 5.13, 5.14 ve 5.15'te NZN-Alan ve NZN yöntemlerinin kusur sayıları bazında karşılaştırmaları verilmiş ve NZN-Alan'ın daha iyi çözümlerle daha istikrarlı sonuçlar sunduğundan bahsedilmiştir. Çizelge 6.3 ve 6.4 incelendiğinde IVN WASPAS ve IVN EDAS yöntemlerinin sonuçlarının NZN-Alan sonuçlarına yakın olduğu ancak beş farklı durum tanımlamalarının geneli için daha iyi çözüm üretilmedikleri görülmektedir. Bunun yanında 5 farklı durum sonuçlarına bakıldığında NZN-Alan'ın da çeşitli kriter ağırlıkları altında daha istikrarlı sonuçlar üretebildiği görülmektedir. IVN-WASPAS Durum II için kusur sayısına göre 0.5 daha az değere ulaşmasına rağmen çözüm süresi 396.18'de kalarak NZN-Alan Durum III'te elde edilen 390.29 değerinden daha yüksek kişi-gün çözüm süresi sunmuştur. NZN-Alan'ın, karşılaştırma kriterleri olarak en iyi kusur çözümleme süreleri ve sayılarını bulan yöntem olduğu ifade edilir.



7. BULGULAR VE TARTIŞMA

7.1. Bulgular

Bu çalışma ile projelerin doğru takımlar tarafından, doğru geliştirme süreçleriyle yürütüldüğünün doğrulanmasıyla projelerin başarı oranlarında artış sağlanabileceği gösterilmiştir. Böylece bilgi teknolojileri alanındaki yatırımlarda başarısızlık oranının azalmasından fayda sağlanacak ve yatırımların gerçekleşmesine katkı sağlanacaktır.

Projelerin özelliklerine göre en uygun geliştirme sürecinin belirlenmesi, geliştirme sürecine en uygun takımların projelere atanması, bu çalışmanın önerdiği adımlar uygulanarak sağlanmıştır. Böylece basit projelerde yetkin personelin yanlış geliştirme süreci üzerinden atanması ile tam tersi zor projelerde yetkin olmayan personelin yanlış geliştirme süreci üzerinden atanmasından kaynaklanan yönetsel riskler en aza indirilmektedir. Yazılım projelerinde en büyük risk, projelerin başarısız olması ve hedefe ulaşılamaması nedeniyle projelere aktarılan kaynakların israf olmasıdır. Bu çalışmada yönetsel atama modelinin, hem NZN'nin kesin olmayan bilgileri işleme yeteneği, hem de NZN-Alan'ın değişken kriter ağırlıklarına göre daha güçlü olması ve geliştirme süreci detayı açısından çok yönlü olması, ele alınan probleme daha iyi bir çözüm sunmaktadır.

Yalnızca bir kuruluştaki yürütülen yedi proje için, Durum III: NZN-*Alan_{hibrit}* yaklaşımıyla kusur sayısında %22 iyileşme değerine ulaşılmıştır. Bu orana göre kusurlara harcanan gerçek kişi-gün değeri olan 523'ten %22 oranında azaltma yapıldığında 413.41 kişi-gün sayısına ulaşılmaktadır. Ancak aynı Durum III için 390.29 kişi-gün olarak hesaplanmıştır. Bu da 413.41 olması gereken kusur çözüm süresinde yaklaşık %6 verimlilik artışı ile 390.29 kişi-güne ulaşıldığı anlamına gelmektedir. Sonuç olarak model sayı bazında iyileşme sağlarken atama yapılan takımlar üzerinden aynı zamanda kişi-gün çıktısını da iyileştirmiştir. Ele alınan yedi proje için NZN-Alan yöntemi toplam 132.71 kişi-gün kazanımı sağlamıştır.

Proje ve takım atamalarında NZN-Alan yönteminin güvenilirliği kanıtlanmış olup, büyük organizasyonlara sağlayacağı fayda, kişi-gün değerlerinin üzerinde verimlilik artışı ile sonuçlandığı tespit edilmiştir. Ayrıca yazılım geliştirme projelerinin uygun geliştirme sürecine ve ilgili proje üzerinden uygun ekibe atanması ile sürdürülebilir yazılım geliştirme inşa edilecek ve sürdürülebilir sistemlerin oluşmasına katkı sağlanacaktır. Bu tezde önerilen çözüm adımları ve NZN-Alan yönteminin ister büyük

organizasyonlar olsun ister küçük organizasyonlar olsun, karar vericilerin sözlü değerlendirme sonuçlarının alınabileceği tüm organizasyonlarda kullanılabileceği unutulmamalıdır. İhtiyaç duyulan tek kaynak, deneyimli ve yetenekli insan kaynağı içerisinde karar verici olarak görevlendirilebilecek kişilerdir.

7.2. Tartışma

Giderek dijitalleşen dünyada yazılım geliştirme projelerinin başarısı oldukça önemlidir. Yazılım geliştirme projelerinin proje dinamiğine göre hangi geliştirme süreci ile yürütülmesi gerektiğinin belirlenmesi ve bu atama sonucuna göre yetkinlikler üzerinden en uygun ekibin projelere atanması sorununa bu çalışma ile çözüm önerilmiştir.

NZN ile sıralama kriteri olarak sabit bir değer yerine elips alanı bazlı NZN-Alan yönteminin kullanılmasının literatürdeki örneklerinden olan bu çalışmanın sonuçları, yöntemin etkililiğini bir örnek olay çalışmasıyla da desteklemiş, gerçek verilere dayanmaktadır.

Kusur sayısına dayalı iyileştirme oranları ve çözüm süresini ifade eden iyileştirme oranları incelendiğinde, NZN-Alan yönteminin, kusur sayıları ve çözüm süreleri ortalamalarına göre NZN yöntemine nazaran daha iyi sonuçlar verdiği görülmektedir. Firmalarda geliştirilen projelerin fazlalığı ve takım sayısının eksikliği göz önüne alındığında, NZN-Alan'ın en iyi sonuçları %22 daha az kusur sayısı ve %25 daha az kusur çözüm süresi değerleri, büyük miktardaki âtıl kapasitenin iş gücüne dönüştürülebileceğini göstermektedir. Bunun sonucunda da işgücünün ekonomik değere dönüşme değerinde bir artış yaşanacaktır.

Ayrıca doğru geliştirme süreci ile yürütülmesi sağlanan projeler için en uygun ekibin ilgili geliştirme sürecine atanmasının yönetimsel anlamda planlama, organizasyon ve kontrol faaliyetlerine ayrılacak zamanı daha azaltabileceği sonucuna varılmıştır. Hem çevik ve geleneksel geliştirme süreçlerine, hem de yürütülen projelere en uygun takıma atanması, ilgili takımın işi yürütmedeki başarısını ve kararlılığını arttırıcı etkisi göz ardı edilemez.

NZN sayılarının içerdiği bulanık değer ve ikili güvenilirlik değerlendirmesi, sıralama aşamasında ideal çözüm kriterine olan mesafe, tek boyutlu bir sonuç (MD_{wp}) veren denklem (4.31) ile bağlantılıdır. Bunun yerine güvenilirlik kriterinin NZN-Alan ile tek boyutlu mesafe yerine iki boyutlu değerlendirilmesi NZN'ye göre daha etkili sonuçlar vermektedir. Yani ideale yakın ancak güvenilirliği nispeten düşük bir alternatif ile ideale

nispeten yakın ancak güvenilirliği yüksek bir alternatifi kıyaslamak artık çok daha kolaydır.

7.3. Gelecek Araştırmalar İçin Öneriler

Tezde sunulan çözüm adımları ve NZN-Alan yönteminin çıktıları üzerinden sağlanabilecek verimlilik artışları ile iş gücü kazançlarının sektör tarafında karşılık bulacağı düşünülmektedir. Ancak unutulmamalıdır ki gelişim sürekli ve baş döndürücü hızla devam etmektedir.

NZN-Alan yöntemi, duyarlılık analizinde değişken kriter ağırlıklarına göre daha tutarlı ve daha az duyarlı olmasına rağmen, ileriki çalışmalarda değerlendirici ağırlıklarındaki değişkenlik modele dahil edilebilir. Alan formülleri üzerinden yapılan hesaplamalarda kullanılan elips, dikdörtgen ve hibrit yöntemler aynı kriter ağırlıklarında farklı geri kazanım oranlarına sahiptir. NZN-Alan, NZN yöntemine göre daha kararlı sonuçlar üretse de iki farklı değişkenin (büyüklük, pozitiflik, negatiflik, kareden küçük, kök değerinden biraz büyük vb.) arasındaki ilişkiye dayalı olarak daha farklı alan hesaplaması içeren yöntemler geliştirilebilir.

Bu çalışmayı da benzer şekilde ileriye taşıyacak öneriler aşağıda maddeler halinde yer almaktadır;

1. Proje değerlendirme kriterlerinin ve çalışanların yetkinlik değerlendirme kriterlerinin daha da derinleştirilerek fazlalaştırılması ile daha etkili sonuçlar üzerinden atamaların yapılması vasıtasıyla daha etkin sonuçlar elde edilebilir.
2. İkili kıyaslamalarda elde edilen sonuçlar üzerinden nütrosifik TOPSIS yöntemiyle çözümlenerek, pozitif ideale yakınlık ve negatif ideale uzaklık üzerinden sıralamalar ile atama yapılabilir.
3. Atamaların yapılması esnasında kullanılan geliştirme sürecine uygunluk kriterinin yanında, çalışanların tecrübeleri üzerinden belli bir işi “t” süresinde yapma hızları (tempoları) modele yani amaç fonksiyonuna eklenebilir. Böylece doğru projeleri doğru kişilere atama yanında en kısa sürede tamamlanabilmesine yönelik proje – geliştirme süreci – takım atamaları yapılmış olacaktır.
4. Ayrıca çalışma kapsamında yapılan ankete katılan uzman sayısına yazılım geliştirme sektöründe çalışan farklı kuruluşlardan ve ülkelerden kişiler eklenerek, kriter değerlendirmeleri daha da çeşitlendirilebilir. Böylece geliştirme süreci

temelinde ortak fikirlerin bütünleştirilmesi ve daha farklı deęerlendirmeler yapılarak proje ve takım atamaları yapılabilecektir.

5. Farklı alıřma alanları olarak enerji kaynaęı seimi, benzin – elektrik istasyonu seimi (Seikh ve Mandal, 2022), tıbbi atık ve/veya teřhis sorunları, ulařım aracı seimi, personel seimi gibi ok kriterli karar vermenin gerekli olduęu problemlere de NZN-Alan uyarlanabilmektedir (Mandal ve Seikh, 2022). Bu uyarlamalarda asıl konu atama modeli, atanacak proje, hastalık tr, ulařım aracı vb. konular olabilir.



8. SONUÇLAR

Son yıllarda nihai kullanıcılar açısından teknoloji yöneliminin artmasıyla birlikte ticaret, sosyal medya ve büyük veri analizi gibi farklı alanlardaki yazılım geliştirme projelerinin başarılı olması önem kazanmıştır. Gerçek kişilerin dâhil olduğu yazılım geliştirmenin emek yoğun aşamalarında en önemli kaynak olan insanın doğru görevlendirmeler ile bireysel motivasyon artışı sağlandığında, projelerin başarısı üzerindeki olumlu etkisi göz ardı edilemez. Doğru takımların doğru projelere, doğru geliştirme süreci ile atanmasıyla projelerin başarı oranları artacaktır. Bu çalışma, projelerin doğru gereksinim tanımlarına göre değerlendirilmesi ile doğru geliştirme sürecinin bulanık ÇKKV yöntemleri ile çevik ve şelale geliştirme süreçlerinin alt kırılımlarına dayanması artı bir değerdir. Benzer bir değerlendirme ile takımların projelere atanması, ara katman olarak geliştirme sürecinin doğru belirlenmesi, sonuçlar özelinde projelerin başarılı sonuçlanmasını sağlayacaktır.

Önerilen NZN-Alan yönteminin kusur analizinde çok daha iyi sonuçlar üretmesi ve değişken kriter ağırlıklarına göre çok daha tutarlı olması, yönetsel bakış açısının bilgi teknolojileri alanındaki proje geliştirmelerine etkisinin açık bir göstergesidir. Proje yatırımlarının gerçek hayatta önemli bir adım olarak bu çalışmanın çıktılarının özellikle büyük ölçekli firmalarda uygulanması halinde elde edilecek mali kazancın çok yüksek olacağı düşünülmektedir. Ancak ölçülemeyen çıktılar üzerindeki olumlu etkinin daha büyük olabileceği varsayılmaktadır.

Çizelge 6.3 ve 6.4'te yer alan doğrulama amaçlı IVN EDAS, IVN WASPAS ve NZN-Alan yöntemleri kıyaslamasında yöntemlerin değişken kriter ağırlıkları üzerinden türetilen durumların çözümünde verdikleri cevaplar yer almaktadır. Buna göre 5 farklı duruma ait sonuçlar kusur sayısı bakımından standart sapması alındığında IVN EDAS, IVN WASPAS ve NZN-Alan için 16.9, 12.7 ve 7.7 değerlerine ulaşılmaktadır. Aynı sıra ile kusur çözüm süreleri için standart sapmalar 59.7, 47.8 ve 21.7 olarak tespit edilmiştir. Bu değerlendirmelere göre NZN-Alan yönteminin değişken koşullara göre daha istikrarlı sonuçları ürettiği ifade edilebilir.

Pandemi nedeniyle dijitalleşmeye hız veren dünyada yazılım geliştirme projelerinin başarısı iş dünyası açısından oldukça önemlidir.

Doğru projelerin doğru çalışan ya da takımlara, ÇKKV yöntemleri üzerinden yapılan uzman değerlendirmeleri ışığında atamalarının, proje başarısını iyileştirici etkisi bulunmaktadır. Bu kapsamda Yel (2019) tarafından yapılan çalışmada sadece projelerin

alıřanlara KKV ile atanması ile elde edilen iyileřmeler ve iř yknn daha dengeli arttıęı alıřma, Yel ve ark. (2021) ve alıřmadan yola ıkılarak geliřtirilmiř hali olan Yel ve Baysal (2023), alıřmalarında evik ve řelale geliřtirme sreleri bazında deęerlendirilen proje ve alıřanların deęerlendirmeleri sonucu yapılan atamalarda daha da iyileřtięi grlmřtr.

Yel ve Engin (2022) tarafından yapılan alıřmada endstri mhendislięi disiplininde yer alan bilimsel yntemlerin yazılım geliřtirme sreci, ynetim, karar verme gibi alanlarda kullanılması ile iyileřtirmelerin saęlanabileceęi gsterilmiřtir.

İncelenen problem(lerin) zmnde derinlemesine arařtırmaların yapılması ve daha anlamlı kıyaslamalar ile ynetsel aıdan alınacak kararlarda, bulanık KKV yntemlerinin kullanılması ile srekli iyileřme saęlanabilir.



KAYNAKLAR

- Abdel-Basset, M., Mohamed, M., & Smarandache, F., 2018, An extension of neutrosophic AHP–SWOT analysis for strategic planning and decision-making, *Symmetry*, 10(4), 116.
- Abdel-Basset, M., Mohamed, M., Zhou, Y., & Hezam, I., 2017, Multi-criteria group decision making based on neutrosophic analytic hierarchy process, *Journal of intelligent & fuzzy systems*, 33(6), 4055-4066.
- Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J., 2017, Agile software development methods: Review and analysis[online], *arXiv preprint arXiv:1709.08439*, <https://arxiv.org/ftp/arxiv/papers/1709/1709.08439.pdf> [Ziyaret Tarihi: 01 Eylül 2023]
- Adanna, A. A., & Nonyelum, O. F., 2020, Criteria for Choosing the Right Software Development Life Cycle Method for the Success of Software Project, *IUP Journal of Information Technology*, 16(2), 39-65.
- Ahmad, G., Soomro, T. R., & Naqvi, S. M. R., 2016, An overview: merits of agile project management over traditional project management in software development, *J. Inf. Commun. Technol*, 10(1), 105-120.
- Ahmad, M. O., Markkula, J., & Oivo, M., 2013, Kanban in software development: A systematic literature review, *2013 39th Euromicro conference on software engineering and advanced applications*, Stander Spain, 9-16.
- Akça, H. K., 2022, Bilgi teknolojilerinde proje yönetim metodolojilerinin bilişsel ergonomik açıdan değerlendirilmesi, Yüksek Lisans Tezi, Sakarya Üniversitesi Fen Bilimleri Enstitüsü, 1-70.
- Al Khalidi, N., Saifan, A. A., & Alsmadi, I. M., 2012, Selecting a standard set of attributes for cost estimation of software project, *Amman*, 1-5.
- Alava, M. V., Figueroa, S. P. D., Alcivar, H. M. B., & Vázquez, M. L., 2018, , *Single valued neutrosophic numbers and analytic hierarchy process for project selection*. Neutrosophic Sets and Systems, 21, 122-130.
- Anonim, 2019, Agile's 3 categories: Iterative, Incremental, and Evolutionary[online], Web Adresi: <https://agilenotion.com/agile-categoriesiterative-incremental-evolutionary/> [Ziyaret Tarihi: 20 Mart 2019]
- Antony, M., 2018, Agile Project Management[online], Web Adresi: <https://vitalitychicago.com/blog/agile-projects-are-more-successful-traditional-projects/> [Ziyaret Tarihi: 01 Aralık 2019]
- Anwer, F., Aftab, S., Shah, S. S. M., & Waheed, U., 2017, Comparative analysis of two popular agile process models: extreme programming and scrum, *International Journal of Computer Science and Telecommunications*, 8(2), 1-7.
- Archibald, R. D. (2004). A global system for categorizing projects: the need for, recommended approach to, practical uses of, and description of a current project

to develop the system, *2nd Latin American PMIGOVSIG Forum on Project Management In Government*, Brasil, 2-16.

- Ayyildiz, E., & Taskin, A., 2022, 251 A Novel Interval Valued Neutrosophic AHP-WASPAS Methodology for Emergency Supply Depot Location Selection Problems, *Multi-Criteria Decision Analysis: Case Studies in Disaster Management*, 251-266.
- Barlow, J. B., Giboney, J., Keith, M. J., Wilson, D., Schuetzler, R. M., Lowry, P. B., & Vance, A., 2011, Overview and guidance on agile development in large organizations. *Communications of the Association for Information Systems*, 29(2), 25-44.
- Baysal, M. E., Kaya, İ., Kahraman, C., Sarucan, A., & Engin, O., 2015, A two phased fuzzy methodology for selection among municipal projects, *Technological and Economic Development of Economy*, 21(3), 405-422.
- Beatty, J., & Wiegers, K., 2015, Forward thinking for tomorrow's projects requirements for business analytics[online], Seilevel Whitepaper, Web Adresi: https://d1wqtxts1xzle7.cloudfront.net/54058905/Requirements-for-Business-Analytics-Seilevel-libre.pdf?1501863254=&response-content-disposition=inline%3B+filename%3DForward_Thinking_for_Tomorrows_Projects.pdf&Expires=1696170591&Signature=P65gA2sp5w77a3LLWOjD73xxhCQZCALUWr~ocCOX5bTFL8X8Ivr7oRzzWL-wVKKMX4hvhTeGI9xdoorD34MNJH5QBzCMelRhVkc7n~mdBk7TofEef8PkqUwa~l-KWTTscjJPKhv~86S0i~e0OQvIJJeZ2hHQUV6eleCH2JDGAR-SvCwi0o4~x1k4aBSPGSbzkUYhZZMr-zsEZuvWkf6zZ7fxYrZWSdz3n~R0e7W8gy0NgVt7DbcXO5r9k4Hg-C~8BaNYABm~NB7DhgOjbopYkYSp3pKQgOtgW7f2SLKzC8dkSwnSslO~ZZOzdoohpZCf5opIqiu2M7u1izelSzv0sw__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA [Ziyaret Tarihi: 01 Mart 2023]
- Begel, A., & Nagappan, N. (2007). Usage and perceptions of agile software development in an industrial context: An exploratory study. *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, Madrid Spain, 255-264.
- Boehm, B., & Turner, R., 2003, Balancing agility and discipline: A guide for the perplexed[online]. Addison-Wesley Professional, Web Adresi: <https://people.eecs.ku.edu/~saiedian/811/Papers/Stu-Presentations/Agility-vs-Discipline/wishnie-agility-discipline.pdf> [Ziyaret Tarihi: 0 Ekim 2023]
- Boehm, B. W., 1988, A spiral model of software development and enhancement, *Computer*, 21(5), 61-72.
- Bolturk, E., & Kahraman, C., 2018, A novel interval-valued neutrosophic AHP with cosine similarity measure, *Soft Computing*, 22(15), 4941-4958.
- Bosch-Rekveltdt, Marian Jongkind, Yuri Mooi, Herman Bakker, & Hans Verbraeck, A., 2011, Grasping project complexity in large engineering projects: The TOE (Technical, Organizational and Environmental) framework, *International journal of project management*, 29(6), 728-739.

- Broumi, S., Nagarajan, D., Bakali, A., Talea, M., Smarandache, F., & Lathamaheswari, M., 2019, The shortest path problem in interval valued trapezoidal and triangular neutrosophic environment, *Complex & Intelligent Systems*, 5, 391-402.
- Buckley, J. J., 1985, Fuzzy hierarchical analysis, *Fuzzy sets and systems*, 17(3), 233-247.
- Butler, C. W., Vijayasathya, L. R., & Roberts, N., 2020, Managing software development projects for success: Aligning plan-and agility-based approaches to project complexity and project dynamism, *Project management journal*, 51(3), 262-277.
- Büyüközkan, G., Kahraman, C., & Ruan, D., 2004, A fuzzy multi-criteria decision approach for software development strategy selection, *International Journal of General Systems*, 33(2-3), 259-280.
- Ceylan, Z., & Gürsev, S., 2020, AHP ve TOPSIS Yöntemleri ile Bilgi Teknolojileri Projelerinde Scrum-Kanban-Şelale Uygulamaları Karşılaştırması. *Bilişim Teknolojileri Dergisi*, 13(3), 329-339.
- Chang, C. K., Jiang, H.-y., Di, Y., Zhu, D., & Ge, Y., 2008, Time-line based model for software project scheduling with genetic algorithms, *Information and Software Technology*, 50(11), 1142-1154.
- Chen, C.-T., 2000, Extensions of the TOPSIS for group decision-making under fuzzy environment, *Fuzzy sets and systems*, 114(1), 1-9.
- Chen, W.-N., & Zhang, J., 2012, Ant colony optimization for software project scheduling and staffing with an event-based scheduler, *IEEE Transactions on Software Engineering*, 39(1), 1-17.
- Chi, P., & Liu, P., 2013, An extended TOPSIS method for the multiple attribute decision making problems based on interval neutrosophic set, *Neutrosophic Sets and Systems*, 1(1), 63-70.
- Clarke, P., & O'Connor, R. V., 2012, The situational factors that affect the software development process: Towards a comprehensive reference framework, *Information and Software Technology*, 54(5), 433-447.
- Çetiner Karataş, P., 2019, Aralık değerli nütrosifik AHP ve aralık değerli nütrosifik TOPSIS yöntemleri ile personel seçimi, Yüksek Lisans Tezi, *Çankaya Üniversitesi Sosyal Bilimler Enstitüsü*, Ankara, 1-96.
- Çulha, D., 2014, An Agile business process software development methodology, Doktora Tezi, *Orta Doğu Teknik Üniversitesi Doğa ve Uygulamalı Bilimler Fakültesi Bilgisayar Mühendisliği*, Ankara, 1-183.
- de Oliveira Lacerda, R. T., Sacco Calvetti, E., Lucia Bernardes, M., & Cargnin da Silveira, J. F., 2020, Research opportunities about measuring agility in software development: A constructivist perspective. *Revista de Gestao e Projetos*, 11(2).
- Deming, W., 1998, Krizden çıkış (Çev: Cem AKAŞ). *KalDer Kalite Derneği, İstanbul*.

- Demircan, M. L., & Özcan, B., 2021, A proposed method to evaluate warehouse location for 3PL cold chain suppliers in Gulf countries using neutrosophic fuzzy EDAS, *International Journal of Computational Intelligence Systems*, 14(1), 202.
- Dennis, A., Wixom, B. H., & Roth, R. M., 2008, Systems Analysis and Design. John Wiley & sons, 10-60.
- Du, S., Ye, J., Yong, R., & Zhang, F., 2021, Some aggregation operators of neutrosophic Z-numbers and their multicriteria decision making method, *Complex & Intelligent Systems*, 7(1), 429-438.
- Ellappan, V., & Ashwini, J, 2017, Ant colony optimization and event-based dynamic task scheduling and staffing for software projects. *IOP Conference Series: Materials Science and Engineering*, Tamil Nadu India, 263(4), 1-14.
- Elmas, Ç., 2003, Bulanık mantık denetleyiciler. Seçkin Yayıncılık.
- Erickson, J., Lyytinen, K., & Siau, K., 2005, Agile modeling, agile software development, and extreme programming: the state of research, *Journal of Database Management (JDM)*, 16(4), 88-100.
- Everett, G. D., & McLeod Jr, R., 2007, *Software testing: testing across the entire software development life cycle*. John Wiley & Sons, 93-98.
- Ewusi-Mensah, K., 2003, Software Development Failures. *The MIT Press*, Cambridge-Massachusetts London England, 19-20.
- Farid, A. B., Abd Elghany, A., & Helmy, Y. M., 2016, Implementing project management category process areas of CMMI version 1.3 using scrum practices, and assets, *International Journal of Advanced Computer Science and Applications (IJACSA)*, 7(2), 243-252.
- Farrell, A. (2008). Selecting a software development methodology based on organizational characteristics, Yüksek Lisans Tezi, *Master of Science in Information Systems* Athabasca University, Alberta, 50-57.
- Fitsilis, P., 2008, Comparing PMBOK and Agile Project Management software development processes, *Advances in Computer and Information Sciences and Engineering* (pp. 378-383), Springer.
- Fowler, M., & Highsmith, J., 2001, The agile manifesto. *Software Development*, 9(8), 28-35.
- Gencer, C., & Kayacan, A., 2017, Yazılım Proje Yönetimi: Şelale Modeli ve Çevik Yöntemlerin Karşılaştırılması, *Bilişim Teknolojileri Dergisi*, 10(3), 335-352.
- Gheorghe, A.-M., Gheorghe, I. D., & Iatan, I. L., 2020, Agile Software Development, *Informatica Economica*, 24(2), 90-100.
- Ghorabae, M. K., Olfat, L., Zavadskas, E. K., & Turskis, Z., 2015, Multi-Criteria Inventory Classification Using a New Method of Evaluation Based on Distance from Average Solution (EDAS), *Informatica (Netherlands)*, 26(3), 435-451. <https://doi.org/10.15388/Informatica.2015.57>

- Gitinavard, H., 2019, Strategic evaluation of sustainable projects based on hybrid group decision analysis with incomplete information, *Journal of Quality Engineering and Production Optimization*, 4(2), 17-30.
- Gitinavard, H., Ghaderi, H., & Pishvae, M. S., 2018, Green supplier evaluation in manufacturing systems: a novel interval-valued hesitant fuzzy group outranking approach, *Soft Computing*, 22, 6441-6460.
- Gitinavard, H., Mousavi, S. M., & Vahdani, B., 2016, A new multi-criteria weighting and ranking model for group decision-making analysis based on interval-valued hesitant fuzzy sets to selection problems, *Neural Computing and Applications*, 27, 1593-1605.
- Gitinavard, H., Mousavi, S. M., & Vahdani, B., 2017, Soft computing-based new interval-valued hesitant fuzzy multi-criteria group assessment method with last aggregation to industrial decision problems, *Soft Computing*, 21, 3247-3265.
- Gupta, D., Ahlawat, A. K., & Sagar, K., 2017, Usability prediction & ranking of SDLC models using fuzzy hierarchical usability model, *Open Engineering*, 7(1), 161-168.
- Gündoğdu, F. K., & Kahraman, C., 2019, A novel fuzzy TOPSIS method using emerging interval-valued spherical fuzzy sets, *Engineering Applications of Artificial Intelligence*, 85, 307-323.
- Güven, F., 2021, Çok projeli kaynak kısıtlı yazılım geliştirme projelerinin benzetimle gösterimi ve benzetim eniyilemesi, Yüksek Lisans Tezi, *Başkent Üniversitesi Fen Bilimleri Enstitüsü*, 1-146.
- Harley, J., 2007, Public sector IT projects have only 30% success rate - CIO for Department for Work and Pensions[online]. *Computer Weekly*, Web Adresi: <https://www.computerweekly.com/blog/Public-Sector-IT/Public-sector-IT-projects-have-only-30-success-rate-CIO-for-Department-for-Work-and-Pensions> [Ziyaret Tarihi: 01 Mart 2019]
- Hastie, S., & Wojewoda, S., 2015, Standish group 2015 chaos report-q&a with jennifer lynch[online]. *InfoQ*, Web Adresi: <https://www.infoq.com/articles/standish-chaos-2015/> [Ziyaret Tarihi: 01 Ekim 2023]
- Heidary Dahooie, J., Beheshti Jazan Abadi, E., Vanaki, A. S., & Firoozfar, H. R., 2018, Competency-based IT personnel selection using a hybrid SWARA and ARAS-G methodology *Human Factors and Ergonomics in Manufacturing & Service Industries*, 28(1), 5-16.
- Heinsman, H., 2008, The Competency Concept Revealed: Its Nature, Relevance, and Practice, Doktora Tezi, *Academisch Proefschrift, Vrije Üniversitesi*, PrintPartners Ipskamp BV, Enschede. In..
- Hickey, A. M., & Davis, A. M., 2003, Requirements elicitation and elicitation technique selection: model for two knowledge-intensive software development processes.

36th Annual Hawaii International Conference on System Sciences, 2003, Big Island, HI, USA, 1-10.

- Highsmith, J., & Cockburn, A., 2001, Agile software development: The business of innovation. *Computer*, 34(9), 120-127.
- Ikonen, M., Pirinen, E., Fagerholm, F., Kettunen, P., & Abrahamsson, P., 2011, On the impact of Kanban on software project work: An empirical case study investigation, 2011 16th IEEE international conference on engineering of complex computer systems, Las Vegas, NV, USA, 305-314.
- Iqbal, J., Ahmad, R. B., Khan, M., Alyahya, S., Nizam Nasir, M. H., Akhunzada, A., & Shoaib, M., 2020, Requirements engineering issues causing software development outsourcing failure, *PloS one*, 15(4), e0229785, 1-36.
- Jiang, W., Zhang, Z., & Deng, X., 2019, Multi-attribute decision making method based on aggregated neutrosophic set, *Symmetry*, 11(2)-267, 1-13.
- Karayel, A., 2023, Şelale Yöntemi Waterfall Modeli[online], *aspmvc.net*, Web Adresi: <http://www.aspmvcnet.com/tr/m/yazilim-muhendisligi/selale-yontemi-waterfall-modeli.html> [Ziyaret Tarihi: 01 Ekim 2023]
- Kayacan, A., 2017, Şelale modeli ve çevik yöntemlerin yazılım proje yönetimi bakış açısıyla incelenmesi ve karşılaştırılması, *Bilişim Teknolojileri Dergisi*, 10(3), 335-352.
- Khan, A. A., Shameem, M., Kumar, R. R., Hussain, S., & Yan, X., 2019, Fuzzy AHP based prioritization and taxonomy of software process improvement success factors in global software development, *Applied Soft Computing*, 83, 105648, 1-24.
- Khan, M. A., Parveen, A., & Sadiq, M., 2014, A method for the selection of software development life cycle models using analytic hierarchy process, 2014 *International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, Ghaziabad India, 534-540.
- Khan, P., & Beg, M. S., 2013, Extended decision support matrix for selection of sdlc-models on traditional and agile software development projects, 2013 *Third International Conference on Advanced Computing and Communication Technologies (ACCT)*, Rohtak India, 8-15.
- Khatibi, E., & Bardsiri, V. K., 2015, Model to estimate the software development effort based on in-depth analysis of project attributes, *IET Software (Institution of Engineering & Technology)*, 9(4), 109-118. <https://doi.org/10.1049/iet-sen.2014.0169>
- Korkmaz, A. C., 2021, Identifying management errors that lead to failure in it projects and recommendations for project managers, Yüksek Lisans Tezi, *Doğa ve Uygulamalı Bilimler Çankaya Üniversitesi Bilgisayar Mühendisliği*. Ankara, 1-82.

- Kutlu Gündoğdu, F., & Kahraman, C., 2019, Spherical fuzzy sets and spherical fuzzy TOPSIS method, *Journal of intelligent & fuzzy systems*, 36(1), 337-352.
- Kutlu Gündoğdu, F., & Kahraman, C., 2020, A novel spherical fuzzy analytic hierarchy process and its renewable energy application, *Soft Computing*, 24(6), 4607-4621.
- Leau, Y. B., Loo, W. K., Tham, W. Y., & Tan, S. F., 2012, Software development life cycle AGILE vs traditional approaches, *International Conference on Information and Network Technology*, Singapore, 162-167.
- Lebdeh, L. A., Qasim, A., & Kharbat, F., 2020, Implementing Agility in Large Software Development Projects, *TEM Journal*, 9(3), 1285-1294.
- Lei, H., Ganjeizadeh, F., Jayachandran, P. K., & Ozcan, P., 2017, A statistical analysis of the effects of Scrum and Kanban on software development projects, *Robotics and Computer-Integrated Manufacturing*, 43, 59-67.
- Li, J., & Eberlein, A., 2007, Selecting Requirements Engineering Techniques Based on Project Attributes--A Case Study, *14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'07)*, Tucson AZ USA, 269-278.
- Liu, R., & Zhai, F., 2007, Multi-Attribute Evaluation for Software Project Risk, *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, Shangai China, 4942-4945.
- Liu, Y., Eckert, C. M., & Earl, C., 2020, A review of fuzzy AHP methods for decision-making with subjective judgements, *Expert Systems with Applications*, 161, 113738, 1-30.
- Luna, F., González-Álvarez, D. L., Chicano, F., & Vega-Rodríguez, M. A., 2011, On the scalability of multi-objective metaheuristics for the software scheduling problem, *2011 11th International Conference on Intelligent Systems Design and Applications*, Cordoba Spain, 1110-1115.
- Mandal, U., & Seikh, M. R., 2022, Interval-valued Fermatean Fuzzy TOPSIS Method and Its Application to Sustainable Development Program, *Congress on Intelligent Systems: Proceedings of CIS 2021*, Bengaluru India, Volume 2, 783-796.
- Mandal, U., & Seikh, M. R., 2023, Interval-valued spherical fuzzy MABAC method based on Dombi aggregation operators with unknown attribute weights to select plastic waste management process, *Applied Soft Computing*, 145 110516, 1-26.
- Maylor, H., Vidgen, R., & Carver, S., 2008, Managerial complexity in project-based operations: A grounded model and its implications for practice, *Project management journal*, 39(1_suppl), S15-S26.
- Minku, L. L., Sudholt, D., & Yao, X., 2013, Improved evolutionary algorithm design for the project scheduling problem based on runtime analysis, *IEEE Transactions on Software Engineering*, 40(1), 83-102.

- Mintzberg, H., 1989, Mintzberg on management: Inside our strange world of organizations. *Simon and Schuster Inc.*, 110-111.
- Mnkandla, E., 2008, A selection framework for agile methodology practices: A family of methodologies approach, Doktora Tezi, *Faculty of Engineering and the Built Environment, University of The Witwatersrand*, Johannesburg South Africa, 1-247.
- Mnkandla, E., & Dwolatzky, B. (2007). Agile methodologies selection toolbox, *International Conference on Software Engineering Advances (ICSEA 2007)*, Cap Esterel France, 1-6.
- Molina Chalacan, L. J., Checa Cabrera, M. A., Aguirre Paz, L. M., Lalama Flores, R. V., & Martín Rabelo, A., 2021, Selection of Criteria for the Prioritization of Information Technology Services in a Neutrosophic Environment, *Neutrosophic Sets and Systems*, 44(1), 41.
- Moustafaev, J., 2017, How to Categorize IT and Software Development Requirements[online], *Thinktank Consulting*, Web Adresi: <https://www.thinktankconsulting.ca/how-to-categorize-it-and-software-development-requirements> [Ziyaret Tarihi: 01 Ekim 2023]
- Munassar, N. M. A., & Govardhan, A., 2010, A comparison between five models of software engineering, *International Journal of Computer Science Issues (IJCSI)*, 7(5), 94-101.
- Özdemir, H., 2022, Proje Yönetimi Yaklaşımının Matematiksel Yöntemlerle Seçilmesi ve Uygulanması, Yüksek Lisans Tezi, *Fen Bilimleri Enstitüsü Yıldız Teknik Üniversitesi*, İstanbul, 1-68.
- Pabuccu, Y. U., Yel, I., Helvacioğlu, A. B., & Asa, B. N., 2022, The Requirement Cube: A Requirement Template for Business, User, and Functional Requirements With 5W1H Approach, *International Journal of Information System Modeling and Design (IJISMD)*, 13(1), 1-18.
- Peker, K., 201, SDLC Nedir Modelleri Nelerdir[online]. Web Adresi: <http://koraypeker.com/2017/12/31/sdlc-nedir-modelleri-nelerdir/> [Ziyaret Tarihi 01 Ekim 2023]
- PMI, 2017, A Guide To The Project Management Body Of Knowledge (PMBOK-Guide). *Project Management Institute*.
- Polito, T., Watson, K., & Berry, R., 2001, An exploratory identification of differences between Deming's PDSA improvement cycle and the system development life cycle (SDLC), *Proceedings of the 2001 Annual Meeting of the Midwest Region of Decision Sciences Institute*. Dearborn, Michigan. April 26-28, 2001.
- Ponce Ruiz, D. V., Díaz Vásquez, R. A., Villalta Jadan, B. E., & Dorado Caballos, C. Y., 2021, Neutrosophic Statistics in the Strategic Planning of Information Systems. *Neutrosophic Sets and Systems*, 44(1), 44.

- Radack, S., 2009, The system development life cycle (sdlc)[online]. *Computer Security Resource Center*, 1-7.
- Rani, S. B. A. S. U., 2017, A detailed study of Software Development Life Cycle (SDLC) models, *International Journal Of Engineering And Computer Science*, 6(7), 22097 - 22100.
- Ruparelia, N. B., 2010, Software development lifecycle models, *ACM SIGSOFT Software Engineering Notes*, 35(3), 8-13.
- Salah, A., Darwish, N. R., & Hefny, H. A., 2017, Towards a Hybrid Approach for Software Project Management using Ontology Alignment, *International Journal of Computer Applications*, 168(6), 12-19.
- Sarıdoğan, M., 2004, Yazılım Mühendisliği, *Papatya Yayıncılık*.
- Sarker, I. H., Faruque, F., Hossen, U., & Rahman, A., 2015, A survey of software development process models in software engineering, *International Journal of Software Engineering and Its Applications*, 9(11), 55-70.
- Seikh, M. R., & Mandal, U., 2022, Multiple attribute group decision making based on quasirung orthopair fuzzy sets: Application to electric vehicle charging station site selection problem, *Engineering Applications of Artificial Intelligence*, 115, 105299, 1-16.
- Seikh, M. R., & Mandal, U., 2023, Interval-valued Fermatean fuzzy Dombi aggregation operators and SWARA based PROMETHEE II method to bio-medical waste management, *Expert Systems with Applications*, 226, 120082, 1-27.
- Seker, S. E., 2015, Yazılım geliştirme modelleri ve sistem/yazılım yaşam döngüsü, "Software development models and system/software lifecycle", *YBS Ansiklopedi*, 2(3), 18-29.
- 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.
- Shen, X., Minku, L. L., Bahsoon, R., & Yao, X., 2015, Dynamic software project scheduling through a proactive-rescheduling method, *IEEE Transactions on Software Engineering*, 42(7), 658-686.
- Shen, X. N., Minku, L. L., Marturi, N., Guo, Y.-N., & Han, Y., 2018, A Q-learning-based memetic algorithm for multi-objective dynamic software project scheduling, *Information Sciences*, 428, 1-29. <https://doi.org/10.1016/j.ins.2017.10.041>
- Singh, S., & Saikia, L., 2016, Formal Methods and its Importance in Minimizing Ambiguity in the Requirement Engineering Document Phase of SDLC, *Journal of Applied Information Science, ISSN*, 4(2), 2321-6115.

- Sinha, A., & Das, P., 2021, Agile methodology Vs. traditional waterfall SDLC: A case study on quality assurance process in software industry, *2021 5th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*, Kolkata India, 1-4.
- Smarandache, F., 1998, Neutrosophy: neutrosophic probability, set, and logic: analytic synthesis & synthetic analysis, *American Research Press*, 1-105.
- Špundak, M., 2012, Defining a Conceptual Categorization Model for Agile Software Development Projects. *Procedia - Social and Behavioral Sciences*, 119, 939-948.
- Špundak, M. (2014). Mixed agile/traditional project management methodology–reality or illusion? *Procedia-Social and Behavioral Sciences*, 119, 939-948.
- Stević, Ž., Vasiljević, M., Puška, A., Tanackov, I., Junevičius, R., & Vesković, S., 2019, Evaluation of Suppliers Under Uncertainty: A Multiphase Approach Based on Fuzzy Ahp And Fuzzy Edas, *Transport (16484142)*, 34(1), 52-66. <https://doi.org/10.3846/transport.2019.7275>
- Taş, A., & Karataş, P. Ç., 2021, Yazılım Sektöründe Nitelikli Personel Seçiminin Nötrosofik AHP ve TOPSİS Yöntemleri İle İncelenmesi, *İşletme Araştırmaları Dergisi*, 13(1), 969-979.
- Tavakkoli-Moghaddam, R., Gitinavard, H., Mousavi, S. M., & Siadat, A., 2015, An interval-valued hesitant fuzzy TOPSIS method to determine the criteria weights, *Outlooks and Insights on Group Decision and Negotiation: 15th International Conference*, GDN 2015, Warsaw, Poland, June 22-26, 157-169.
- Turk, D., France, R., & Rumpe, B., 2002, Limitations of agile software processes, *Proceedings of the Third International Conference on eXtreme Programming and Agile Processes in Software Engineering*, Alghero Italy, 43-46.
- Turskis, Z., Zavadskas, E. K., Antucheviciene, J., & Kosareva, N., 2015, A Hybrid Model Based on Fuzzy AHP and Fuzzy WASPAS for Construction Site Selection, *International Journal of Computers, Communications & Control*, 10(6), 873-888.
- Ulutaş, A., Özkan, A. M., & Tağraf, H., 2018, Bulanık Analitik Hiyerarşi Süreci ve Bulanık Gri İlişkisel Analizi Yöntemleri Kullanılarak Personel Seçimi Yapılması, *Electronic Journal of Social Sciences*, 17(65), 223-232.
- Vafadarnikjoo, A., Tavana, M., Botelho, T., & Chalvatzis, K., 2020, A neutrosophic enhanced best–worst method for considering decision-makers’ confidence in the best and worst criteria, *Annals of Operations Research*, 289(2), 391-418.
- Vijayarathy, L. R., & Butler, C. W., 2015, Choice of software development methodologies: Do organizational, project, and team characteristics matter? *IEEE software*, 33(5), 86-94.
- Whitty, S. J., & Maylor, H., 2009, And then came complex project management (revised), *International journal of project management*, 27(3), 304-310.

- Xiao, J., Ao, X.-T., & Tang, Y., 2013, Solving software project scheduling problems with ant colony optimization, *Computers & Operations Research*, 40(1), 33-46.
- Ye, J., 2014, Multiple attribute group decision-making method with completely unknown weights based on similarity measures under single valued neutrosophic environment, *Journal of intelligent & fuzzy systems*, 27(6), 2927-2935.
- Ye, J., 2021, Similarity measures based on the generalized distance of neutrosophic Z-number sets and their multi-attribute decision making method, *Soft Computing*, 25(22), 13975-13985.
- Yel, İ., & Engin, O., 2022, Process Improvement for Error Records in a Financial Institution with the Analysis of Simulation on Using Value Stream Mapping, *Journal of Advanced Research in Natural and Applied Sciences*, 8(2), 171-187.
- Yel, İ., 2019, Project and Analyst and Software Engineer Ranking and Job Assignment Problem Solution in Software Development Projects with Fuzzy Multi Criteria Decision Making Techniques, International Conference on Intelligent and Fuzzy Systems, Istanbul Turkey.
- Yel, İ., & Baysal, M. E., 2023, Yazılım projeleri geliştirme süreci seçimi için bulanık çok kriterli karar verme yöntemlerinin kullanımına ilişkin bir uygulama, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 38 (2023), 2325-2338. <https://doi.org/10.17341/gazimmfd.1132638>
- Yel, İ., Baysal, M. E., & Sarucan, A., 2023, A new approach to developing software projects by assigning teams to projects with interval-valued neutrosophic Z numbers, *Engineering Applications of Artificial Intelligence*, 126, 106984, 1-16. <https://doi.org/https://doi.org/10.1016/j.engappai.2023.106984>
- Yel, İ., Sarucan, A., & Baysal, M. E., 2021, An Application of Fuzzy AHP, EDAS and WASPAS for the Selection of Process Method in Software Projects. *International Conference on Intelligent and Fuzzy Systems*, Online Conference.
- Yong, R., Ye, J., & Du, S., 2021, Multicriteria Decision-Making Method and Application in the Setting of Trapezoidal Neutrosophic Z-Numbers. *Journal of Mathematics*, 2021, 1-13.
- Zadeh, L. A., 1965, Fuzzy sets. *Information and control*, 8(3), 338-353.
- Zadeh, L. A., 2008, Is there a need for fuzzy logic? *Information Sciences*, 178(13), 2751-2779.
- Zadeh, L. A., 2011, A note on Z-numbers, *Information Sciences*, 181(14), 2923-2932.
- Zavadskas, E. K., Turskis, Z., Antucheviciene, J., & Zakarevicius, A., 2012, Optimization of Weighted Aggregated Sum Product Assessment, *Svorinio agreguoto alternatyvių sprendimų vertinimo optimizavimas*.6(122), 3-6.

Zayat, W., & Senvar, O., 2020, Framework study for agile software development via scrum and Kanban, *International journal of innovation and technology management*, 17(04), 2030002, 1-24.

