

Büyük Dil Modelleri ile Otomatik Gereksinim Analizi

¹Ayça Coşkun *^{1,2}Selami Bağrıyanık

¹Wingie Enuygun Group, ARGE Merkezi, Türkiye

²Bahçeşehir Üniversitesi Lisansüstü Eğitim Enstitüsü, Türkiye

Özet

Gereksinim analizi, yazılım geliştirme süreçlerinde kritik bir aşamadır ve yetenek açığı ve yapısal dokümantasyon eksikliği gibi zorluklarla karşı karşıyadır. Çevik yöntemlerin benimsenmesiyle birlikte, bu zorluklar daha da önemli hale gelmiştir çünkü çevik yöntemler, hızlı geri bildirim döngüleri ve değişebilen gereksinimler ile uyumlu hale gelmeyi gerektirir. Bu çalışmada, var olan bir büyük dil modeli, gereksinim mühendisliği alanına özelleştirilmiş ve başarıyla uygulanmıştır. Model, doğal dil işleme teknikleri ve büyük dil modeli mimarisi kullanarak, gereksinimlerin olgunlaştırılmasını, analizini ve dokümantasyonunu sistematik hale getirmeyi amaçlamaktadır. Yapılan ilk testlerde, gereksinim analiz süreçlerinin hızlandığı ve gereksinim içerik kalitesinin arttığı gözlemlenmiştir. Elde edilen bulgular, büyük dil modeli tabanlı çözümlerin gereksinim analizi süreçlerini daha verimli hale getirme potansiyeline sahip olduğunu göstermektedir. Bu çalışma, gereksinim analizi süreçlerinde verimliliği artıran yeni bir yaklaşım sunmuştur; gelecekte daha geniş ölçekli uygulamalarla metodolojinin etkinliği daha kapsamlı bir şekilde değerlendirilecektir.

Anahtar kelimeler: Gereksinim analizi, büyük dil modeli, GPT, use case, yapay zeka

Abstract

Requirement analysis is a critical stage in software development processes, facing challenges such as skills gap and lack of structured documentation. With the adoption of agile methods, these challenges have become even more significant, as agile approaches require alignment with fast feedback loops and changing requirements. In this study, an existing large language model has been customized and successfully applied to the field of requirements engineering. The model aims to systematize the elicitation, analysis, and documentation of requirements using natural language processing techniques and large language model architecture. Initial tests have shown that the aforementioned processes have been accelerated, and the quality of requirement content has improved. The findings suggest that solutions based on large language models have the potential to make requirement analysis processes more efficient. The effectiveness of the methodology will be evaluated more comprehensively in future, adopted in large-scale applications.

Key words: Requirements analysis, large language models, GPT, use case, artificial intelligence

1. Giriş

Gereksinim mühendisliği, yazılım mühendisliğinin temel süreçlerinden biridir ve bir yazılım sisteminin başarısını doğrudan belirleyen kritik bir alt disiplindir; bu etkinlik, tüm ilgili paydaşların beklentilerini yazılım gereksinimlerine dönüştürmeyi amaçlar [1] ve genellikle yetenekli ve tecrübeli uzman eksikliği ve yapısal dokümantasyon eksikliği gibi zorluklarla karşı karşıyadır [2]. Türkiye ekosisteminde yapılan bir çalışmada en fazla güçlükle karşılaşılan safha

*Sorumlu Yazar: Adres: Wingie Enuygun Group, Küçükbakkalköy Mah. Kayışdağı Cd. Allianz Tower 1/16 34750 Ataşehir İstanbul E-mail adresi: selami.bagriyanik@ou.bau.edu.tr, Tel: +90 850 333 88 88

olarak gereksinim analizi süreci göze çarpmakta ve pratisyenlerin çoğunun geleneksel metin yöntemi ile gereksinimleri kayıt altına aldığı görülmektedir [3]. Bu durumun doğal bir uzantısı olarak gereksinimlerin doğruluğunu ve tutarlılığını sağlamak için yoğun uzman müdahalesi ve manuel süreçler gerekmektedir. Aynı durum yazılım projelerinde zaman kayıpları, gereksinim değişikliklerinin yönetilmesinde zorluklar ve hatalı gereksinimlerin geliştirme sürecine aktarılması gibi sonuçlara yol açabilmektedir.

21. yüzyılın başlarından itibaren ve özellikle son on senede yazılım geliştirme süreçlerinde çevik yöntemlerin yaygın olarak benimsenmesi, gereksinim analizi süreçlerini daha da kırılgan ve karmaşık hale getirmiştir. Çevik yöntemler, değişen müşteri ihtiyaçlarına ve pazar koşullarına hızlı uyum sağlayabilmek için kısa iterasyonlarla çalışan, sürekli geri bildirim mekanizmaları içeren ve esnek gereksinim yönetimi gerektiren bir yapıya sahiptir [4]. Ancak, çevik geliştirme süreçlerinin uygulandığı ortamlarda, özellikle büyük ölçekli projelerde, gereksinimlerin sürekli değişmesi ve belirsizliğin yüksek olması, gereksinim dokümantasyonu ve yönetimini daha da zorlaştırmış ve gereksinimlerin ve dolayısıyla geliştirilen sistemin niteliklerinde kayıplara yol açmıştır [5]. Tablo 1’de gereksinim mühendisliğinin son 40 sene içerisindeki gelişimi, iki önemli konu uzmanı araştırmacı ile yapılan bir tartışma [5] baz alınarak gösterilmiştir.

Tablo 1. Gereksinim Mühendisliğinin 40 Yıllık Gelişimi [5]

Dönem	Ana Odak Noktası	Öne Çıkan Gelişmeler
1970’ler 1980’ler	– Belgelendirme & Şablonlar	- Gereksinim mühendisliği disiplininin ortaya çıkışı. - IEEE 830 gibi yazılım gereksinim şablonlarının geliştirilmesi. - ABD hükümet kurumlarının SGML tabanlı elektronik dokümantasyona geçişi.
1990’lar	Modelleme & Yapay Zekâ Uygulamaları	- Use case yaklaşımı ve UML’in doğuşu. - Yapay zekâ tekniklerinin gereksinim analizinde kullanılmaya başlanması. - İlk akademik gereksinim mühendisliği konferanslarının (RE, ICRE, REFSQ) düzenlenmesi. - Bilgi mühendisliği ve doğal dil işleme (NLP) teknikleriyle gereksinimlerin analiz edilmesi.

2000'ler	Fonksiyonel Gereksinimler & Güvenlik	Olmayan	- Fonksiyonel Olmayan Gereksinim Mühendisliği alanının yükselişi. - Sistem kalitesi, güvenlik ve kullanılabilirlik gibi kriterlerin gereksinim analizine dahil edilmesi. - Gereksinim yönetimi ve önceliklendirme tekniklerinin geliştirilmesi.
2010'lar	Çevik Dönüşüm & DevOps		- Çevik yöntemlerin (Agile) yaygınlaşmasıyla geleneksel gereksinim mühendisliğinin "hafif" bir versiyonuna dönüşmesi. - Kullanıcı hikayeleri (User story) ve İş listesi (Backlog) yönetimi gibi yeni yaklaşımların ortaya çıkışı. - Büyük ölçekli firmaların gereksinim yönetim araçlarına (DOORS vb.) yaptığı yatırımlar. - DevOps ve sürekli entegrasyon süreçlerinin gereksinim yönetimine etkisi.
2020'ler sonrası	Büyük Dil Modelleri (LLM) & Yapay Zekâ Destekli Gereksinim Analizi		- LLM'lerin gereksinim mühendisliği süreçlerine entegrasyonu. - Otomatik gereksinim çıkarımı, analiz ve test üretimi için yapay zekâ tabanlı sistemlerin kullanımı. - Veri odaklı gereksinim yönetimi ve kullanıcı geri bildirimlerinin doğrudan gereksinim süreçlerine entegre edilmesi.

Son dönemde, büyük dil modelleri (BDM) ve doğal dil işleme (NLP) teknikleri, yazılım mühendisliği alanında yeni fırsatlar sunmaktadır. Büyük dil modelleri, büyük veri kümeleri üzerinde eğitilmiştir ve doğal dili işleyerek metin anlama, analiz etme, oluşturma ve dönüştürme gibi yeteneklere sahiptir [6]. Bu yeteneklerin, gereksinim mühendisliği süreçlerinde otomatik gereksinim dokümantasyonu oluşturma, gereksinimlerin çeşitlendirilmesi, paydaş katkısının alınması, gibi alanlarda fayda potansiyeli yüksektir [6]. Bu çalışmamızda, var olan büyük dil modeli sağlayıcılarından olan OpenAI'nin özelleştirilebilen GPT model özelliği [7] kullanılarak gereksinim mühendisliği alanına özelleştirilmiş bir model tasarlanmış ve ticari bir organizasyonda uygulanmıştır. Elde edilen bulgular, bahsedilen çözümün gereksinim mühendisliği alanında verimliliği artırma ve sürdürülebilir standart bir dokümantasyon sağlama potansiyeline sahip olduğunu ortaya koymaktadır. Çalışmanın literatüre en önemli katkısı, bu

alandaki yayımlanmış olan az sayıda deneysel çalışmalardan biri olmasıdır.

Konferans bildirisinin bundan sonraki kısımları şu şekilde yapılandırılmıştır: Sonraki bölümde konuyla ilişkili önceki çalışmalara yer verilmiştir. Daha sonra, önerilen model ve değerlendirmesi sunulmuştur. Son olarak, sonuçlar bölümü gelecek çalışmaları kapsayacak şekilde verilerek bildiri tamamlanmıştır.

2. İlgili Çalışmalar

Gereksinimlerin çoğunlukla metin olarak kayıt altına alınması pratiği büyük dil modellerinin en güçlü yönü olan doğal dil işleme özelliği ile örtüştüğü için bu alanda kullanım potansiyellerini görünür kılmıştır. Yakın zamanda yapılan bir vaka çalışması, özelleştirilmiş bir büyük dil modelinin karmaşık kullanıcı gereksinimlerinden hareketle tasarım ve kod çözümleri üretebildiğini ortaya koymaktadır [8]. Söz konusu araştırmada, büyük dil modellerinin yazılım geliştirme süreçlerindeki verimliliği artırma potansiyelini vurgulanmakta ve gelecekte farklı gereksinim tanımlama teknikleri ile yazılım tasarım metodolojilerinin daha geniş bir perspektiften değerlendirilmesi önerilmektedir. Bahsi geçen araştırmada da tespit edildiği üzere alana özgü gereksinim ihtiyaçları gerek içerik gerekse de gereksinim tanımlama yöntemi açısından farklılaştığı için büyük dil modeli özelleştirilirken bu bilgilerin modele tanıtılması gereklidir. Bir başka çalışmada, bu sorun bağlamsal anlama olarak isimlendirilmiş ve GPT-3.5'in gereksinim toplama sürecinde anket ve mülakat soruları oluşturma gibi görevlerde yüksek düzeyde okunabilirlik, açıklık ve tutarlılık sağlamasına rağmen bağlamsal anlayış ve alan bilgisinde bazı sınırlamalar taşıdığı belirtilmiştir [9]. Bu konuya dair bir örnek olarak çevik ortam içinde yapılan bir çalışmada GPT-3.5'in doğal dil işleme yeteneklerinden yararlanarak yazılım gereksinim metinlerini otomatik olarak kullanıcı hikâyelerine dönüştüren bir yapay zeka destekli yaklaşımda farklı sufle (prompt) mühendisliği teknikleri kullanılarak, gereksinimlerin daha verimli bir şekilde kullanıcı hikâyelerine dönüştürüldüğü ve bu sayede sürecin hızlandırıldığı gösterilmiştir [10]. Yaptığımız çalışmada önceki çalışmalardan farklı olarak özelleştirme esnasında çevrimiçi seyahat iş alanı ve use case şablonu kullanılıp bağlam sorununa çözüm aranmış, sonuçlar paylaşılarak literatüre katkı sağlanmıştır.

Büyük dil modelleri, yazılım mühendisliği alanında önemli bir etki yaratıp, gereksinim mühendisliği süreçlerinde de giderek daha fazla kullanılacak olsa da aynı zamanda büyük dil modellerinin önyargı, yanlış bilgi üretme (halüsinasyon), açıklanabilirlik eksikliği, saldırılara karşı zayıflık, mantıksal hatalar ve aşırı bağımlılık gibi kritik riskler taşıdığı da kapsamlı bir literatür araştırmasında [11] vurgulanmaktadır. Çalışmamızda bu tür riskleri minimize etmek amacıyla, uzman kişi ve model etkileşimi ile simbiyotik bir üretim ve gözden geçirme süreci kurgulanmıştır. Gereksinim analizi işinin tamamen Yapay Zeka (YZ) modeline havale edilmesi yerine insan - YZ işbirliği yaklaşımı benimsenmiştir. Bu yaklaşımın izlenmesi gerektiği, özellikle GPT-4'ün, yüksek doğruluklu sonuçlar sağladığı ancak, modelin yanlılık ve hata yapma eğilimleri nedeniyle insan geri bildiriminin ve insan-makine işbirliğinin kritik olduğunun vurgulandığı bir başka araştırmada da belirtilmiştir [12].

Gereksinim mühendisliği ve sufle mühendisliğinin kesişim noktası olan doğru ve etkili sufle

tasarımının, geleneksel gereksinim analiz teknikleri ile nasıl entegre edilebileceğini araştırılması, üretici yapay zekânın gereksinim analizi süreçlerinde verimli kullanılabilmesi için kritiktir [13]. Bu nedenle çalışmamızdaki model özelleştirilirken durumsal bilgilerin sufler aracılığıyla verilmesi çözümün omurgasını oluşturmuştur.

3. Önerilen Model

Araştırmanın yapıldığı kurum, Türkiye'nin önde gelen çevrimiçi seyahat acentesi ve teknoloji şirketlerinden biri olan Wingie Enuygun Group'tur [14]. Yurtdışında Wingie, Türkiye'de ise Enuygun markası altında faaliyet gösteren firma, çevrimiçi seyahat acenteliği ve uçuş arama motorları gibi hizmetler sunarak, müşterilerinin seyahat planlamalarını daha hızlı, verimli ve maliyet etkin bir şekilde yapmalarını sağlar. Aynı zamanda bir ARGE merkezi de olan firma, teknoloji ekibiyle birlikte geniş bir çalışan kadrosuna sahip olup, iş süreçlerinde çevik yöntemleri benimsemiş bir organizasyon kültürüne sahiptir. Şirketin bu çalışmayı desteklerken temel amacı, ürün yönetimi fonksiyonlarından biri olan gereksinim analizi etkinliklerinin daha hızlı, verimli ve etkili hale getirilmesi ve bu süreçlerin standart bir dokümantasyonla kayıt altına alınmasının garanti edilmesidir. Bu sayede, gereksinimlerin net bir şekilde belirlenmesi, izlenebilirliğin artırılması, süreçlerin şeffaf bir şekilde yönetilmesi ve geliştirme ekiplerine daha hızlı ve kaliteli gereksinimlerin iletilmesi hedeflenmektedir.

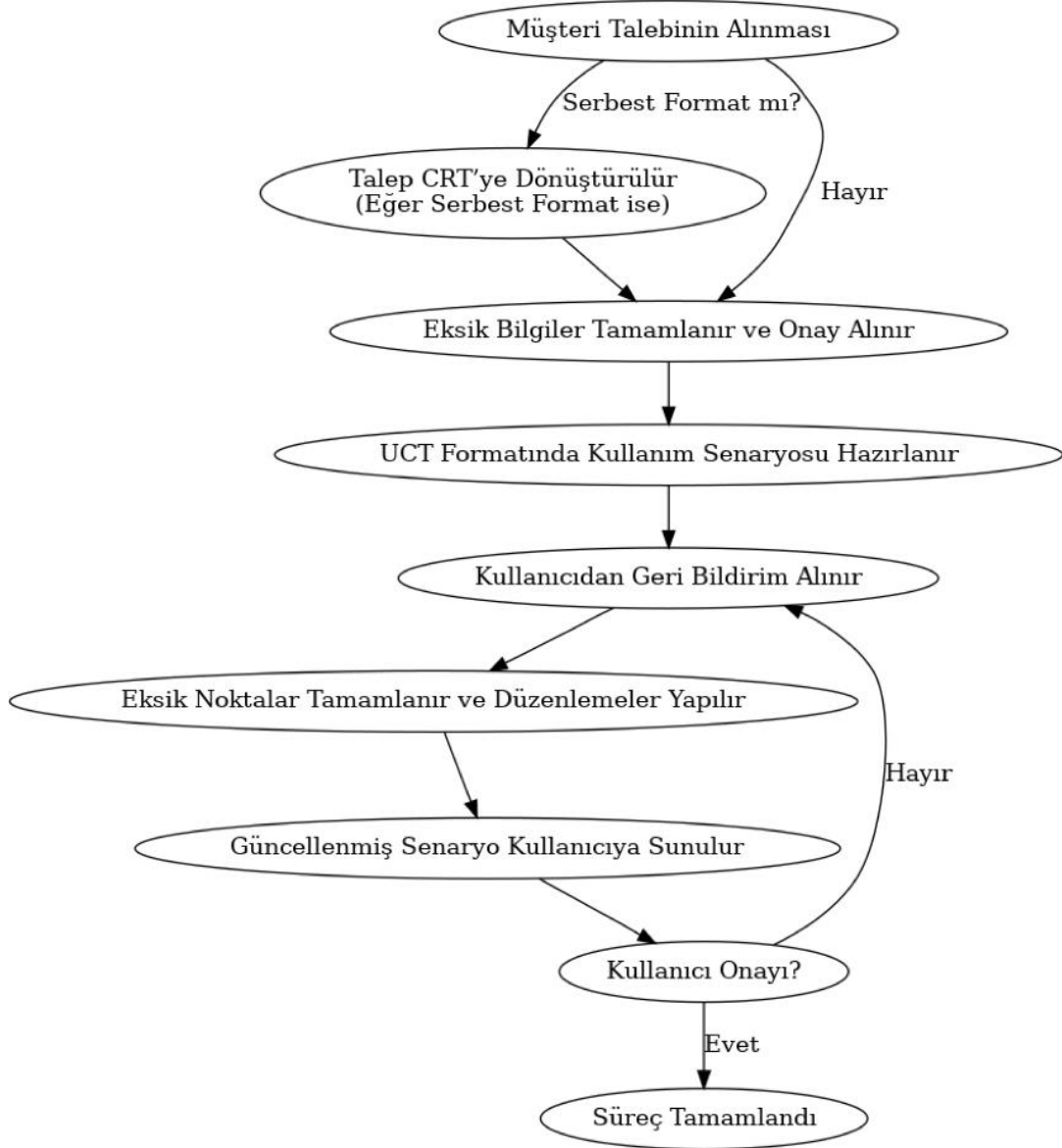
Öncelikle büyük dil modelinin özelleştirilebilmesi için kullanılacak olan ortam olarak OpenAI tarafından sunulan özelleştirilmiş GPT (Custom GPT) [7] özelliği seçilmiştir. Bu özelliğin, modellerin özelleştirilmiş eğitimi (fine-tuning) [15] yönteminden farklı olduğunu belirtmekte fayda vardır. İlk yöntemin en önemli faydası alan uzmanlığı verilerek belli konularda spesifik uzmanlıkların daha kolay bir şekilde, sadece bir ön yüz aracılığıyla geliştirilebilmesidir. Söz konusu ürün özelliği kullanılarak özelleştirilmiş bir modeli geliştirip kullanabilmek için şu temel bilgilerin OpenAI arayüzü ile sağlanması gerekmiştir:

i. İsim: Modelin ne işe yaradığının kullanıcıları tarafından anlaşılması için kısa bir isim verilmektedir. Bu araştırma kapsamında oluşturulan modelin ismi “Ürün Yöneticileri için Use Case Stüdyosu” olarak belirlenmiştir.

ii. Açıklama: Benzer şekilde kullanıcılar için modelin işlevine dair daha detaylı bir açıklama verilmiştir. Kullanıcılar bu modeli açıp kullanmadan önce detaylı açıklamayı görebilmektedir.

iii. Yönerge: ChatGPT [16] gibi büyük dil modeli tabanlı sohbet robotlarındaki suflere benzeyen bu alanda, alan uzmanlığına dair temel bilgiler ve eşlik eden ilave metin bilgilerinin verilmesi sağlanmıştır. Alana dair bilgilerin model tarafından öğrenilmesi en kritik konuların başında geldiği için sınırlı sayıda karakter kullanarak en hayati bilgilerin verilmesi gerekmektedir. Bu çalışmada, bizim sağladığımız temel bilgiler şu şekilde özetlenebilir: a) Gereksinim mühendisliği rolü ve temel varlık amacı b) Hangi kurum bağlamında çalışacağı c) Kullanıcı ile etkileşimin nasıl olacağının detaylı tarifini içeren iş akışı adımları d) Kurum bağlamında hangi kullanıcı talep gereksinim ve use case şablonlarının kullanılacağı. Üçüncü maddede belirtilen iş akış diyagramı Şekil 1’de gösterilmiştir.

iv. Sohbet Başlatıcılar: İsteğe bağlı olan bu alanda, kullanıcı ile sohbeti başlatmak için kullanılabilecek kalıp metinler girilebilir.



Şekil 1. Özelleştirilmiş GPT Akış Diyagramı

v. Bilgi: Yukarıda açıklanmış olan “Yönerge” alanında atıf yapılmış olan ve eşlik edecek olan dokümanlar da ilave bilgi olarak arayüzden yüklenmiştir. Bu araştırmada doküman olarak iki şablon tasarlanmıştır. Bu şablonlardan birincisi, firmadaki ürün yönetimi ve iş analizi süreçlerine ilk girdi olacak “Müşteri Talep Şablonu” (CRT: Customer Request Template) ve detaylı iş analizinin kayıt altına alınacağı “Use Case Şablonu”dur (UCT: Use Case Template). Bu şablonlar oluşturulurken, gereksinim mühendisliği literatüründeki [17, 18, 19] bazı genel şablonlardan

faýdalanılmış ve şirkete özel ihtiyaçlarla özelleştirilmişlerdir.

vi. Yetkinlikler: Burada özelleştirilmiş olan modelin belli yetkinliklere sahip olup olmayacağı belirlenmektedir. Araştırmamız kapsamında geliştirilen modelin Web arama, görsel üretimi, kod yorumlama ve veri analizi özelliklerinin olması sağlanacak şekilde konfigürasyon yapılmıştır. Bunun nedeni, gereksinimler Use Case formatında kayıt altına alınırken görselleştirme ihtiyacı olabileceği veya belli sözde kod yapılarının da (Log, XML yapıları) kullanılabilecek olmasıdır. Ayrıca gereksinimlerin olgunlaştırılması sürecinde Web’den gerçek zamanlı olarak firmanın web veya mobil arayüzlerine erişim de gerekebilir.

4. Değerlendirme

Wingie Enuygun Group'ta Ürün Yöneticisi rolünde çalışan, alanında uzmanlaşmış bir PM ve ikinci yazar olan araştırmacı görüşü değerlendirmelerde baz alınmıştır. Ürün Yöneticileri İçin Use Case Stüdyosu modeli geliştirilmeden öncesinde yapılan gereksinim analizi süreci, boyutlandırma için Tişört modeli XS (Ekstra Küçük), S (Küçük), M(Orta), L (Büyük), XL (Ekstra Büyük) boyutlarını kullanmıştır [20]. Tişört Modeline göre XS büyüklükteki (1 hafta-1ay) proje için gereksinim analizi süreci 3 saat sürerken; Use Case Stüdyosu modeli geliştirildikten sonra yapılan gereksinim analizi ortalama 30 dakikada tamamlanmıştır. Tişört Modeline göre S büyüklükteki proje için Use Case Stüdyosu modeli geliştirilmeden öncesinde ortalama 10 saat süren gereksinim analiz süreci, model sonrasında ortalama 3 saatte tamamlanmıştır. Tişört Modeline göre M büyüklükteki (3-6 ay) proje için 24 saat sürerken; Use Case Stüdyosu modeli geliştirildikten sonra yapılan gereksinim analizi ortalama 6 saatte tamamlanmıştır. Gereksinim analiz sürecindeki 30 adet gerçek müşteri talebinin 20 adeti XS büyüklükte (1 hafta – 1 ay), 8 adeti S büyüklükte (1-3 ay) ve 2 adeti de M büyüklükte (3-6 ay).

Bir önceki bölümde detayları verilen model, şirket bünyesinde Otel ürün geliştirme bölümünde çalışan bir ürün yöneticisi ve bir araştırmacı tarafından test edilerek değerlendirilmiştir. Değerlendirme yapılırken kurumdaki 30 adet gerçek müşteri talebi baz alınarak modelin etkinliği incelenmiştir. Firmanın üçüncü bölümde açıklanan motivasyonu baz alınarak modelin değerlendirilmesi için beş temel kriter belirlenmiştir. Değerlendirme beşli bir Likert ölçeği [21] tasarlanarak yapılmıştır: **Çok Kötü (1)** – Beklentilerin çok altında, ciddi eksiklikler bulunmaktadır. **Kötü (2)** – Kısmen işe yarıyor ancak önemli eksiklikleri veya hataları bulunmaktadır. **Orta (3)** – Ne iyi ne kötü, bazı iyileştirmelere ihtiyaç duyulmaktadır. **İyi (4)** – Genel olarak başarılı, ancak küçük iyileştirmeler yapılabilir. **Mükemmel (5)**– Beklentilerin üzerinde, eksiksiz ve çok etkili çalışmaktadır.

Aşağıda bu kriterler açıklanmış ve modelin her bir kriter için başarısı değerlendirilmiştir:

i. Doğruluk ve Tutarlılık: Modelin kullanıcının girdilerini doğru yorumlayabilme becerisi, eksiksiz ve net gereksinimleri ne derece oluşturabildiği (*Gereksinim Tanımlayabilme*), aynı türde girdilere benzer çıktılar ne derece üretebildiği ve çelişkili veya değişken yanıtlar verip vermediği (*Tutarlılık*) ve son olarak üretilen gereksinimlerin sonradan düzeltilme veya değiştirilme ihtiyacının ne ölçüde olduğu (*Doğruluk Oranı*) bu kriter altında değerlendirilmiştir. Modelin gereksinimleri tanımlayabilme, tutarlılık ve doğruluk açısından güçlü olduğu tespit edilmiştir. Bu kriter için yapılan değerlendirme sonucunda 30 adet gerçek müşteri talebi proje büyüklükleri göz

önünde bulundurulduğunda 4.0'lık bir skor ortaya çıkmıştır.

ii. Kullanıcı Etkileşimi ve İş Birliği: Eksik veya belirsiz noktaları tespit edip, Ürün Yöneticisine ne ölçüde doğru sorular sorabildiği (*Soru Sorma Yetkinliği*), Ürün Yöneticisinin geri bildirimlerini ne ölçüde anlayıp, doğru ve uygulanabilir revizyonlar yapabildiği (*Yanıt Kalitesi*) ve son olarak Gereksinim analizini tamamlamak için Ürün Yöneticisi ile etkileşimlerin sayı ve süresi ne ölçüde tatmin edici olduğu (*Etkileşim Süresi*) bu kriter altında değerlendirilmiştir. Özelleştirilmiş model, doğru sorular sorma ve yanıtlara göre sağlıklı revizyonlar yapabilme konusunda tatmin edici seviyede performans göstermiştir. Ürün Yöneticisi ile etkileşimlerinin de oldukça etkili olduğu değerlendirilmiş ve bu kriter için konsolide olarak 30 adet gerçek müşteri talebi proje büyüklükleri göz önünde bulundurulduğunda 3.8'lik bir skor aldığı görülmüştür.

iii. Verimlilik ve Hız: Gelen müşteri talebinden nihai gereksinim dokümanını oluşturmaya olan sürenin ne ölçüde iyileştiği (*Gereksinim Belirleme Süresi*), geri bildirimleri işleyip yeni versiyonu oluşturma hızının ne ölçüde yeterli olduğu (*Düzenleme ve Revizyon Süresi*) ve son olarak Ürün Yöneticisinin yapması gereken manuel düzenleme ve ekleme sayısının ne kadar azaldığı (*Manuel İş Yükünün Azaltılması*) bu kriter altında değerlendirilmiştir. Ürün yöneticilerinin gereksinimlerin olgunlaştırılması ve kayıt altına alması sürecinde mevcut manuel akışa göre yaklaşık % 76 oranında bir verimlilik artışı olduğu görülmüştür. Bu kriterin değerlendirme skoru da tüm kriterler arasında en yüksek (4.3) şekilde gerçekleşmiştir. Bu sonuç, modelin verimlilik artışında çok önemli bir potansiyele sahip olduğunu göstermektedir.

iv. Çıktı Kalitesi ve Standartlara Uygunluk: Gereksinim dokümanlarının belirlenen formatlara (UCT, CRT vb.) ne ölçüde uygun şekilde üretildiği (*Şablon Uyumu*), çıktılarda tüm temel bileşenlerin ne derece eksiksiz şekilde yer aldığı (*Eksiksizlik*) ve son olarak çıktıların Ürün Yöneticisi ve diğer paydaşlar tarafından kolayca anlaşılabilmesi (*Anlaşılabilirlik*) bu kriter altında değerlendirilmiştir. Bu kriter kapsamında genel olarak içerik kalitesinin mevcut duruma göre arttığı ve modelin başarılı bulunduğu görülse de bazen formatlarda eksiklikler tespit edilmesi veya log gibi daha teknik konularda gelişim alanları da tespit edilmiştir. Sonraki versiyonlarda şirket bağlamındaki bilgilerin daha detaylı ve çeşitli olarak modele tanıtılması ile bu kriter yönünden daha güçlü olacağı beklenmektedir. Use Case Template (Kullanım Senaryosu Şablonu) toplamda 9 ana başlık ve 4 alt başlığı; toplamda 13 bileşeni konu ediniyor. Bu bileşenlerin 10'unda toplamda çıktı kitesi ve standartlara uygunluk tespit edilirken, 3'ünde iyileştirme maddesi bulunmaktadır. Bu kriter için yaklaşık 3.8'lik bir Likert skoruna ulaşıldığı görülmüştür.

v. Entegrasyon ve Adaptasyon: Kullanılan proje yönetimi veya gereksinim yönetimi araçlarıyla (JIRA [22], vb.) hangi ölçüde kolaylıkla entegre edilebildiği (*Sisteme Entegrasyon*), kullanıcı geri bildirimleriyle gelişip, sonraki gereksinim analizlerinde daha iyi öneriler sunma derecesi (*Öğrenme Yeteneği*) ve son olarak farklı ürün ve iş alanlarına göre adapte edilip edilemediği (*Özelleştirme Yeteneği*) bu kriter altında değerlendirilmiştir. Ajan Temelli Yapay Zeka (Agentic Artificial Intelligence) araçları ile entegrasyon imkanlarının bulunması, özelleştirilmiş modelin yeni materyal ile alana özgü sürekli eğitilmesine açık bir önyüz bulunması ve bağlam bilgileri ile farklı iş alanlarına özelleştirilebilmesi nedeniyle model bu kriter için yüksek bir skor (4.2) almıştır.

Tüm kriterlerin ağırlıklarının eşit olduğu varsayıp toplam skor hesaplandığında yaklaşık 4'lük Likert skoru ortaya çıkmıştır. Bu değer, henüz prototip aşamasında olan bir çözüm için oldukça umut vadeden başarılı bir sonuç olarak değerlendirilmiş ve şirket içinde kullanıma başlanmıştır.

Sonuç

Yazılım geliştirme süreçlerinin en kritik safhalarından gereksinim analizi aşamasının büyük dil modelleri kullanılarak daha verimli ve kaliteli hale getirilmesini konu olan çalışmamızda özelleştirilmiş bir GPT modeli tasarlanarak çalışmanın yürütüldüğü kurum içerisindeki gerçek projeler kullanılarak test edilmiş ve sonuçları değerlendirilmiştir.

Belirlenmiş olan beş temel kritere göre yapılan değerlendirmelerde modelin en güçlü olduğu alan verimlilik ve hız kriteri olarak belirlenmiştir. Yapılan ön değerlendirme çalışmasında ürün yöneticilerinin gereksinim analizi işlerini %76'ya yaklaşan oranlarda daha hızlı ve verimli yaptıkları tespit edilmiştir. Bununla birlikte yine tatmin edici sonuçlar alınsa da çıktı kalitesi ve standartlara uygunluk kriterinde özellikle log kayıtları ve olayların belirlenmesinde; mevcut veya öngörülen kullanım frekansının tahminlenmesinde ve uygulama mimarisine etkisinin netleştirilmesinde iyileşme alanları tespit edilmiştir. Yine yapılan ilk değerlendirmede konsolide Likert skor hesaplandığında 5 üzerinden 4'lük bir değer ortaya çıkmıştır. Toplam olarak değerlendirildiğinde firmanın beklentileri ile uyumlu olarak ihtiyaçları etkin bir şekilde karşılayabileceği ve yapılacak iyileştirmelerle önemli bir potansiyelinin de olduğu tespit edilmiştir.

Gelecek çalışmaları olarak loglama standartları, kapsam büyüklüğü ölçümü, uygulamalar mimarisinin tanıtılması, mevcut veya öngörülen kullanım frekansının aktarılması gibi alanlarında iyileşme sağlanması için daha detaylı araştırmalar yapılacak ve bağlam bilgisi ile modelin daha geniş ölçekte kullanılması sağlanıp daha kapsamlı değerlendirmeler ve ileri araştırmalar yapılacaktır.

Teşekkürler

Bu çalışmanın sahada uygulanması sürecinde sağladıkları değerli destek için **Wingie Enuygun Group**'a ve özellikle **Hakan Kanar, Behçet Mutlu, Yankı Öksüz ve Nihan Çolak Erol** 'a içten teşekkürlerimizi sunarız. Verdikleri katkılar, araştırmamızın gerçek ortam koşullarında test edilip değerlendirilmesine önemli ölçüde yardımcı olmuştur.

Kaynaklar

- [1] Wiegers KE, Beatty J. *Software requirements*. 3rd ed. Pearson Education; 2013.
- [2] Curcio K, Navarro T, Malucelli A, Reinehr S. Requirements engineering: A systematic mapping study in agile software development. *J Syst Softw*. 2018;139:32–50.
- [3] Garousi V, Coşkunçay A, Betin-Can A, Demirörs O. A survey of software engineering practices in Turkey. *J Syst Softw*. 2015;108:148–77.

- [4] Beck K, Beedle M, Van Bennekum A, Cockburn A, Cunningham W, Fowler M, et al. The agile manifesto. 2001 Feb. Available from: <https://www.agilealliance.org/wp-content/uploads/2019/09/agile-manifesto-download-2019.pdf>
- [5] Sandahl K, Regnell B, Borg M. The magazine at 40: Viewing requirements engineering through a ruby lens. *IEEE Softw.* 2024;41(6):17–22.
- [6] Arora C, Grundy J, Abdelrazek M. Advancing requirements engineering through generative AI: Assessing the role of LLMs. In: *Generative AI for Effective Software Development*. Cham: Springer Nature Switzerland; 2024. p. 129–48.
- [7] OpenAI. Introducing GPTs [Internet]. Available from: <https://openai.com/index/introducing-gpts/>
- [8] Wei B. Requirements are all you need: From requirements to code with LLMs. *arXiv preprint arXiv:2406.10101*. 2024.
- [9] Yeow JSN, Rana ME, Majid NAA. An automated model of software requirement engineering using GPT-3.5. *2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)*; 2024 Jan. p. 1746–55. IEEE.
- [10] Oswal JU, Kanakia HT, Suktel D. Transforming software requirements into user stories with GPT-3.5: An AI-powered approach. *2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*; 2024 Jan. p. 913–20. IEEE.
- [11] Marques N, Silva RR, Bernardino J. Using ChatGPT in software requirements engineering: A comprehensive review. *Future Internet*. 2024;16(6):180.
- [12] Abrahão S, Staron M, Gay G, Penzenstadler B, Honnenahalli C. Emerging trends in requirements engineering and testing. *IEEE Softw.* 2024;41(6):126–9.
- [13] Vogelsang A. From specifications to prompts: On the future of generative large language models in requirements engineering. *IEEE Softw.* 2024;41(5):9–13.
- [14] Wingie Enuygun Group. Wingie Enuygun Group. LinkedIn. Erişim tarihi: 4 Ocak 2025. Erişim adresi: <https://tr.linkedin.com/company/wingieenuygungroup>.
- [15] Parthasarathy VB, Zafar A, Khan A, Shahid A. The ultimate guide to fine-tuning LLMs from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities. *arXiv preprint arXiv:2408.13296*; 2024.
- [16] Wikipedia. ChatGPT. Wikipedia, The Free Encyclopedia. Erişim tarihi: 27 Şubat 2025. Erişim adresi: <https://en.wikipedia.org/wiki/ChatGPT>.
- [17] Jacobson I, Cockburn A. Use cases are essential: Use cases provide a proven method to capture and explain the requirements of a system in a concise and easily understood format. *Queue*. 2023;21(5):66–86.
- [18] Cockburn A, Cockburn L. *Writing effective use cases*. Pearson Education India; 2008.
- [19] Bagriyanik S, Karahoca A. Automated COSMIC function point measurement using a requirements engineering ontology. *Inf Softw Technol*. 2016;72:189–203.
- [20] Mallidi RK, Sharma M. Study on agile story point estimation techniques and challenges. *Int. J. Comput. Appl.* 2021 Jan 13;174(13):9-14.
- [21] Joshi A, Kale S, Chandel S, Pal DK. Likert scale: Explored and explained. *Br J Appl Sci Technol*. 2015;7(4):396.
- [22] Atlassian. Jira [İnternet]. [erişim tarihi 26 Aralık 2024]. Erişim adresi: <https://www.atlassian.com/software/jira>