

**TÜRK HAVA KURUMU ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**TÜRKÇE YAZILIM GEREKSİNİMLERİ İÇİN DOĞAL DİL İŞLEME  
TABANLI BİÇİMSEL GÖZDEN GEÇİRME ARACI**



**YÜKSEK LİSANS TEZİ**

**Elif YALÇINKAYA**

**Elektrik ve Bilgisayar Anabilim Dalı**

**Elektrik ve Bilgisayar Mühendisliği Programı**

**EYLÜL 2018**

**TÜRK HAVA KURUMU ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**TÜRKÇE YAZILIM GEREKSİNİMLERİ İÇİN DOĞAL DİL İŞLEME  
TABANLI BİÇİMSEL GÖZDEN GEÇİRME ARACI**

**YÜKSEK LİSANS TEZİ**

**Elif YALÇINKAYA**

**1303617007**

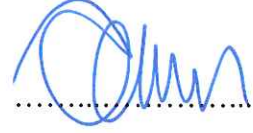
**Elektrik ve Bilgisayar Anabilim Dalı**

**Elektrik ve Bilgisayar Mühendisliği Programı**

**Tez Danışmanı: Doç. Dr. Tansel DÖKEROĞLU**

Türk Hava Kurumu Üniversitesi Fen Bilimleri Enstitüsü'nün 1303617007 numaralı Yüksek Lisans öğrencisi, Elif YALÇINKAYA ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "TÜRKÇE YAZILIM GEREKSİNİMLERİ İÇİN DOĞAL DİL İŞLEME TABANLI BİÇİMSEL GÖZDEN GEÇİRME ARACI" başlıklı tezini, aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

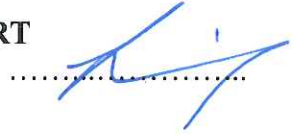
**Tez Danışmanı** : **Doç. Dr. Tansel DÖKEROĞLU**  
TED Üniversitesi



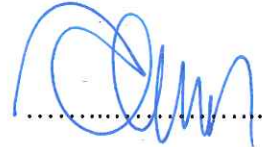
**Jüri Üyeleri** : **Prof. Dr. Ahmet COŞAR**  
Türk Hava Kurumu Üniversitesi



: **Dr. Öğr. Üyesi Selen PEHLİVAN TORT**  
TED Üniversitesi



: **Doç. Dr. Tansel DÖKEROĞLU**  
TED Üniversitesi



**Tez Savunma Tarihi:** 07 Eylül 2018

**TÜRK HAVA KURUMU ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜ'NE**

Yüksek Lisans Tezi olarak sunduğum, “TÜRKÇE YAZILIM GEREKSİNİMLERİ İÇİN DOĞAL DİL İŞLEME TABANLI BİÇİMSEL GÖZDEN GEÇİRME ARACI” adlı çalışmamın, tarafımdan akademik etik ve kurallara aykırı düşecek bir yardıma başvurmaksızın yazıldığını ve yararlandığım kaynakların kaynakçada gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve bunu onurumla doğrularım.

07.09.2018

Elif YALÇINKAYA



## TEŐEKKÜR

Tez alıŐmalarım sűresince, beni yűnlendiren ve desteęini hi esirgemeyen tez danıŐmanım ve deęerli hocam Do. Dr. Tansel DűKEROęLU'na teŐekkűr ve saygılarımı sunarım.

alıŐmalarım sırasında, deneyler kapsamında ihtiyacım olduęunda hızlı dűnűŐleri iin İTŪ doęal dil iŐleme araŐtırma grubuna ve beni yalnız bırakmayıp sabırla destek olan sevgili eŐim ve kızıma teŐekkűr ederim.

Eylűl, 2018

Elif YALINKAYA

## İÇİNDEKİLER

TEŞEKKÜR.....	iv
İÇİNDEKİLER .....	v
TABLOLAR LİSTESİ.....	vii
ŞEKİLLER LİSTESİ .....	ix
KISALTMALAR .....	x
ÖZET.....	xi
ABSTRACT.....	xii
<b>BİRİNCİ BÖLÜM</b> .....	1
<b>1. GİRİŞ</b> .....	1
<b>İKİNCİ BÖLÜM</b> .....	14
<b>2. LİTERATÜR ÖZETİ</b> .....	14
2.1 Standartlar .....	14
2.1.1 ISO/IEC FDIS 9126-1 .....	14
2.1.2 IEEE Std 830-1993 .....	15
2.1.3 RTCA DO-178 .....	16
2.1.4 NASA-STD Yazılım Mühendisliği Standartları .....	16
2.2 Otomatik Gereksinim Gözden Geçirme .....	18
2.3 Doğal Dil İşleme (NLP) .....	20
2.4 İngilizce Yazılım Gereksinimleri için Otomatik Gözden Geçirme Yaklaşımları .....	21
2.4.1 NASA-ARM .....	21
2.4.2 Quars .....	22
2.4.3 QVscribe .....	23
2.4.4 RQA Client .....	23
2.4.5 Ambiguity Detector .....	24
2.4.6 Tiger Pro .....	25
2.4.7 Diğerleri .....	26
2.5 Türkçe Gereksinimler için Otomatik Gözden Geçirme Yaklaşımları.....	27
2.5.1 İngilizce analiz yaklaşımlarının Türkçe gereksinimlere uygulanması .....	27
2.5.2 Dilbilimsel analizler ile Türkçe gereksinimlerin kalitesinin arttırılması.....	29
<b>ÜÇÜNCÜ BÖLÜM</b> .....	32
<b>3. ÖNERİLEN YÖNTEM</b> .....	32
3.1 Veri Seti.....	33
3.1.1 Gereksinimler.....	33
3.1.2 Kullanılan Gereksinim Dokümanları.....	34
3.1.3 Veri seti gereksinimlerin dosya formatı ve yazım standardı .....	35
3.2 Otomatik Türkçe Gereksinim Gözden Geçirme Kontrol Listesi .....	36
3.3 Otomatik Gereksinim Gözden Geçirme .....	41
3.3.1 Tez kapsamında Türkçe Doğal Dil İşleme .....	42

3.3.1.1 Türkçe Dil Yapısının Etkisi .....	42
3.3.1.2 ITÜ Türkçe Doğal Dil İşleme Yazılım Zinciri .....	42
3.3.2 Gereksinim gözden geçirme algoritmaları .....	45
3.3.3 Otomatik Gereksinim Gözden Geçirme Yazılım Tasarımı (OG3) .....	55
<b>DÖRDÜNCÜ BÖLÜM</b> .....	59
<b>4. DENEYSEL ÇALIŞMALAR</b> .....	59
4.1 Test Ortamı.....	60
4.2 Tiger Pro ile İngilizce gereksinimlerin otomatik gözden geçirilmesi .....	60
4.2.1 İngilizce Havayolu Rezervasyon Sistemi gereksinimlerinin gözden geçirme sonuçları .....	60
4.2.2 İngilizce E-Market gereksinimlerinin gözden geçirme sonuçları.....	62
4.2.3 İngilizce Uçuş Kontrol Sistemi gereksinimlerinin gözden geçirme sonuçları .....	63
4.2.4 İngilizce Eğitimsel Oyun Projesi gereksinimlerinin gözden geçirme sonuçları .....	63
4.2.5 İngilizce Proje Yönetim Sistemi gereksinimlerinin gözden geçirme sonuçları .....	64
4.2.6 Tiger Pro ile gözden geçirme özeti .....	65
4.3 Tez yazılımı OG3 ile Otomatik Gözden Geçirme.....	65
4.3.1 Türkçe Havayolu Rezervasyon Sistemi gereksinimlerinin gözden geçirme sonuçları .....	66
4.3.2 Türkçe E-Market gereksinimlerinin gözden geçirme sonuçları .....	67
4.3.3 Türkçe Uçuş Kontrol Sistemi gereksinimlerinin gözden geçirme sonuçları .....	69
4.3.4 Türkçe Eğitimsel Oyun gereksinimlerinin gözden geçirme sonuçları.....	71
4.3.5 Türkçe Proje Yönetim Sistemi gereksinimlerinin gözden geçirme sonuçları .....	72
4.3.6 Türkçe Dokunmatik Gazete gereksinimlerinin K2 kontrol maddesine özel gözden geçirmesi.....	74
4.3.7 OG3 ile gözden geçirme özeti .....	75
4.4 Karşılaştırma .....	76
<b>BEŞİNCİ BÖLÜM</b> .....	81
<b>5. SONUÇ</b> .....	81
<b>KAYNAKLAR</b> .....	84
<b>EKLER</b> .....	87
Ek A- Gereksinim Yönetim Araçları Listesi .....	88
Ek B- Gereksinim Yönetim Araç Karşılaştırma Kriterleri.....	91
Ek C- DDİ Zinciri Sözlük Detayları.....	99
Ek D- OG3 Gözden Geçirme Yazılımı Sözcüksel Analiz Veri tabanı.....	102
Ek E- Gereksinim gözden geçirme kayıtları (CD) .....	103
<b>ÖZGEÇMİŞ</b> .....	104

## TABLolar LİSTESİ

<b>Tablo 1.1</b>	: Riskler ve Karşılama teknikleri .....	3
<b>Tablo 1.2</b>	: Risklerin ölçüm kriterleri .....	3
<b>Tablo 1.3</b>	: Hataların giderilmesine dair göreceli maliyet oranları .....	5
<b>Tablo 1.4</b>	: En başarılı yazılımlar .....	10
<b>Tablo 2.1</b>	: ISO/IEC 9126'ya göre yazılım kalite nitelikleri .....	15
<b>Tablo 2.2</b>	: IEEE Std 830-1993'da tanımlı iyi gereksinim nitelikleri .....	15
<b>Tablo 2.3</b>	: DO-178B yazılım gereksinim gözden geçirme ve analiz kriterleri .....	16
<b>Tablo 2.4</b>	: NASA Yazılım Gereksinim gözden geçirme kontrol listesi .....	17
<b>Tablo 2.5</b>	: Quars Kalite göstergeleri .....	22
<b>Tablo 2.6</b>	: Gereksinim dokümanlarında bulunan hata oranları .....	24
<b>Tablo 2.7</b>	: İngilizce gereksinim analiz yazılımları .....	26
<b>Tablo 2.8</b>	: Hata tipleri ve analiz yöntemleri .....	28
<b>Tablo 3.1</b>	: Otomatik Gözden Geçirme Kontrol Listesi .....	37
<b>Tablo 3.2</b>	: Sözdizimsel bağlılık etiketleri .....	44
<b>Tablo 3.3</b>	: Analiz edilen örnek gereksinim .....	46
<b>Tablo 3.4</b>	: DDİ Biçim bilimsel analiz sorgusu dönüşü .....	47
<b>Tablo 3.5</b>	: DDİ formal dil zincir sorgusu dönüşü .....	50
<b>Tablo 4.1</b>	: Test Ortam Konfigürasyonu .....	60
<b>Tablo 4.2</b>	: Tiger Pro ile Havayolu Rezervasyon Sistemi gereksinim gözden geçirme sonuçları .....	61
<b>Tablo 4.3</b>	: Tiger Pro ile E-Market Sistemi gereksinim gözden geçirme sonuçları .....	62
<b>Tablo 4.4</b>	: Tiger Pro ile Uçuş Kontrol Sistemi gereksinim gözden geçirme sonuçları .....	63
<b>Tablo 4.5</b>	: Tiger Pro ile Eğitimsel Oyun Projesi gereksinim gözden geçirme sonuçları .....	63
<b>Tablo 4.6</b>	: Tiger Pro ile Proje Yönetim Sistemi gereksinim gözden geçirme sonuçları .....	64
<b>Tablo 4.7</b>	: Tiger Pro ile gözden geçirme istatistikleri .....	65
<b>Tablo 4.8</b>	: OG3 Yazılımı ile Havayolu Rezervasyon Sistemi otomatik gözden geçirme sonuçları (Bulgu sayısı 0 olanlar çıkartılmıştır.) .....	66
<b>Tablo 4.9</b>	: OG3 Yazılımı ile E-Market otomatik gözden geçirme sonuçları (Bulgu sayısı 0 olanlar çıkartılmıştır.) .....	68
<b>Tablo 4.10</b>	: OG3 Yazılımı ile Uçuş Kontrol Sistemi otomatik gözden geçirme sonuçları (Bulgu sayısı 0 olanlar çıkartılmıştır.) .....	69
<b>Tablo 4.11</b>	: OG3 Yazılımı ile Eğitimsel Oyun otomatik gözden geçirme sonuçları (Bulgu sayısı 0 olanlar çıkartılmıştır.) .....	71
<b>Tablo 4.12</b>	: OG3 Yazılımı ile Proje Yönetim Sistemi otomatik gözden geçirme sonuçları (Bulgu sayısı 0 olanlar çıkartılmıştır.) .....	73



<b>Tablo 4.13 :</b> OG3 yazılımı ile Dokunmatik Gazete proje gereksinimleri için K2 ile gözden geçirme sonuçları .....	75
<b>Tablo 4.14 :</b> OG3 yazılımı ile gözden geçirme istatistikleri .....	75
<b>Tablo 4.15 :</b> Gözden geçirme istatistikleri .....	76
<b>Tablo 4.16 :</b> Tiger Pro yazılımı doğru bulgu istatistikleri .....	77
<b>Tablo 4.17 :</b> OG3 yazılımı doğru bulgu istatistikleri .....	78
<b>Tablo 4.18 :</b> Yazılım Karşılaştırma Tablosu .....	79



## ŞEKİLLER LİSTESİ

Şekil 1.1	: Geliştirme yaşam döngüsünde oluşan hataların dağılımı .....	4
Şekil 1.2	: Sistem hatalarının giderilmesinde maliyet oranları .....	5
Şekil 1.3	: Gereksinim kalitesinin proje bütçe ve zaman kalemlerine etkisi .....	6
Şekil 2.1	: Firmaların proje gereksinimlerindeki terminoloji kullanımı .....	18
Şekil 2.2	: Projelerde önem verilen ve geliştirilmek istenen aktiviteler .....	19
Şekil 2.3	: Tiger Pro yazılımı kullanıcı ara yüzü .....	25
Şekil 3.1	: Otomatik gereksinim gözden geçirme yaklaşımı.....	33
Şekil 3.2	: XML formatlı Türkçe gereksinim seti .....	36
Şekil 3.3	: Formal dil zincir algoritma safhaları .....	43
Şekil 3.4	: Örnek bir gereksinim için bağlılık ağacı.....	45
Şekil 3.5	: OG3 dağılım şeması.....	45
Şekil 3.6	: K1 algoritması ile dil kararı .....	48
Şekil 3.7	: K2 ile metin normalizasyonu.....	49
Şekil 3.8	: Kontrol algoritmalarında yüklem analizi.....	50
Şekil 3.9	: Tek bağlı gereksinim örneği .....	51
Şekil 3.10	: K7 algoritmasının tespit edeceği birden fazla bağ içeren gereksinim.....	52
Şekil 3.11	: Fiilimsi koşul analiz - Tip 1 .....	52
Şekil 3.12	: Fiilimsi koşul analizi -Tip 2.....	53
Şekil 3.13	: Fiil koşul analizi.....	53
Şekil 3.14	: Bağlaç koşul analizi .....	54
Şekil 3.15	: K7 Algoritma Akış diyagramı .....	54
Şekil 3.16	: Kullanıcı arayüzü ile beklenen numara ve izlenebilirlik formatı .....	55
Şekil 3.17	: OG3 İle Otomatik Gereksinim Gözden Geçirme Süreci .....	56
Şekil 3.18	: OG3 yazılımı Gereksinimler ekran görüntüsü.....	56
Şekil 3.19	: OG3 yazılımı Kontrol Listesi ekran görüntüsü.....	57
Şekil 3.20	: OG3 yazılımı Analiz & Rapor ekran görüntüsü .....	58
Şekil 4.1	: OG3 ile Havayolu Rezervasyon Sistemi gereksinimleri için gereksinim tipi/bulgu grafiği .....	67
Şekil 4.2	: OG3 ile Havayolu Rezervasyon Sistemi gereksinimleri için hata tipi/bulgu grafiği .....	67
Şekil 4.3	: OG3 ile E-Market gereksinimleri için gereksinim tipi/bulgu grafiği.....	68
Şekil 4.4	: OG3 ile E-Market gereksinimleri için hata tipi/bulgu grafiği .....	69
Şekil 4.5	: OG3 ile Uçuş Kontrol Sistemi için gereksinim tipi/bulgu grafiği .....	70
Şekil 4.6	: OG3 ile Uçuş Kontrol Sistemi gereksinimleri için hata tipi/bulgu grafiği .....	70
Şekil 4.7	: OG3 ile Eğitimsel Oyun gereksinimleri için gereksinim tipi/bulgu grafiği .....	72
Şekil 4.8	: OG3 ile Eğitimsel Oyun gereksinimleri için hata tipi/bulgu grafiği.....	72
Şekil 4.9	: OG3 ile Proje Yönetim Sistemi gereksinimleri için gereksinim tipi/bulgu grafiği .....	73
Şekil 4.10	: OG3 ile Proje Yönetim Sistemi gereksinimleri için hata tipi/bulgu grafiği .....	74

## KISALTMALAR

<b>ARM</b>	: Automated Requirement Measurement
<b>CRF</b>	: Conditional Random Fields
<b>DDİ</b>	: Doğal dil işleme
<b>EUROCAE</b>	: European Organization for Civil Aviation Equipment
<b>FAA</b>	: Federal Havacılık İdaresi
<b>HRS</b>	: Havayolu Rezervasyon Sistemi
<b>HTTP</b>	: Hypertext Transfer Protocol
<b>ISO</b>	: Uluslararası Standart Örgütü
<b>İTÜ</b>	: İstanbul Teknik Üniversitesi
<b>IEEE</b>	: Institute of Electrical and Electronics Engineers
<b>NASA</b>	: Ulusal Havacılık ve Uzay Dairesi
<b>NLP</b>	: Natural Language Processing
<b>OGGKL</b>	: Otomatik Gözden Geçirme Kontrol Listesi
<b>OG3</b>	: Otomatik Gereksinim Gözden Geçirme
<b>PYS</b>	: Proje Yönetim Sistemi
<b>QUARS</b>	: Quality Analyzer for Requirements Specifications
<b>RQA</b>	: Requirement Quality Analyzer
<b>RTCA</b>	: Radio Technical Commission for Aeronautics
<b>TIGER PRO</b>	: Tool to Ingest and Elucidate Requirements Professional
<b>UKS</b>	: Uçuş Kontrol Sistemi

## ÖZET

### TÜRKÇE YAZILIM GEREKSİNİMLERİ İÇİN DOĞAL DİL İŞLEME TABANLI BİÇİMSEL GÖZDEN GEÇİRME ARACI

YALÇINKAYA, Elif

Yüksek Lisans Tezi, Elektrik ve Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Tansel DÖKEROĞLU

Eylül 2018, 104 sayfa

Yazılım geliştirme yaşam döngüsü içerisinde oluşabilecek hataların erken safhada tespit edilip ortadan kaldırılması gereksinim yönetim süreci içerisinde gözden geçirme aktiviteleri ile gerçekleştirilmektedir. Literatürde belirli kurum ve standartlar tarafından yönlendirilen bu sürecin Türkçe gereksinimler üzerinde objektif, hızlı ve hatasız bir şekilde uygulanabilmesi için otomasyon ihtiyacı vardır. Türkçe yazılım gereksinimlerinin otomatik olarak gözden geçirilerek başarıyla hatalarından arındırılması kullanılan gözden geçirme kontrol listesinin kapsamı, doğruluğu ve ortaya konulan dil işleme algoritmalarının başarısıyla doğru orantılıdır. Bu amaçla, Türkçe yazılım projeleri için yeni bir biçimsel Otomatik Gereksinim Gözden Geçirme (OG3) aracı geliştirilmiştir. OG3, literatürdeki gereksinimler ile ilgili standart ve çalışmalarda ortaya konulan gözden geçirme kontrol listelerinin ölçme ve değerlendirme kriterleri esas alınarak yeniden oluşturulmuş kapsamlı bir kontrol listesi kullanmaktadır (OGGKL). Bu kontrol listesine karşılık geliştirilen yazılımın başarısını ölçme ve değerlendirme amacıyla farklı sektör ve projelerden gereksinim örnekleri incelenmiş, söz konusu gereksinimlerin İngilizce versiyonları literatürde öne çıkan bir gözden geçirme yazılımıyla analiz edilirken, diğer taraftan Türkçe'ye çevrilen aynı gereksinimler OG3 yazılım ile gözden geçirilmiş ve sonuçlar karşılaştırılmıştır.

**Anahtar Kelimeler:** Gereksinim yönetimi, otomatik gereksinim gözden geçirme, doğal dilde gereksinim analizi, hata tespiti

## ABSTRACT

### NATURAL LANGUAGE PROCESSING BASED FORMAL AUTOMATED REVIEW TOOL FOR TURKISH SOFTWARE REQUIREMENTS

YALÇINKAYA, Elif

Master, Department of Electrical and Computer Engineering

Thesis Supervisor: Doç. Dr. Tansel DÖKEROĞLU

September 2018, 104 pages

Early detection and removal of defects that may occur in software development lifecycle is achieved through requirements review activities within the requirements management process. There is an automation need for applying this process driven by specific institutions and standards, to requirements in Turkish language on an objective, fast and error-free basis. The success of automatic detection and removal of defects in Turkish software requirements is directly proportional to the success of natural language processing algorithms and the content and accuracy of requirements review checklist that it is built on. For this purpose, a new formal Automatic Requirement Review (OG3) tool has been developed for Turkish software projects. OG3, uses a comprehensive checklist (OGGKL) that is rebuilt and based on the measurement and evaluation criteria of the requirements review checklists from standards and related studies in the literature. In order to measure and evaluate the success of the developed software based on this checklist, requirements were gathered from different sectors and projects and while analyzing English documents on a review tool which is reported in the literature, on the other side the same requirements that were translated to Turkish are reviewed by OG3 software and the results are compared.

**Keywords:** requirement management, automated requirement review, requirement analysis on natural language, defect detection

## BİRİNCİ BÖLÜM

### GİRİŞ

Yazılım veya sistem mühendisliği temelli projelerde, tasarlanan ürün her ne olursa olsun geliştirecek olan kurum ile talep eden arasında mutabakat sağlanan, ürünü ve projeyi tanımlayıcı bir dokümantasyon ihtiyacı vardır. Bu dokümanlar proje ve ürün özelliklerine, müşteri veya geliştirici kurum istek ve süreçlerine, denetleyici kurumun yönlendirmesi veya sektör standartlarına bağlı olarak değişiklik gösterebilir. Projenin büyüklük ve yapısına bağlı olarak kimi zaman basit bir analiz dokümanı ile taraflar anlaşılabilirken, kimi zamanda en üst seviyede kontrat ile başlayan ve aşağıya doğru farklı seviyelerde detaylandırılan analiz ve gereksinim dokümanları yazılmakta ve proje yaşam döngüsü boyunca bu dokümanlar arasında izlenebilirlik kurularak yönetilmektedir. Bu dokümanların temel birimi, gereksinimi basitçe tanımlamak gerekirse, müşterinin geliştirilmesini istediği ürünün gerçekleştirilmesi gereken fonksiyonları tanımlayan en küçük ister, ihtiyaç ve doküman birimi olarak ifade edilebilir. Sistemin veya yazılımın belirli bir anda gerçekleştireceği belirli bir işi ifade eden bu doküman birimi, literatürde gerçekleştirilmek istenen fonksiyonun resmi ifadelerle dönüşmüş hali, ya da gerçekleştirilecek olan bütünün ayrılmaz parçalarını ifade eden nitelikler olarak da tanımlanmıştır [1]. Gereksinim yönetimi ise, tüm paydaşlar tarafından gereksinimlerin görüşülerek üzerinde anlaşılması, önceliklendirilmesi, analiz edilmesi, izlenebilirliklerin çekilmesi, değişiminin ve güncellenmesinin sağlanması sürecidir [5]. Yazılım ve sistem mühendisliği özelinde ise, gereksinimlerin tanımlanması, atanması, ara yüzlerin belirlenmesi, doğrulama, güncellenme ve statü takip aktiviteleri ile birlikte sistem seviyesinden alt seviye modüllere akış süreçlerinin tamamı olarak ifade edilmiştir [15].

Gereksinim yönetim sürecinin başarısı taraflar arasındaki kurulan iletişimin başarısı ile doğru orantılıdır. Projeyi geliştiren ekip ile son ürün kullanıcıları

(müşteriler) arasında iletişim gereksinimler üzerinden sağlanır. Belirli bir standarta uygun dokümanlar ile bu süreç yürütülse de, çoğu zaman taraflar arasında yanlış anlaşılmalara, hem fikir olamama gibi sorunlar ile sıkça karşılaşmaktadır. Geliştiren ekip ile kullanıcılar arasında iletişim sorunları, aynı gereksinim metninden aynı anlamın çıkarılmaması, karşılıklı beklentilerin karşılanamaması gibi sorunlar ile mücadele edilmektedir. Geliştirme ekipleri üzerinde yapılan ankete göre, katılanların %46'sı müşteri ihtiyaçlarının karşılanmadığı argümanı ile projelerin sürekli gecikmesi veya hiç tamamlanamamasından şikâyet etmektedirler [5]. Geliştirme ekiplerinin bu yaklaşımından, müşterilerin isteklerinin hiç bitmediği, sık sık değiştiği veya çok zor ikna oldukları sonucu çıkarılabilir. Müşterinin ne istediği kolay olduğu düşünülen fakat genelde cevaplanması çok zor bir sorudur [5]. Projelerde oluşan hata ve arızaların %50 ile %60 civarı yazılım yönetimindeki başarısızlığa bağlanmıştır [1]. Bir diğer çalışmada ise son ürün hatalarının %50'si, projede oluşan beklenmeyen ekstra efor ve işlerin ise % 80'i gereksinimlere dayandırılmıştır [9]. Bu hatalar hangi sektörde olursa olsun projelere ekstra maliyet ve gecikme gibi negatif etkilerinin yanında, emniyet kritik sistemlerde ise telafisi olmayan can ve mal kaybı ile sonuçlanabilecek kazalara sebep olabilmektedir. Bu sebeple düzgün işletilen ve denetlenen bir gereksinim yönetim süreci proje yönetimi için öncelikli bir ihtiyaçtır.

Gereksinim yönetimi süreci ilk olarak bilim insanlarının kendi ihtiyaçları ve araştırmaları için geliştirdikleri projelerde görülmektedir. Son kullanıcı ve müşterilerin yine kendileri oldukları düşünüldüğünde, bugün tanımladığımız süreçlerden çok daha sade ve basit yönetimlerin varlığından bahsedebiliriz. Daha sonra iş amaçlı geliştirilen ve müşterilerin bulunduğu projeler ile birlikte gereksinim yönetimi önem kazanmış ve zamanla şekillenmeye başlamıştır. Projelerde yaşanan sorun ve hatalardan proje ve gereksinim yönetiminin eksiklikleri ortaya çıkarılmış ve ilgili riskler belirlenmiştir. Gereksinim yönetim süreci de bu risklerin yönetimi ve etkilerini azaltma amacı ile yeni yöntem ve yaklaşımlar ile gelişerek günümüze kadar ulaşmıştır. Literatürde bu riskleri ve onlara karşılık geliştirilen çalışma ve yöntemleri net bir şekilde ortaya koymak amacı ile bir analiz çalışması yapılmıştır [6]. Bu çalışmada gereksinim yönetimi adına gereksinim karmaşıklığı, uygunluğu ve geçerliliği temel riskler olarak ortaya çıkmış ve bu riskleri giderebilecek teknikler gereksinim belirleme, deneyleme ve ilk ortaya çıkarılma süreçleri işaret edilmiştir.

**Tablo 1.1:** Riskler ve Karşılama teknikleri [6]

<b>Riskler</b>	<b>Teknikler</b>
Gereksinim Karmaşıklığı	Gereksinim belirleme
Gereksinim Uygunluğu	Gereksinim deneyleme
Gereksinim Geçerliliği	Gereksinim keşfi (ortaya çıkarılması)

Gereksinim risklerinin pratikteki yansımaları ancak gereksinim mühendislik süreci ile yazılım geliştirme süreçleri üzerinden analiz edilebilir. Bu amaçla, Tablo 1.1’de literatürde ilk ve kapsam açısından en geniş çalışmalardan olan, Barki’nin (1993) yazılım geliştirme süreci için belirlediği riskler ile Davis’in (1982) gereksinim mühendisliği üzerine sunduğu riskler Tablo 1.2’de tanımlanan genel gereksinim riskleri için ölçüt olacak şekilde eşleştirilmiştir [6].

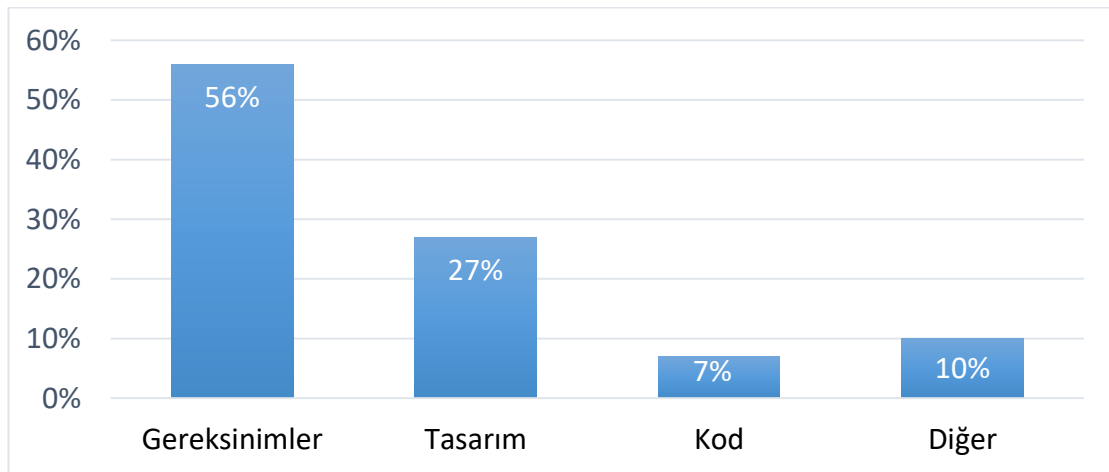
**Tablo 1.2:** Risklerin ölçüm kriterleri [6]

<b>Riskler</b>	<b>Ölçütler</b>	
	Yazılım Geliştirme [2]	Gereksinim Mühendisliği [10]
Gereksinim Karmaşıklığı	<ul style="list-style-type: none"><li>• Teknik zorluk</li><li>• Benzer proje büyüklüğü</li><li>• Var olan sistemlere olan bağlantı\ilişki sayısı</li><li>• Gelecek sistemlere olan bağlantı\ilişki sayısı</li><li>• Donanım sağlayıcı sayısı</li><li>• Yazılım sağlayıcı sayısı</li><li>• Yeni donanım ihtiyacı</li><li>• Yeni yazılım ihtiyacı</li></ul>	<ul style="list-style-type: none"><li>• Karmaşık sistem</li><li>• Sistemin iyi anlaşılmuş bir modelinin oluşmaması</li><li>• Karar veya aktiviteler için yapısal eksiklik</li></ul>
Gereksinim Uygunluğu	<ul style="list-style-type: none"><li>• Görev zorluğu</li><li>• Değişikliklerin boyutu</li><li>• Ekipte geliştirme tecrübe eksikliği</li><li>• Uygulama özelinde tecrübesizlik</li><li>• Genel iş tecrübesi eksikliği</li><li>• Kaynak yetersizliği</li><li>• Potansiyel kaybın büyüklüğü</li><li>• Anlaşmazlıkların yoğunluğu</li></ul>	<ul style="list-style-type: none"><li>• Bilgi sistemini kullanım istikrar eksikliği</li><li>• Sistemdeki değişiklikler</li><li>• Sistemdeki yapısal ve operasyonel istikrar eksikliği</li><li>• Bilgi kullanımında değişiklikler</li><li>• Kullanıcıların sisteme özel tecrübesizlikleri</li></ul>
Gereksinim Geçerliliği	<ul style="list-style-type: none"><li>• Organizasyon dışı kullanıcı sayısı</li><li>• Organizasyon içi kullanıcı sayısı</li><li>• Kullanıcı tecrübe ve destek eksikliği</li><li>• Kullanıcılar tarafından oluşturulan hiyerarşi seviyeleri</li><li>• Ekibin ilgili iş tecrübe eksikliği</li><li>• Ekipteki eleman sayısı</li><li>• Rol tanımlarının iyi yapılamaması</li><li>• Ekip çeşitliliği, farklılığı</li></ul>	<ul style="list-style-type: none"><li>• Kullanıcı sayısının fazlalığının gereksinimlerin oluşması ve olgunlaşmasına engel oluşu</li><li>• Kullanıcı sayısının fazlalığının gereksinim belirlenme sürecine dahil olma veya sorumluluk almaya etkisi</li></ul>



Listelenen bu temel riskler ve onları tetikleyen ve seviyesi gösteren ölçütler analiz edilerek, bunların giderilmesi amacı ile birçok deney içeren çalışma gerçekleştirilmiştir. Bu çalışmalar sonucunda gereksinim risklerini azaltma amacı ile gereksinim belirleme faaliyetleri üzerinde çözümler aranmıştır. Gereksinim karmaşıklığı için, gereksinim belirleme teknikleri olarak formal, bileşik ve pragmatik yaklaşımlar, gereksinim uygunluk riski için, deneyleme tekniklerinden tekrarlama (*iteration*) ve işbirliği analizi (*collaboration*), gereksinim geçerliliği için ise keşif tekniklerinden, grup çalışması ve gözlemlene gibi paydaşları birbirine bağlayan yaklaşımlar önerilmiştir [6].

Gereksinimler gibi yazılım geliştirme yaşam döngüsünün farklı safhalarında yapılan hataların ürün ve projeye olan etkileri de farklılık göstermektedir. Ortaya konulan birçok araştırma proje başlangıcında belirlenemeyen hataların etkisinin yazılım yaşam döngüsü boyunca katlanarak arttığını ortaya koymaktadır. Proje ve üründe oluşan hataların kaynağı araştırıldığında ise Şekil 1.1'deki dağılım ortaya çıkmaktadır [2]. Gereksinim hatalarının proje boyunca ortaya çıkan tüm hataların %50'sinden fazlasını oluşturduğu görülmekte ve bu hataların ileriki safhalarda çözülmesinin zorluğu ve sebep olacağı yüksek maliyetler düşünüldüğünde gereksinim hatalarının gereksinim yönetim süreci içerisinde olabildiğince erken bulunarak çözülmesi gerekmektedir.



Şekil 1.1: Geliştirme yaşam döngüsünde oluşan hataların dağılımı [2]

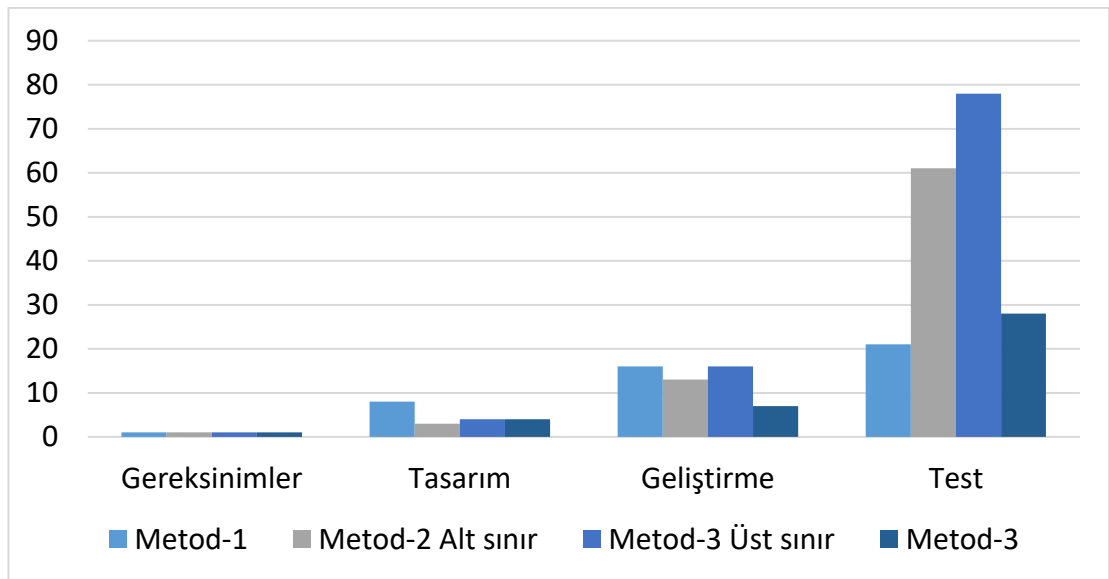
Farklı safhalarda ortaya çıkan bu hatalar için tasarımsal, fonksiyonel, mantıksal, performans, gereksinim ve kullanıcı ara yüzü gibi farklı gruplar tanımlanmıştır [7]. Bu potansiyel hataların test aşamasından önce giderilmesi için

gereken maliyetin sonraki safhalarda karşılaşılabilecek maliyet ile kıyaslanması Tablo 1.3’de özetlenmiştir [2]. Örnek vermek gerekirse, 2000’li yıllardan önce tasarlanmış birçok yazılımın barındırdığı Y2K hatasının birçok ciddi etkileri olmuştur. (Y2K: yılların yazılımlarda son iki rakamı ile tutulması ve 1999 yılından 2000 yılına geçişte bu iki hanenin “99”dan “00”a düşmesi nedeniyle sorun yaşanması) Önceden fark edilemeyen bu hatanın sonradan çözümü ise firma ve kurumlara milyonlarca dolara mal olmuştur [8].

**Tablo 1.3:** Hataların giderilmesine dair göreceli maliyet oranları [2]

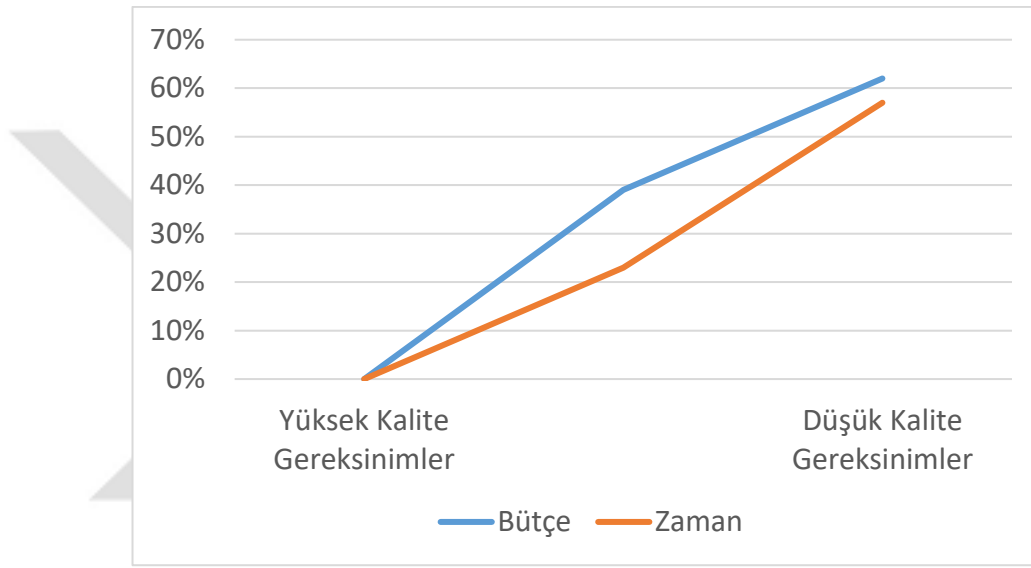
Yazılım Yaşam Döngü Safhası	Maliyet Oranı
Gereksinimler	1
Tasarım	3-6
Kodlama	10
Birim / Entegrasyon testleri	15-40
Sistem ve Kabul testleri	30-70
Üretim	40-1000

Benzer şekilde 2004 yılında NASA tarafından ortaya konulan bir diğer çalışmada birçok büyük çaplı yazılım ve sistem projesi, hataların farklı safhalarda ayıklanma maliyetlerinin değişkenliği incelenmiştir. Farklı metodolojilerin kullanıldığı araştırmada hataların giderilmesine bağlı maliyet oranı kıyaslamaları Şekil 1.2’de özetlenmiştir [12].



**Şekil 1.2:** Sistem hatalarının giderilmesinde maliyet oranları [12]

Gereksinimlerin ortaya konduğu ve gözden geçirilerek resmileşmesi gerektiği ilk safhada hataların ayıklanmasının projelere ek hiçbir maliyet getirmediği kullanılan tüm ölçüm metotları ile ortaya konmuştur. [12] İlerleyen tasarım, geliştirme ve test aktiviteleri için sonuçlar ölçüm metotları arasında farklılık gösterse de maliyetlerin ilerleyen aşamalarda doğrusal bir şekilde arttığı netçe görülebilmektedir. Maliyetleri bütçe ve zaman olarak ayırdığımızda gereksinimlerin kalitesine bağlı olarak benzer şekilde her ikisinde de kaybın %60'lara ulaştığı Şekil 1.3'de görülmektedir [12].



Şekil 1.3: Gereksinim kalitesinin proje bütçe ve zaman kalemlerine etkisi [12]

Projelerin başarısında büyük etkileri olan bu hataların engellenebilmesi için öncelikle hatalar belirlenip, sebepleri analiz edilerek çözümleri üzerine yöntem ve yaklaşımlar geliştirilmelidir. Bu amaçla yapılan bir çalışmada genel olarak projelerde başarısızlığa sebep olan gereksinim süreci özelindeki sorun ve hatalar on iki madde ile özetlenmiştir [4];

1. Düşük kalite gereksinim yazımı
2. Sadece basit kullanıcı senaryolarının kullanımı
3. Uygun olmayan kısıtlar
4. İzlenebilirliği olmayan gereksinimler
5. Eksik gereksinimler
6. Yönetilmeyen kapsam kaymasına bağlı aşırı gereksinim volatilitesi
7. Yetersiz gereksinim kalite doğrulaması

8. Yetersiz gereksinim kalite geçermesi
9. Yetersiz gereksinim yönetimi
10. Yetersiz gereksinim süreci
11. Yetersiz yazılım ve araç desteği
12. Hazırlıksız gereksinim mühendisleri

Listelenen maddeler kimi zaman birbirini kapsayan süreçleri ifade etse de projeleri başarı, bütçe ve zaman gibi kriterler açısından zora sokan gereksinimler ile ilgili tüm potansiyel sebepleri listelemektedir. Özellikle ilk madde; gereksinim kalitesinin düşüklüğü, diğer tüm maddeleri içeren ve onlar tarafından tetiklenen tüm gereksinim sorunlarının bir bileşim kümesi olarak da ifade edilebilir. Düşük kalite gereksinimlerin altında, projelerde dokümantasyon süreçlerine diğer geliştirme ve üretim aktivitelerine kıyasla daha az önem verilmesi ve hızlı bir şekilde tamamlanması gerektiği anlayışı yatmaktadır. Gereksinim dokümanları da projenin devam eden süreçlerinin hızlı bir şekilde beslenip tetiklenmesini sağlamak amacı ile genellikle hızlı ve yetersiz gözden geçirme ve doğrulama faaliyetleri ile beraber hazırlanmaktadır. Bu sebeple muğlak, uyumsuz, eksik, tutarsız, yanlış, çelişkili, anlaşılmalı fazla teknik jargon içeren, hayata geçirilmesi mümkün olmayan, hedeflenen sistemden sapmış ve doğrulanıp geçirilemeyen gereksinimler ortaya çıkabilmektedir [13]. Kalitesiz gereksinimlere işaret eden tüm bu özellikler diğer taraftan gereksinim gözden geçirme aktivitesinin nasıl yapılacağını ve temel olarak gereksinim gözden geçirme kontrol listesi maddelerini işaret eden kriterleri ifade etmektedir. Bir diğer önemli hata da gereksinimlerin hangi dokümanlardan türetildiğini ve hangi üst gereksinimleri karşıladığını gösteren izlenebilirliklerin eksikliğidir. İzlenebilirlik problemleri basit dokümantasyon sorunları gibi görünse de hedeflenen üründen sapma, müşteri beklentilerinin karşılanamaması, ciddi tasarım hataları ve eksik fonksiyonlar gibi sonuçlara neden olabilmesi sebebi ile yine gereksinim gözden geçirme kontrol listelerinin kritik maddelerinden biri olmak durumundadır.

Yazılım geliştirme yaşam döngüsü başlangıç kavramsal tasarımdan, ürünün bakım ve destek aktiviteleri eşliğinde kullanımdan kalkıncaya kadar birbirini takip eden farklı süreçlerden oluşmaktadır. Gereksinim yönetim sürecini de içeren döngü dokuz ayrı fazdan oluşmaktadır [16]:

- a. Kavramsal tasarım ve başlangıç
- b. Gereksinimler

- c. Mimari tasarım
- d. Detay tasarım
- e. Geliştirme
- f. Entegrasyon ve test
- g. Kabul ve teslimat
- h. Operasyonlar ve bakım
- i. Kullanımdan kalkış

Yaşam döngüsü boyunca tüm fazlarda ve özellikle son üründe başarıyı yakalamada en önemli rolü hatasız ve doğru tanımlanmış gereksinimler üstlenmektedir. Gereksinimlerin doğruluğu ve kalitesi döngü boyunca farklı süreçler ile desteklense de, gereksinim gözden geçirme süreci gereksinim doğrulama ve geçirme aktivitelerinin başlangıcı ve hatanın ortadan kaldırılabileceği ilk aşama olması sebebi ile gereksinim yönetim sürecinin en önemli aktivitelerindendir. Proje başlangıcında ayıklanamayan her hata projenin diğer safhalarına etkisi bir kartopunun çığa dönüşmesine benzetilebilir. Gereksinimler üzerine kurulan tasarım, geliştirme ve test aktivitelerinden önce ayıklanan gereksinim hataları birçok maliyetin ortaya çıkmadan önce ortadan kaldırılması anlamına gelmekte, bu da tüm proje paydaşları için her anlamda kazancı ifade etmektedir.

Gereksinim yönetim süreci yapısal olarak, gereksinimlerin belirlenip hazırlanması, belirli bir konfigürasyon kontrolü altında tutulup güncellenmesi, üst veya alt seviye dokümanlara bağlanması, veya izlenebilirliklerinin kurulması gibi alt süreçlerden oluşmaktadır. Tüm bu süreçlerin sorunsuz bir şekilde gerçekleştirilmesi ancak belirli yazılım ve araçların desteği ile sağlanabilmektedir. Farklı sektörlerde ihtiyaç ve maliyetlerine göre farklı yazılım ve araçlar tercih edilmekte ve bu yazılımlar için hazırlanan en kapsamlı liste Ek A'da sunulmuştur.

Proje başlangıcında seçilen gereksinim yönetim yazılımları, gereksinim yönetimine ek olarak proje yaşam döngüsü boyunca diğer tüm proje süreçlerini de desteklemektedir. Sistem ve yazılım tasarımı, geliştirilmesi, kodlanması ve doğrulanması gibi birçok sürece doğrudan etki ettiği gibi son ürünün de hatasız bir şekilde ortaya konması gereksinim yönetim sürecindeki başarıya bağlıdır. Konfigürasyon yönetimi ile desteklenen bu süreçte, proje başlangıcında seçilmiş olan yazılım ve araçların, projenin ilerleyen aşamalarında kullanım zorluğu, yetersiz fonksiyonlar, hatalar, performans sorunları, komut dosyası (*script*) desteğinin yetersiz olması veya diğer kullanılan araç ve yazılım ile entegre olamaması gibi

sebepler ile ihtiyaçları karşılamadığı fark edildiğinde, artık proje açısından araç deęiřtirmenin mümkün olmadığı ve var olan aracın eksikliklerinin getirdiđi ek maliyetlerinin karşılanmak durumunda kalındığı bir süreç başlamıştır. Bu sebeple gereksinim yönetimi için seçilecek olan yazılım ve araçlar, tüm özellikleri, maliyetleri, güçlü ve zayıf yönleri açısından karşılaştırılarak, projeye en uygun olanın seçilmesi, gereksinim yönetim süreci başarısında ve buna bađlı olarak proje başarısında kritik rol oynayacaktır.

Projeleri geliřtiren kurumlar iç süreçlerine, proje ihtiyaç ve dinamiklerine, müşteri isteklerine, sektör standartlarına, denetim otorite kurallarına ve diđer tüm paydařların etkilerine bađlı olarak en dođru yazılımı seçmek durumundadırlar. Bu sebeple sektörde var olan tüm yazılımları incelemek, karşılaştırma yaparak kendilerine en uygun aracı arařtırmaları önemlidir. Bu süreç için her kurumun yeterli vakit ve kaynak ayırması zor olduđundan, genelde gereksinim yönetim aracı geliřtiren firmaların demo ve tanıtımları incelenerek kısa zaman içerisinde, çođunlukla maliyetlere bađlı kararlar verilmektedir. Bu hızlı kararlar, proje yařam döngüsü boyunca kullanılan araçta ortaya çıkan bazı kullanım zorlukları, hatalar veya eksik özellikler proje başlangıcında ayrılamayan maliyetlerden çok daha fazlasına neden olabilmektedir. Firmaların tanıtım ve demolarının bir diđer handikabı da bađımsız olmamaları ve sadece güçlü yönlerini ön plana çıkarmalarıdır. Bu sebeple bađımsız bir otoritenin yapacağı bir karşılaştırma tüm kullanıcılar için önemli bir ihtiyaçtır. Bu ihtiyaçı karşılamaya yönelik nadir ve en kapsamlı çalışmalar INCOSE ve Seilevel kurumları tarafından, Ek A- Gereksinim Yönetim Araçları Listesi'nde verilen 108 gereksinim yönetim aracı üzerinde, Ek B- Gereksinim Yönetim Araç Karşılaştırma Kriterleri'nde belirtilen 233 kritere göre yapılmış ve en başarılı yazılımlar sıra ile Tablo 1.4'de listelenmiştir [3]. Bu araçlar ve verilen detaylı karşılaştırma kriterleri gereksinim gözden geçirme sürecine ve tezde ortaya konulacak otomatik gözden geçirme yaklaşımına önemli ölçüde girdi sağlamaktadır. Gereksinimleri saklama ve yönetme şekilleri, özellikleri, dosya formatları ve gereksinimleri yazılımdan dışarı veya dışarıdan yazılıma aktarma şekilleri otomatik gözden geçirme sürecini yapısal olarak etkileyen faktörlerdendir.

**Tablo 1.4:** En başarılı yazılımlar

1	IBM Rational DOORS
2	Siemens Teamcenter
3	Blueprint Requirements Center
4	eDevTECH inteGREAT Requirements Studio
5	IBM Rational Composer
6	3SL Cradle
7	Microsoft Team Foundation Server
8	Jama Software Contour
9	Polarion Requirements
10	HP Quality Center
11	Orcanos Qpack
12	TraceCloud
13	Sparx Systems Enterprise Architect
14	Kovair Application Lifecycle Management
15	TechnoSolutions TopTeam Analyst
16	MKS Integrity
17	Micro Focus Caliber RM/RM

Gereksinimlerin doğru ve anlaşılabilir bir şekilde oluşmasını destekleme amacı ile farklı kurum ve organizasyonlar tarafından gereksinim yönetim süreçleri üzerine sektörlere özel yol gösterici ve tavsiye niteliği taşıyan dokümanlar oluşturulmuştur. Bu dokümanlar gereksinimlerin belirlendiği başlangıç fazından, proje boyunca gereksinimlerin yaşadığı her safhaya belirli kural ve yönlendirmeler ile katkı vermektedirler. Gereksinim gözden geçirme süreci özelinde, gereksinimlerin hangi kriterlere göre nasıl gözden geçirilmesi gerektiğini tanımlayan kontrol listeleri sunulurken, aynı zamanda gözden geçirmeye katılan paydaşların gözden geçirme süreci boyunca nasıl davranmaları gerektiğini detaylı anlatan rol ve sorumluluk tablolarına kadar tüm süreç dokümanına edilmektedir. Bu sayede gereksinimlerin olabildiğince eksiksiz, standartlara uygun, objektif bir şekilde gözden geçirilmesi hedeflenirken, sübjektif ve kişiden kişiye değişen bakış açısı veya yaklaşımların önüne geçilmesi amaçlanmaktadır. Bu standartlardan önce çıkan ve tezde faydalanılacak olanlar ISO 9126-1, IEEE Std 830-1993, RTCA DO-178B, NASA STD.

Hangi sektörde olursa olsun proje başlangıcında gereksinimlerin açık, net ve hatasız belirlenebilmesi gereksinimlerin belirli bir standart ve kontrol listesine göre başarı ile gözden geçirilmesine bağlıdır. Fakat genelde bu kontrol listesinin yetersizliği, gözden geçirecek proje paydaşlarının yeterli zamanı ayıramaması, standart veya kontrol listesi olmaksızın sübjektif gözden geçirmeler sebebi ile yanlış anlaşılmaya açık veya hatalar içeren gereksinimler ortaya çıkabilmektedir. Bu sebeple sektör veya proje yapısına özel kontrol liste maddeleri haricinde, tüm yazılım projeleri için ortak bir kontrol listesinin oluşturulması ve mümkün olduğunca bu listenin otomatik kontroller ile gözden geçirilmesi gereksinim yönetim sürecine çok önemli katkılar sağlayacaktır. Tez ile hedeflenen bu yaklaşımın arkasında yatan temel motivasyon ve ilham kaynağı maddeler aşağıdaki gibi sıralanabilir;

1. Proje yaşam döngüsü seviyelerinde var olan hataların dağılımları ve gereksinim yönetim sürecinin dağılımdaki oranı,
2. Proje yaşam döngüsü boyunca farklı seviyelerde hataların ayıklanma maliyetlerinin gereksinim yönetiminin sürecine oranla çok yüksek olması
3. Gözden geçirme sürecinin manuel olarak proje ekipleri tarafından yapılması ve ilgili kalite, maliyet, zaman gibi handikaplar,
4. Doğal dilde yazılan gereksinimlerin kişiden kişiye farklı yorumlanabilmesi ve sübjektif yaklaşımlar,
5. Proje takvim ve zaman kısıtları altında gözen geçirme süreçlerinin göz ardı edilmesi,
6. Farklı sektörlerden tüm yazılım proje gereksinim gözden geçirme süreçlerine uygulanabilir temel bir Türkçe Gereksinim gözden geçirme kontrol listesi eksikliği,
7. Otomasyona uygun Türkçe Gereksinim gözden geçirme kontrol listesi eksikliği,
8. Gözden geçirme sonuç raporlarının otomatik ve standart olarak üretilme ve denetim otoritelerine sunum için resmi dokümantasyon ihtiyacı,
9. Gereksinim gözden geçirme sonuçlarının metrik ve istatistiki anlamda yetersiz olması, detaylı ve standart ölçümler ve ilgili metrikler ışığında yeni projelere planlama, süreç ve mühendislik iyileştirmeleri anlamında katkı yapabilme ihtiyacı,



10. Gereksinimlerin oluşturulması sürecinde elimine edilemeyecek olan insan faktörü ve potansiyel hataların, gereksinim gözden geçirme sürecinde otomasyon ve bağımsız analizler ile ortadan kaldırılması.

Bu motivasyon ile tezin birinci bölümünde gereksinim yönetim sürecinin tüm safha ve paydaşları göz önüne alınarak, süreç girdileri, çıktıları, riskleri, proje metrikleri üzerinden sık yapılan hata analizleri, hataların etki ve maliyetleri, kullanılan gereksinim yönetim araçları ve özellikleri, kullandıkları gereksinim ve dosya formatları, gereksinim yönetim süreci ile ilgili ortaya konmuş sektörel standartlar detaylı incelenerek bir alt yapı sağlanmış ve tezde ortaya konulan yazılımın temelini oluşturan otomatik gereksinim gözden geçirme kontrol listesi (OGGKL) de bu sayede şekillenmiştir. Gereksinimlerin otomatik analizinde kullanılacak olan OGGKL, daha önce Türkçe ve İngilizce gereksinimler için ortaya konulan otomatik gözden geçirme kriter ve kontrol listeleri ile kapsam ve içerik olarak kıyaslandığında her biri farklı analizleri tetikleyen 20 ayrı maddeden oluşan zengin bir kontrol listesi ile ön plana çıkmaktadır.

Ortaya konulan OGGKL'nin hayata geçirilmesi, doğruluk ve kapsam olarak başarısının değerlendirilmesi bu kontrol maddelerini otomatize edecek olan yazılım ve içerdiği dil işleme algoritmaları ile gerçekleştirilebilir. Bu yazılımı geliştirmeden önce literatürde ön plana çıkan İngilizce ve Türkçe gereksinimleri otomatik olarak analiz eden tüm yaklaşımlar detaylı incelenerek güçlü ve zayıf yönleri karşılaştırılmıştır.

Literatür var olan çalışmaların eksiklikleri ve zayıf yönleri tez için geliştirilen yazılıma yön verme adına ciddi katkıları olmuştur. Bu tecrübeler ile yeni ve gelişmiş bir otomatik gözden geçirme yaklaşımı ortaya koymak amacıyla en güncel doğal dil işleme yaklaşımları incelenmiş ve Türkçe üzerine öne çıkan İTÜ doğal dil işleme yazılım zinciri [33] belirlenerek otomatik gözden geçirme yaklaşımına entegre edilmiştir.

Tezde ortaya konulan yaklaşım ve OG3 gözden geçirme yazılımının başarısını ölçme ve değerlendirme amacı ile yapılacak deneylerde karşılaştırmaya uygun ve geniş bir kontrol listesine sahip Türkçe gereksinim gözden geçirme yazılımı bulunmadığından deneyler kapsamında İngilizce gereksinimler üzerinde analiz yapan ve birçok akademik çalışmaya konu olmuş Tiger Pro yazılımı belirlenmiştir. Deneysel çalışmalara girdi olarak farklı sektörlerden 6 ayrı gereksinim dokümanı

seçilmiş ve bu dokümanların İngilizce versiyonları Tiger Pro yazılımı ile gözden geçirilmiştir. İngilizce gereksinimler bir makine çevirici ile Türkçe 'ye çevrilerek OG3 yazılımına veri seti olarak sunulmuştur. Daha önce ürüne dönüşmüş bu olgun gereksinim dokümanlarında çeviri sırasında oluşabilecek bozulmalar manuel gözden geçirme ile engellense de, gözden kaçabilecek hatalar, hata ayıklama amacı ile toparlanan bu veri setine zenginlik katacaktır. Tez sonunda yapılan deneyler sonucunda elde edilen sonuç ve raporlar ışığında her iki ayrı yazılım hem birbirleri ile kıyaslanmış, hem de kendi özellik ve fonksiyonları özelinde bireysel analizler ile değerlendirilmişlerdir.



## İKİNCİ BÖLÜM

### LİTERATÜR ÖZETİ

Yazılım ve sistem mühendisliği alanında geliştirilen projelerin büyüklük, kapsam, zorluk derecesi, bütçe, insan gücü ve görev kriterleri düşünüldüğünde, projelerin başarısı, gecikmesi veya son üründe oluşabilecek hataların insan ve çevre sağlığından, firma, kurum ve ülke ekonomisine kadar farklı sonuçlara sebep olduğu görülmektedir. Geliştirilecek olan ürünlerin sahip oldukları fonksiyonların önemi, projelerin bütçe ve verimlilik anlamında zorlayıcı takvimleri, proje paydaşlarının beklentileri, yetkilendiren veya denetleyici kurumlar proje süreçleri içerisinde gereksinim yönetimine ayrı bir önem kazandırmaktadır. Gereksinim yönetim süreci içerisinde ise hataların ortadan kaldırılmasında en önemli safha olan gereksinim gözden geçirme süreci sektörlere ve ilgili proje süreçlerine yön veren kurum ve kuruluşların oluşturdukları belirleyici standartlar ile şekillenmiştir. Gereksinim gözden geçirme süreci özelinde literatürde öne çıkan ve tez boyunca yapılacak çalışmalara temel oluşturacak standartlar sırası ile gereksinimler özelinde özetlenmiştir.

#### 2.1 Standartlar

##### 2.1.1 ISO/IEC FDIS 9126-1

Uluslararası bir standart olan ISO 9126-1, yazılımlarda kalite standardı oluşturma adına yazılım kalite kontrol, yazılım kalite güvence ve yazılım süreç iyileştirme konularını temel almıştır. Bir kalite modeli sunulan standartta, kalitenin ölçümü adına iç ve dış metrik ve ölçümler işaret edilmektedir. Yazılım için oluşturulan kalite modeli hem son ürün için hem de ürüne giden yolda diğer süreçler için de yol gösterici durumdadır. Bu sebeple yazılım kalitesi için oluşturulan nitelikler yazılımı tanımlayan ve belirleyen gereksinimler içinde kalite kontrolü adına oluşturacak kontrol

listelerine temel oluşturmaktadır. ISO/IEC 9126’da yazılım gözden geçirmeye temel oluşturacak yazılım kalite nitelikleri Tablo 2.1’de listelenmiştir [19].

**Tablo 2.1:** ISO/IEC 9126’ya göre yazılım kalite nitelikleri [19]

Fonksiyonellik	Yazılım belirlenmiş şartlar altında kullanıldığında ima edilen ve ifade edilen ihtiyacı duyulan fonksiyonları sağlayan yazılım becerisidir. <ul style="list-style-type: none"> <li>• Uygunluk, doğruluk, karşılıklı işlerlik, uyumluluk, güvenlik</li> </ul>
Güvenilirlik	Belirlenmiş şartlar altında kendi performans seviyesini yazılımın muhafaza edebilme becerisidir. <ul style="list-style-type: none"> <li>• Olgunluk, hata toleransı, kurtarılabirlik</li> </ul>
Kullanılabilirlik	Belirlenmiş şartlar altında anlaşılacak, öğrenilecek, kullanılacak ve kullanıcı tarafından beğenilecek yazılım olma becerisi olarak tanımlanmıştır. <ul style="list-style-type: none"> <li>• Anlaşılabilirlik, öğrenilebilirlik, işlerlik</li> </ul>
Etkinlik (Verimlilik)	Etkinlik kullanılan kaynağa bağlı olmak kaydıyla ifade edilmiş olan şartlar altında yazılımın gereken performansı sağlama kabiliyetidir. <ul style="list-style-type: none"> <li>• Zaman davranışı, kaynak yararlanımı,</li> </ul>
Bakım Yeteneği ve Kurtarılabirlik	Çalışma çevresinde, gereksinimlerde ve fonksiyonel özelliklerde değişiklik söz konusu olduğunda yazılımın düzeltilmesini, geliştirilmesini ve adaptasyonunu kapsayabilmektedir. <ul style="list-style-type: none"> <li>• Analiz edilebilirlik, değiştirilebilirlik, durağanlık, test edilebilirlik</li> </ul>
Taşınabilirlik	Bir çalışma ortamından diğerine aktarılabilme becerisini ifade etmektedir. <ul style="list-style-type: none"> <li>• Adapte Edilebilirlik, kurulum kolaylığı, uygunluk, yer değiştirilebilirlik,</li> </ul>

### 2.1.2 IEEE Std 830-1993

Yazılım gereksinimlerine standart getirme adına IEEE tarafından ortaya konulmuş ve gereksinim yazım süreci için yönlendirmeler içeren standart aynı zaman dokümanlar için bir taslak sunmuştur. Tablo 2.2’de standartta tanımlı iyi gereksinim nitelikleri özetlenmiştir.

**Tablo 2.2:** IEEE Std 830-1993’da tanımlı iyi gereksinim nitelikleri [17]

Doğru	Gereksinimler müşteri ihtiyaçlarını karşılayacak şekilde üst seviye gereksinimler ile birebir uyumlu ve doğru olmalıdır.
Muğlak Olmayan	Gereksinimler sadece tek bir anlam ifade edecek, herkes tarafında aynı şekilde anlaşılacaktır.
Tamamlanmış	Gereksinimler her anlamda eksiksiz olmalı ve DSB gibi ifadeler içermemelidir.
Tutarlı	Kendi seviyesindeki gereksinimler ile çelişmemelidir.
Öneme göre sıralama	Gereksinimlerin hepsi farklı neme sahipti ve bunu belirten tip veya tanımlara göre sınıflandırılmalıdırlar.
Doğrulanabilir	Gereksinimlerin ifade ettiği fonksiyonellik doğrulanabilir olmalıdır. İmkânsız, sağlanamayacak ifadelerden arınmış olmalıdır.
Güncellenebilir	Gereksinimler kolayca güncellenebilir şekilde dokümanite edilmelidir.
İzlenebilir	Gereksinimler hem ileri hem de geri izlenebilirliklere sahip olmalıdır.

### 2.1.3 RTCA DO-178

DO-178 dokümantasyonu (*Considerations in Airborne Systems and Equipment Certification*) 1992 yılında Amerika’da RTCA (Havacılık Radyo Teknik Komisyonu) içerisinde özel bir komite (SC-167) tarafından yazılımların uçuşa elverişlilik isterlerini karşılamak amacı ile yönlendirici ve tavsiye nitelikleri taşıyan bir rehber olarak hazırlanmıştır. Dokümanın FAA (Federal Havacılık İdaresi) tarafından referans alınması sonrasında bir standart haline geldiği söylenebilir. Yıllar içerisinde sektörlerden gelen dönüş ve tecrübeler ile yenilenen doküman RTCA ve EUROCAE kontrolünde güncellenmektedir. (DO-178A/B/C) Yazılım geliştirme yaşam döngüsü boyunca hedeflenen emniyet ve sertifikasyon isterlerinin karşılanabilmesi için her sürece ve ilgili dokümanlarına özel yönlendirmeler içeren doküman, yazılım gereksinimleri ve gözden geçirme sürecine ve gereksinim gözden geçirme kontrol listelerine (*checklist*) temel oluşturacak kriterler de sunmuştur [20].

**Tablo 2.3:** DO-178B yazılım gereksinim gözden geçirme ve analiz kriterleri [20]

1	Üst seviye yazılım gereksinimleri sistem gereksinimleri ile uyumlu olmalıdır.
2	Üst seviye yazılım gereksinimleri doğru ve tutarlı olmalıdır.
3	Üst seviye yazılım gereksinimleri yazılımın üzerinde çalışacağı hedef donanım ile uyumlu olmalıdır.
4	Üst seviye yazılım gereksinimleri doğrulanabilir olmalıdır.
5	Üst seviye yazılım gereksinimleri standartlar ile uyumlu olmalıdır.
6	Üst seviye yazılım gereksinimleri sitem gereksinimlerine izlenebilirlik kurulmalıdır.
7	Gereksinimlerde tanımlanan algoritmalar doğru olmalıdır.
8	Sistem ve yazılım gereksinimleri karşılık muğlak, tutarsız ve tanımsız koşullar içermeyecek şekilde analiz edilmiş olmalıdır.
9	Üst seviye yazılım gereksinimleri nicel ifadeler içermeli ve gerektiğinde toleranslar tanımlı olmalıdır.
10	Türetilmiş gereksinimler için sistem emniyet değerlendirme süreci uygulanmalıdır.

### 2.1.4 NASA-STD Yazılım Mühendisliği Standartları

Amerika birleşik devletlerinin uzay programlarında sorumlu Ulusal Havacılık ve Uzay Dairesi NASA, yıllardır sürdürdüğü program ve projelerde edindiği tecrübe ve dersler ışığında, gerçekleştirdiği sistem ve yazılımlarda minimum hata hedefini gerçekleştirmek üzere mühendislik, teknik ve tüm proje süreçlerini içeren kendi

kurum standartlarını oluşturmuştur. Bu standartları yıllar geçtikçe kendi alt yüklenicilerinin ve diğer tüm havacılık ve teknoloji sektörlerinin de zamanla paylaşımına açmış ve bu standartlar birçok kurum ve kuruluş için örnek ve yönlendirici temel standartlar halini almıştır. Yazılım geliştirme yaşam döngüsü süreçlerine farklı farklı standartlar yayınlayan NASA, gereksinim mühendisliği sürecinde de kaliteyi yakalama ve potansiyel hataların oluşmadan önce ortanda kaldırılması hedefi ile gereksinim gözden geçirme sürecinin hem nasıl işletileceğini hem de gözden geçirmenin hangi kriterlere göre yapılacağını yayınladığı gözden geçirme kontrol listeleri ile sağlamıştır. Tablo 2.4’de gereksinimler için sunulan gözden geçirme kontrol listesi verilmiştir.

**Tablo 2.4:** NASA Yazılım Gereksinim gözden geçirme kontrol listesi [14]

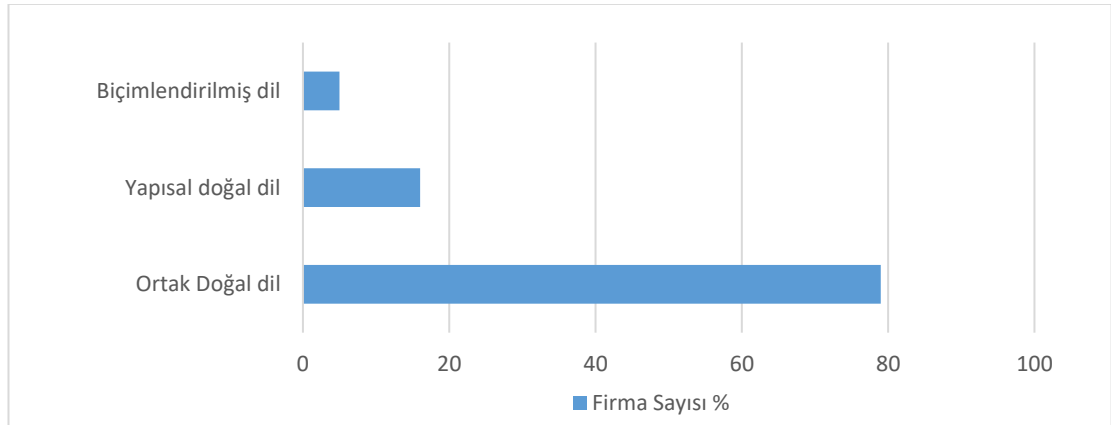
Tamamlanmış	Tüm gereksinimler eksiksiz tamamlanmış mı? DSB gibi ifadeler içeriyor mu?
Açıklık	Gereksinimler bağımsız bir grup tarafından anlaşılıp, kodlanabilecek kadar açık yazılmış mı?
Tutarlılık	Gereksinimler notasyon, terminoloji ve fonksiyonellik anlamında birbirleri ile tutarlı mı? Tüm algoritmalar uyumlu mu?
Test edilebilirlik	Gereksinimler bağımsız bir ekip tarafında test edilebilir mi?
Tasarım bağımsız	Gereksinimler tasarım detayları gibi nasıl yapılacağından çok neyin yapılması gerektiğini açıkça anlatıyor mu?
Okunabilirlik	Gereksinimler tüm paydaşlar tarafından anlaşılacak şekilde yazılmış mı?
Detay seviyesi	Gereksinimler yeterli detay seviyesine sahip mi? Ekstra açıklama gerekir mi?
Gereksinim Tekliği	Gereksinimler tek bir fonksiyon, değer, tanım, konsepti mi tanımlıyor? Birden fazla gereksinimin birleşimi olmamalı.
Girdi Çıktı tanımları	Gereksinim ara yüzleri girdiler, çıktılar ve veri çevrimleri uygun tanımlanmış mı?
Kapsam	Gereksinimlerin sistem veya yazılımın tasarım kapsamının sınırlarını uygun tanımlamış mı? Eksik gereksinim var mı?
Tasarım Kısıtları	Gereksinimler hedeflenen sistemi performans ve fonksiyonellik anlamında sağlayacak şekilde gerçekçi tanımlanmış mı?

Projelerde var olan hataları olabildiğince erken ve ilerleyen süreçlere bırakmadan ortadan kaldırmanın en uygun zamanı gereksinim gözden geçirme aşamasıdır. Bu sebeple bu sürecin verimliliğini arttırmak amacı ile birçok akademik ve sektörel çalışma ortaya konmuştur. Süreci teknik ve yönetsel detaylarını belirleyen standartlar, gözden geçirme sürecinde dikkat edilmesi ve kontrol edilmesi

gereken kontrol listeleri ve bu sürecin yeterli ve doğru şekilde yönetildiğın ispatlanması adına denetlenmesi gibi birçok yaklaşım orta konmuştur. Gereksinin gözden geçirme sürecinin otomasyonu da hedeflenen fakat gereksinimlerin doğal dilde yazılmış olması nedeni ile tam anlamı ile gerçekleştirilemeyen aktivitelerden biridir. Otomasyonu sağlamak ve kolaylaştırmak amacı ile gereksinimlerin yapısal, model bazlı ya da belli formatlar ile doğal dillerden uzaklaştırılarak yazılması da diğer yaklaşımlardandır. Fakat bu yaklaşımlar, gereksinim yönetim sürecinin sadece teknik personeller ile değil, kontrat uzmanı, yönetici, denetmen vs. gibi farklı paydaşlar ile yürütülmesi ve müşteri dâhil tüm bu paydaşların gereksinimlerden aynı şeyi anlayıp, takibini yapabilme ihtiyaçları sebebi genellikle uygulanabilir değildir.

## 2.2 Otomatik Gereksinim Gözden Geçirme

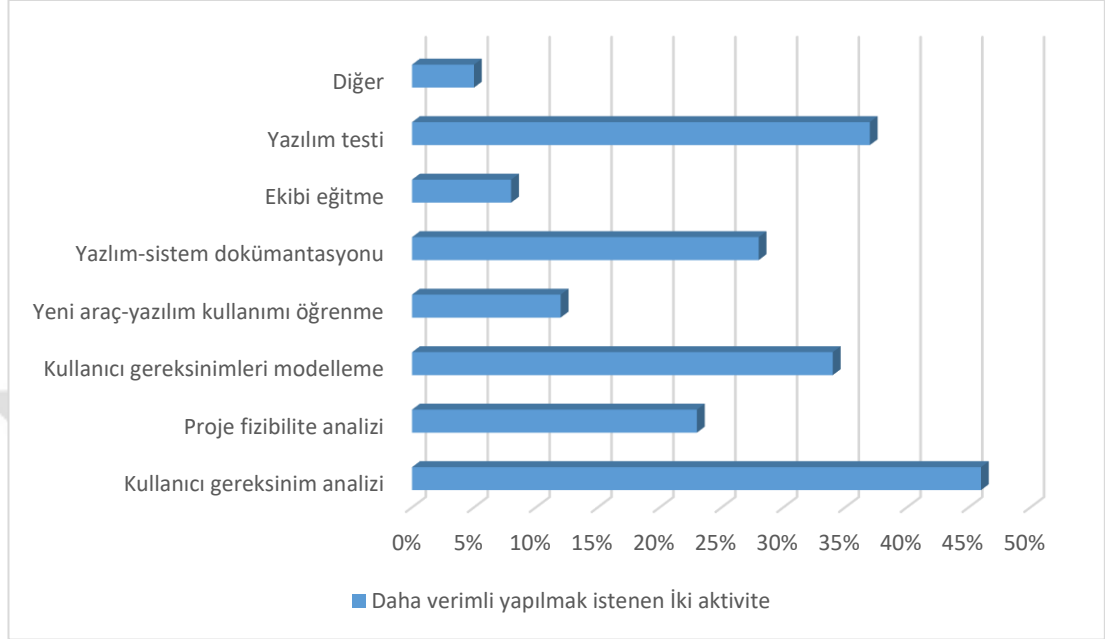
Gereksinimlerin tüm paydaşlar tarafında anlaşılabilirliği ve müşteri ihtiyaçlarının tam anlamı ile hayata geçirilip geçirilemediğinin kolayca takip edilmesi amacı ile gereksinimlerin doğal dilde yazılma ihtiyacının sektörlere yansması Şekil 2.1'deki grafikte özetlenmiştir. Farklı sektörlerden firmalar üzerinde yapılan anketlerde, savunma sanayi ve havacılık projelerindeki gibi katı kural ve denetim otoritelerinin yönlendirmeleri olmamalarına rağmen firmaların %79'unun gereksinim tanımlarken doğal dil tercih ettikleri tespit edilmiştir.



Şekil 2.1: Firmaların proje gereksinimlerindeki terminoloji kullanımı [21]

Avrupa ve Kuzey Amerika ülkelerine ait farklı sektörlerden aynı firmalar üzerinde yapılan çalışmada sorulan bir diğer anket sorusu da gerçekleştirdikleri projelerde hangi iki şeyi daha verimli yapmak istedikleridir. Bu soru için verilen

cevapların oranlarının verildiği Şekil 2.2'ye göre, firmalar %46 gibi yüksek bir oranla en çok kullanıcı gereksinimlerini daha iyi ve verimli şekilde analiz etmek istemektedirler [21].



Şekil 2.2: Projelerde önem verilen ve geliştirilmek istenen aktiviteler [21].

Kullanıcı gereksinimlerini daha iyi ve verimli analiz etmek için en iyi yolun %69 ile otomasyon, %22 ile iç kaynaklardan yeni analiz ekibi, %9 ile alt yüklenicilere analiz görevinin devredilmesi olarak ortaya çıkmıştır [21]. Yapılan anket ve çalışmaların ışığında firmaların proje başarılarında kritik rol oynayan gereksinimlerin otomatik olarak analiz edilerek kalitelerinin artırılmasının bir ihtiyaç olarak ortaya çıktığı açıkça büyük bir ihtiyaç olarak görülmektedir.

Doğal dillerde yazılım gereksinimlerin otomatik olarak gözden geçirilmesi doğal dil işleme algoritma ve yaklaşımları ile hayata geçirilebilmektedir. Makine dillerine göz önüne alarak belirli bir yapıda ve kurallara göre oluşturulmuş dillerin işlenmesi ne kadar kolay ise, insanların kullandığı ve birbirinden farklı özelliklere sahip doğal dillerin işlenmesi de bir o kadar zordur. Doğal dillerin yapısal özelliklerine bağlı olarak dil işleme algoritmaları da değişkenlik göstermektedir. Literatürü incelediğimizde, çoğunlukla en çok kullanılan dillerden olan İngilizce üzerine doğal dil işleme çalışmaları ve yine bazı İngilizce gereksinim gözden geçirme denemelerine (araçlarına, yazılımlarına) rastlamaktayız. Türkçe gereksinim



gözden geçirme için ortaya konulan nadir çalışmalarda da İngilizce için geliştirilen algoritmalar kullanmış ve ciddi başarılar elde edilememiştir [31, 32].

### 2.3 Doğal Dil İşleme (NLP)

İnsan dilleri olarak da adlandırılabilir olan doğal diller üzerine çalışmalar ilk olarak 1950-60 yılları arasında diller arası çevrim ve bazı yapay zekâ çalışmaları ile başlamıştır. Yıllar ilerledikçe gelişen teknoloji ile birlikte makine dilleri ile doğal diller arasındaki etkileşim artmış ve doğal dil işleme ayrı bir disiplin olarak ön plana çıkmıştır. Doğal dil işleme, ana işlevi doğal bir dili işleme, anlama, yorumlama ve üretme olan bilgisayar sistemlerinin tasarımını ve gerçekleştirilmesini konu alan bir bilim ve mühendislik alanıdır [22].

Doğal dillerin algoritmalar ile nasıl işleneceği insanlar tarafından işleme şekilleri incelenerek çözümlenmeye çalışılmaktadır. Kullanılan tüm doğal diller yapısal anlamda birbirlerinden farklılık gösterse de insanlar tarafında işleme şekilleri ortak belirli seviyeler altında gruplandırılabilir. İnsanların doğal dilleri işleme şekilleri yedi ayrı seviye olarak belirlenmiştir [35];

1. Sesbilim (*Phonology*): konuşma ve söylem seslerinin yorumlanması
2. Biçimbilim (*Morphology*): sözcük ve bileşenler
3. Sözlüksel (*Lexical*): kelime anlamlarının yorumlanması
4. Sözdizimsel (*Syntactic*): cümle içindeki sözcüklerin analizi
5. Anlamsal (*Semantic*): cümlenin potansiyel anlamlarını inceleme
6. Söylem (*Discourse*): cümleden uzun söylemlerin analizi
7. Pragmatik (*Pragmatic*): içerikten ziyade bağlam ve duruma göre yorum

Kullanılan doğal dillerin sayısı ve farklı yapıları düşünüldüğünde genel ve tüm dillere uygulanabilir bir dil işleme yaklaşımının ortaya konulması çok mümkün görünmemektedir. Bu sebeple, dil bilimcilerin kendi dillerinin yapısına bağlı olarak değerlendirdikleri ses bilimi, biçim bilimi, söz dizimi ve anlam bilimi gibi konular, dil işleme bilimcileri de tarafından da dillere özel şekilde ele alınmaktadır [23].

Doğal dil işleme algoritmaları işlenecek doğal dile bağlı olarak ciddi farklılıklar gösterebildiği için, literatürde otomatik gereksinim gözden geçirme özelinde, üzerine en çok araştırma yapılan ve diğer doğal dillere yönlendirici olan İngilizce doğal işleme yaklaşımları ve bu tezin esas konusunu oluşturan Türkçe doğal dil işleme yaklaşımları ayrı ayrı detaylandırılacaktır.

## 2.4 İngilizce Yazılım Gereksinimleri için Otomatik Gözden Geçirme Yaklaşımları

Doğal dil işleme çalışmaları öncelikle İngilizce üzerine yapılan çalışmalar ile yaygınlaşsa da daha sonra diğer dillere özel çalışmalar arak arkaya literatürdeki yerini almıştır. Fakat uygulama alanı otomatik gereksinim gözden geçirme olan çalışmalar dünya genelinde çoğunlukla İngilizce özelinde ortaya konmuş, sınırlı sayıda diğer diller için ortaya konulan çalışmalar İngilizce yaklaşımları dönüştürmeyi hedeflediği için genelde başarılı olamamıştır. İngilizce gereksinimlerin otomatik olarak doğal dil işleme yaklaşımları ile analiz edilmesi akademik ve bazı kurumsal proje ihtiyaçları karşılama amacı ile kurumlara özel olarak geliştirilmeye çalışılmış ve ticari hedefler ile ortaya konmamıştır. Fakat yıllar geçtikçe yazılım firmalarının İngilizce gereksinimlerin kalitesi otomatik olarak artırılma ihtiyacının fark edilmesi üzerine ticari yaklaşımlar ortaya konulmuştur.

### 2.4.1 NASA-ARM

Geliştirdiği projelere süreçsel katkı sağlama adına birçok standart ortaya koyan Ulusal Havacılık ve Uzay Dairesi NASA, yazılım geliştirme yaşam döngüsü aşamalarını iyileştirme ve ihtiyaç durumunda potansiyel otomasyon yaklaşımlarını da araştırmıştır. Bu kapsamda gereksinimlerin kalitesinin otomatik olarak kontrol edilmesi üzerine ortaya konulan çalışmaları da bu alanda öncü niteliğindedir. NASA Yazılım Güvence Teknoloji Merkezi tarafından 90'lı yılların sonuna geliştirilen ARM yazılımı NASA'nın belirlediği kriterleri göre gereksinim dokümanlarını analiz ederek raporlar sunmayı hedeflemiştir.

NASA Yazılım Güvence Teknoloji Merkezi proje kütüphanelerindeki gereksinim dokümanları üzerine yapılan analizler sonucunda gereksinim kalite belirteci olarak dokuz ayrı kategori belirlenmiştir; Zorunluluk, yönlendirme, devamlılık, seçenekler, zayıf anlatım, doküman boyutu, doküman bölümlendirme, okunabilirlik ve doküman derinliği [24]. Belirlenen bu göstergelere özel anahtar kelimeler ve analiz yöntemleri ile Tablo 1.8'de belirtilen NASA Yazılım Gereksinim gözden geçirme kontrol listesi maddelerinin otomatik olarak kontrol edilebilirliği incelenmiştir. Gereksinim kalite niteliklerinin bir kısmının sübjektif değerlendirmelere dayandığı ve bunların dışında kalan nitelikler için bazı göstergeler ile analizin

desteklenebileceği ifade edilmiştir [24]. Sonuç olarak göstergelerin kontrol listesi maddelerinin tamamını kontrol edemediği, fakat belirli bir seviyeyi ifade edecek şekilde bir veya birden fazla kontrol listesi maddesi ile ilişkilendirilmiştir.

Doğal dilde İngilizce olarak hazırlanmış gereksinimleri sözlüksel ve sözdizimsel yaklaşımlar ile incelemeye çalışan yazılım, analiz göstergelerine bağlı istatistikler içeren sonuçlar raporlamaktadır. Fakat bu analizler sonucunda gereksinimlerin kontrol listelerine bağlı olarak ne kadar gözden geçirilebildiği ve eksik kalan analizlerin ne olduğu netçe görülememekte, sadece gereksinimler üzerinde belirli seviyede ölçüm ve analiz yapılabilmektedir.

### 2.4.2 Quars

İngilizce gereksinimlerin otomatik olarak kalitesini kontrol etmeyi hedefleyen Quars yazılımı 1994 ile 2005 yılları arasında adım adım geliştirilmiştir. Gereksinimlerin doğal dil işleme algoritmaları ile analizinde sözlüksel ve sözdizimsel yaklaşımlar temel alınmıştır. Gereksinimlerin kalite analizi için kullanılan gösterge ve hangi yöntemler ile belirlendiği Tablo 2.5’de verilmiştir [25].

**Tablo 2.5:** Quars Kalite göstergeleri

<b>Gösterge</b>	<b>Sözlüksel</b>	<b>Sözdizimsel</b>
Belirsizlik	X	
Öznellik	X	
Zorunluluk (negatif)	X	
İma		X
Zayıflık		X
Yetersiz tanım		X
Okunabilirlik	X	
Çokluk		X

Belirlenen göstergeler üzerinden otomatik kalite analizlerini gerçekleştirebilmek için tasarlanan yazılım mimarisi beş ayrı fonksiyonel bölümden oluşmaktadır; sözdizim ayrıştırıcı, sözcüksel ayrıştırıcı, gösterge tarayıcı, görünüm üretici, sözlükler [25]. Analizler sonucunda hata oranı ve okunabilirlik metrikleri analiz sonuç raporu olarak sunulmaktadır. Hata oranı metriği kalite göstergelere göre hatalı gereksinim cümle sayısının diğer cümlelere oranı, okunabilirlik için ise Coleman-Liau okunabilirlik hesaplama formülü kullanılarak yine bir yüzdesel sonuç raporlamaktadır.

### 2.4.3 QVscribe

Akademik ve bazı kurum içi ihtiyaçları karşılamak için ortaya konulan öncü yaklaşımlardan sonra, ticari amaçlı ve firmalara yine İngilizce otomatik gözden geçirme desteği sağlamak amacı ile yazılımlar oluşturulmaya başlanmıştır. Bunlardan biride QA Corp firmasının sunduğu Qvscribe yazılımıdır. Firma NASA'nın ilk olarak ortaya koyduğu ARM yazılımını temel olarak oluşturulmuş ve yine ARM gibi temel yaklaşımlar gibi sadece sözcüksel ve sözdizimsel yaklaşımlar ile analizler yaparak gözden geçirme sürecine destek olmaktadır. Qvscribe ayrı bir yazılım olarak oluşturulmamış, Microsoft Word üzerine bir eklenti olarak geliştirilmiştir [26].

Qvscribe, temel aldığı ARM yazılımı gibi analizler için kalite göstergeleri belirlemiş ve yine ARM ve Quars benzeri bu göstergeleri kullanmıştır.

1. Zorunluluk
2. Negatif Zorunluluk
3. Seçenekler
4. Belirsizlik
5. Zayıflık
6. Öznellik
7. Devamlılık
8. Direktifler
9. Evrensel nicelikler

### 2.4.4 RQA Client

Ticari amaçla geliştirilen gereksinim kalite analizi yapan yazılım, sistem ve yazılım gereksinimlerindeki potansiyel hataları ortadan kaldırmayı hedeflemektedir. Temel olarak hataları üç boyut üzerinden incelemektedir. Doğruluk, tamamlanmış ve tutarlılık üzerine belirlenen kurallar ile analizler yapılmaktadır [28].

Doğruluk kontrollerinde yasaklı kelime kullanımı, dokümantasyon hatalarına, dilbilgisi, sözcük ve yapısal hatalara yoğunlaşmıştır. Fakat bu kontroller sektörel standartlar açısından bakıldığında gereksinimim doğruluğunu kontrol ettiği kesinlikle söylenemez. Gereksinimlerin doğru yazılmış olması ancak üst seviye gereksinimler ve müşteri gereksinimler ile karşılaştırılarak uyumluluk kontrolü ile yapılabilir. Bu

noktada yazılım yaptığı kontroller yazım çoğunlukla yazım standart uyumsuzlukları ile okunabilirlik kontrolleridir.

Gereksinimlerin tamamlanmış olup olmadığını belirleyen kurallar eksik kelime, ilişki, ara yüz ve gereksinim tipi gibi kontroller üzerine kurulmuştur. Tutarlılık kontrolleri ise sadece birden fazla aynı gereksinim olup olmadığına ve ölçü birimlerinde bir uyumsuzluk olup olmadığını incelemektedir. Yine gereksinimler üzerinde yeterli bir tutarlılık kontrolünde söz edilemez fakat bu kontrollere destek amaçlı otomatik kontroller yapılmaktadır.

#### 2.4.5 Ambiguity Detector

Gereksinim gözden geçirme aktivitesine otomasyon anlamında katkı sağlamak adına Nigam ve arkadaşları, muğlaklık (*ambiguity*) bulucu adında bir yazılım ortaya koymuşlardır. İngilizce gereksinimler üzerinde doğal dilde işleme yöntemleri ile sözlüksel, sözdizimsel ve sentaks hatası analizleri yapıldığı ifade edilmektedir. Birden fazla anlama gelen kelimelerin sebep olduğu muğlaklık ve çok anlamlılık analizlerini sözlüksel, sözcüklerin dizilim sırası sebebi ile farklı anlamlar çıkarılabilecek yapısal sorunlar için sözdizimsel, noktalama hataları veya özne gibi eksikliklerin analiz edildiği sentaks kontrolleri olarak sınıflandırılmıştır [11].

Gereksinimler için sunulan muğlaklık analiz algoritmaları muğlaklığa sebep olan kelime ve yapıların bulunduğu bir metin dosyasını temel almaktadır. Bu dosya içerisinden elde edilen veriler ışığında, gereksinimler satır satır okunarak bu verileri içerip içermediği kontrol edilmektedir [11]. Karşılaştırmalar sonucunda muğlaklık göstergelerini içeren gereksinimler, sözlüksel, sözdizimsel ve sentaks grupların altında toplanarak yüzdesel olarak raporlanır. Test amaçlı dört ayrı gereksinim dokümanını incelenmesi sonucunda raporlanan sonuç Tablo 2.6'de verilmiştir.

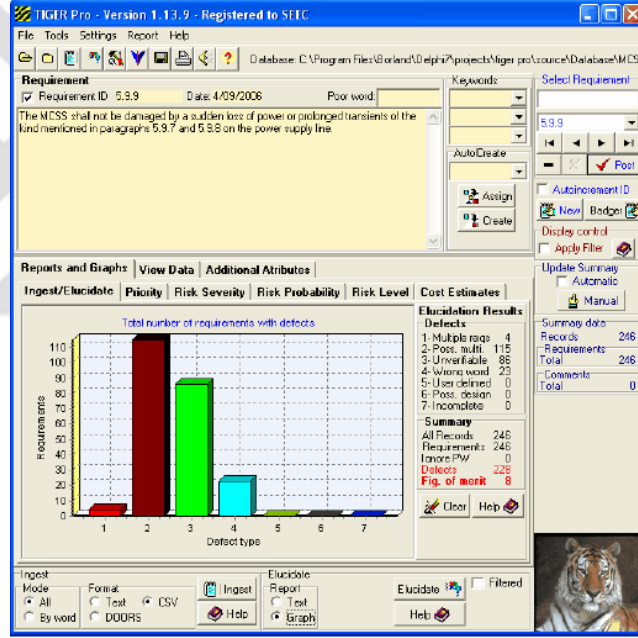
**Tablo 2.6:** Gereksinim dokümanlarında bulunan hata oranları [11]

Doküman	Toplam Hata	Sözlüksel	Sözdizimsel	Sentaks
1	61	%21	%53	%26
2	86	%15	%53	%32
3	60	%15	%11	%74
4	92	%23	%35	%42

Yazılımın için gerçekleştirilen deneylerde gerçekten var olan hataların kaçta kaçının otomatik olarak yakalandığına dair bir veri sunulmadığı için performans anlamında herhangi bir yorum yapılamamaktadır.

## 2.4.6 Tiger Pro

Josph Kasser tarafında 2006 yılında ortaya konulan Tiger Pro yazılımının temel amacı hatasız yazılım gereksinimlerine ulaşırken, gereksinim yazma ve gözden geçirme zorluklarının da belirgin şekilde ortaya çıkarmaktır [29]. İlk olarak eğitim amaçlı geliştirilen yazılım daha sonra yüksek kaliteli gereksinimler yazma hedefi ile literatürde kendine önemli bir yer edinmiştir. Tiger Pro'dan alınmış bir ekran görüntüsü Şekil 2.3 Tiger Pro yazılımı kullanıcı ara yüzünde verilmiştir.



Şekil 2.3: Tiger Pro yazılımı kullanıcı ara yüzü

Yazılım temel olarak yedi gereksinim hatası üzerine yoğunlaşmıştır [29];

1. Bir satırda birden fazla gereksinim
2. Bir satırda muhtemel birden fazla gereksinim
3. Doğrulanamaz ve test edilemez ifadeler
4. Hatalı kelime kullanımı
5. Kullanıcının tanımladığı hatalı kelimeler
6. Muhtemel tasarım kısıtları
7. Tamamlanmamış ifadeler

## 2.4.7 Diğerleri

İngilizce gereksinimler üzerine analizler yapan yazılımlardan öne çıkanlarının haricinde literatürde yerini almış ve benzer yaklaşımlar ile ortaya konulan farklı çalışmalar da bulunmaktadır. Ortaya konulan bu çalışmaların bazıları sektörel ve akademik anlamda ismini duyursa, birçoğu tam otomatik kontroller sunamaması, kullanıcıdan analiz ve kontroller için destek alması, kısıtlı ve sadece belirli analizleri yapabilmeleri sebebi ile doğal dil işleme ile otomatik gereksinim analizi konusunda fazla ön plana çıkamamışlardır. Bu yazılımlardan bazılarını Tablo 2.7’de sıralanmıştır.

**Tablo 2.7:** İngilizce gereksinim analiz yazılımları [27]

Araç	Hedef	Tip	Yıl
Circe	Gereksinim Kalitesi	Yarı Otomatik	2006
NAI	Muğlaklık Analizi	Otomatik	2010
CRF Tool	Belirsizlik Analizi	Otomatik	2012
T1'	Muğlaklık Analizi	Otomatik	2008
RAT	Gereksinim Kalitesi	Otomatik	2009
MaramaAI	Gereksinim Kalitesi	Yarı Otomatik	2011
EuRailCheck	Gereksinim Kalitesi	Yarı Otomatik	2012
DODT	Gereksinim Kalitesi	Yarı Otomatik	2011
SREE	Muğlaklık Analizi	Yarı Otomatik	2013
Extraction of OLAP req.	Gereksinim Kalitesi	Otomatik	2009
HEJF	Gereksinim Kalitesi	Otomatik	2014
Dowser	Muğlaklık Analizi	Yarı Otomatik	2008
Anaphora detection	Muğlaklık Analizi	Otomatik	2011

Bir diğer yazılım ise Münih Teknik Üniversitesi tarafından başlatılan ve devamında ticari bir ürüne dönüşen **Qualicen** yazılımıdır. Gereksinim analizlerinde “*smell*” olarak ifade ettiği potansiyel hata olabilecek bulguların peşine düşmektedir. Bu potansiyel gereksinim hatalarını sıralamak gerekirse [27];

1. Muğlak zar ve sıfatlar
2. Negatif kelimeler
3. Doğrulanamaz terimler
4. Sübjektif dil
5. Kesin olmayan ifade
6. Karşılaştırmalı gereksinimler
7. Belirsiz zamir
8. Boşluk analizi,

9. Kullanıcı ara yüz detay
10. Uzun cümle kullanımı

Ticari olamayan yazılımlar içerisinde hataya ve yanlış anlaşılmalara açık zayıf metin analizleri yapan yaklaşımlardan diğerlerine göre bir adım önde olarak SREE yazılım değerlendirilebilir. Kalite analizlerinde kullanma üzere on ayrı kalite göstergesi belirlemiştir;

1. Devamlılık
2. Koordinatör
3. Yönlendirici
4. Tamamlanmamış
5. Opsiyonel
6. Çoğulcu
7. Zamir
8. Niteliksel
9. Muğlak
10. Zayıf

SREE'nin başarısını göstermek amacı ile yeni bir havalimanı proje gereksinimleri üzerinde başarı ve tutarlılık oranı %68 olarak ölçülmüştür [30].

## **2.5 Türkçe Gereksinimler için Otomatik Gözden Geçirme Yaklaşımları**

### **2.5.1 İngilizce analiz yaklaşımlarının Türkçe gereksinimlere uygulanması**

Halil Kızılcıca ve Egemen Yılmaz tarafından 2008 yılında ortaya konulan çalışmada temel olarak literatürde var olan İngilizce gereksinim analiz yazılım ve yaklaşımları incelenerek, belirli bir analiz kümesi oluşturulmuş ve bunun üzerinden bir deneme yazılımı ortaya konulmuştur. Bu yazılım hem İngilizce hem de Türkçe gereksinimler üzerinden deneyerek sonuçlar karşılaştırılmıştır [31].

Çalışmada, İngilizce gereksinim analizleri yapan RAST, QUARS, Tiger Pro ve ARM yazılımları ana hatları ile incelenmiş ve bu yazılımlarda ki yaklaşımlar H. Kızılcıca ve E. Yılmaz tarafında ortaya konulan yazılıma temel oluşturulmuştur [31].

Gereksinimlerin analiz edilerek hataların ayrıştırılması ve gereksinimlerinin kalitesinin artırılması sürecini öncelikle bu analizleri zorlaştıran faktörlerin belirleyerek başlatmışlar ve bu sürecin tamamına yine var olan İngilizce çalışmalar



üzerinden yön vermişlerdir. Çalışmalarında temel olarak gereksinimleri zorlaştıran faktörler üç ana grup altında toplamışlardır;

- a. Okunabilirlik, tutarlılık ve bütünlük ile ilgili dilbilimsel zorluklar
- b. İşlevsel ve işlevsel olmayan gereksinim tip ayırımı ve önceliklendirme zorluğu
- c. Görselleştirme zorlukları

İlk grupta belirtilen okunabilirliğe etkisi olan dilbilimsel güçlüklerin belirlenmesi adına ARM ve QUARS yazılımlarında tanımlanan hata türleri temel alınarak Tablo 2.8'deki sınıflandırmalar yapılmıştır;

**Tablo 2.8:** Hata tipleri ve analiz yöntemleri [31]

Hata tipi	Analiz Yöntemi
Seçeneklilik	Sözlüksel
Öznellik	Sözlüksel
Belirsizlik	Sözlüksel
Üstü Kapalılık	Sözdizimsel
Zayıflık	Sözdizimsel
Eksiklik	Sözdizimsel
Çokluk	Sözdizimsel
Tanım Eksiği	Sözdizimsel
Atıf Eksiği	Sözlüksel

Araştırmacılar yukarıda çıkardıkları hata tiplerine ek olarak, gereksinim dokümanının okunabilirliği, doküman bütünlük ve tutarlılık analizleri, işlevsel olan ve olmayan gereksinim ayırımı, görselleştirilen gereksinim hataları gibi konularda literatürde var olan İngilizce dil işleme çalışmalarını incelemiş ve bunlar üzerinden kendilerine hedef olarak aşağıdaki fonksiyonları belirlemişlerdir.

- 1- Tablo 2.8'de verilen hataların tamamen ayıklanması
- 2- Gereksinimlere özel ve tüm doküman genelinde kalite ölçüm istatistik raporlamaları
- 3- Verilmiş tüm metinden gereksinim ifadeleri çekme
- 4- İşlevsel ve işlevsel olmayan analizi
- 5- Bütünlük, tutarlılık analizi
- 6- Gereksinimleri XML tabanlı bir formata çevirerek görselleştirme

Çalışma sonunda hedeflenen bu altı ayrı analiz için pratikte geliştirme, deney olarak sadece ilk madde İngilizce gereksinimler üzerinde denenmiştir. Türkçe için hedeflenen çalışmaların henüz tamamlanmadığı, bu analizlerin Türkçe doğal dilde yapılabilmesinin zorlukları aşağıdaki gibi özetlenerek çalışma tamamlanmıştır.

- a. Türkçe'nin eklemeli bir dil olması,
- b. Sert ünsüzlerin yumuşaması, ünsüz benzeşmesi gibi kuralların kök ve ekleri değiştirmesi
- c. Vurgu yapmak için devrik cümle kullanımının İngilizceye göre sık kullanılması
- d. Türkçe'de kiplerin ek olarak gelmesi, İngilizce gibi ayrı kelime olmaması
- e. Türkçe ifadelerde kelime ve hece sayısının fazlalığı
- f. Türkçe okunabilirlik üzerine literatürdeki çalışmaların azlığı

### **2.5.2 Dilbilimsel analizler ile Türkçe gereksinimlerin kalitesinin arttırılması**

İ. Berk Yılmaz tarafından 2010 yılında ortaya konulan çalışmada literatürde var olan İngilizce ve Türkçe doğal dil işleme ve gereksinim analiz yaklaşımları incelenmiş ve üç aşamalı bir gereksinim analiz metodu oluşturularak bir yazılım geliştirilmiştir [32].

#### **1. Muğlaklık hata tipleri ve analizi**

Muğlaklık hata tipleri analizi için Tablo 2.8 Hata tipleri ve analiz yöntemlerinde verilen Yılmaz ve Kızılcıca'nın çalışma ve yöntemleri birebir temel alınarak kullanılmıştır. Bu çalışmada ortaya konulan 8 hata tipinin zayıflık harici diğer 7 hata tipi yakalanmak üzere analizler yapılmıştır [32]. Belirlenen hata tipleri için oluşturulan sözlük üzerinde sözcüksel analizler yapılarak sonuçlar raporlanmıştır.

#### **2. İşlevsellik Analizi**

Ortaya konulan yazılımın ikinci temel fonksiyonu ise işlevsel olan ve olmayan yazılım gereksinimlerinin ayrılması analizidir. Bu analizde Tablo 2.1 ISO/IEC 9126'ya göre yazılım kalite niteliklerinde listelenen maddeler işlevsel olmayan gereksinim analiz kriterleri olarak tanımlanmıştır. Bu kriterlerin işlevsel olmayan gereksinim analiz maddeleri olarak

tanımlanmasının doğruluğu üzerine sonuç ve karşılaştırma bölümlerinde yorumlar getirilecektir.

- i. İşlevsellik
- ii. Güvenilirlik
- iii. Kullanılabilirlik
- iv. Verimlilik
- v. İdame ettirilebilirlik
- vi. Taşınabilirlik

### 3. Benzerlik analizi

Ortaya konulan çalışmanın ana fonksiyonlarında sonuncusu gereksinimler analiz edilerek aralarındaki benzerliğin ölçülmesidir. Literatürde ortaya konulan dilbilimsel ve istatistiksel yaklaşımların özetlendiği çalışma, dilbilimsel analizler kadar başarılı olamasa da uygulama kolaylığından dolayı istatistiksel yaklaşımları temel almıştır. Algoritma olarak Natt och Dag vd. (2001) sunduğu otomatik gereksinim benzerlik analizi kullanılmış ve belirli ön analizler sonucunda Dice, Jacard ve Cosine benzerlik katsayıları yardımı ile istatistiksel sonuçlara varılmıştır [32].

Çalışma sonucunda gerçekleştirilen deneylerde 203 adet gereksinim kullanılmış ve bunlar üzerinde yukarıda üç ana grupta belirtilen analizler yapılmıştır. Analizler sonucunda muğlaklık analizlerini içeren 7 hata tipi sorgulamasında, toplam 72 hatadan 58 adeti yakalanmış, kaçırılan hatalara ek olarak bir gereksinim de hatasız olmasına rağmen hatalı gösterilmiştir. Muğlaklık analizi özelinde, Yılmaz ve Kızılca'nın yaklaşımını kullanan yazılım, deney sonuçları açısından başarılı görünse de, gereksinim sayısı ve sadece yedi hata tipi araması sebebi ile dar kapsamlı bir analiz yaptığı söylenebilir.

İkinci analiz olan işlevsel olmayan gereksinim analizinde ise, var olan 43 hatada 22'sinin yakalandığı raporlanmıştır. Hata olmayan 5 gereksinim ise hatalı olarak raporlanmıştır. Bu analizin başarısı üzerine yorum yapmaktan ziyade başlangıçta bu analizin doğru tanımlanmadığı ve bir kavram karmaşası yaşandığı söylenebilir. İşlevsel olmayan gereksinim kriterleri olarak belirlenen ISO/IEC 9126 maddelerinin tamamı aslında işlevsel gereksinimler dâhil tüm yazılım gereksinimlerinin kalite kriterlerini tanımlamaktadır ve standart dokümanstasyonu içerisinde ve literatürde ki diğer tüm standart ve çalışmalarda bu netçe ifade

edilmektedir. İşlevsel olmayan gereksinim aslında hatalı bir gereksinim değil, sadece ürünün fonksiyonel özellikleri yerine, performans, dayanıklılık ve zamanlama gibi işlevsel olmayan detaylarını tanımlayan gereksinimlerdir.

Benzerlik analizlerinin yapıldığı üçüncü analizde gereksinimler çiftler halinde gruplanarak istatistiksel analizler ile karşılaştırılmış ve 0 ile 1 arasında bir benzerlik ölçüm sonucu sunulmuştur [32]. Bu analizler sonucunda 57 gereksinimin birbirine benzer olduğu yorumlanmış fakat dilbilimsel analizler yapılmadığı için bu analizin yeterli seviyede başarılı olmadığı ifade edilmiştir [32].



## ÜÇÜNCÜ BÖLÜM

### ÖNERİLEN YÖNTEM

Gereksinim gözden geçirme aktivitelerinin temel amacı, hataların yazılım geliştirme yaşam döngüsünün devam eden süreçleri başlamadan en erken safhada yani gereksinimler hazırlanırken ortadan kaldırılmasıdır. Fakat projelerdeki yoğunluk, gözden geçirme sürecine dâhil olan paydaşların tecrübe eksikliği veya farklı sebepler gözden geçirme aktivitelerine gerekli zaman ve önemin ayrılamamasına ve sürecin başarıyla gerçekleştirilmesine engel olmaktadır. Bu sebeple gereksinim gözden geçirme sürecinin otomatik olarak araçlar yardımı ile gerçekleştirilebilmesi ve insan faktörünün minimuma indirilmesi projelere hem maliyet, hem de ileriki aşamalarda ortaya çıkacak potansiyel hataların en erken safhada ortadan kaldırılmasını sağlayarak etkisi sayısal olarak hesaplanamayan dolaylı maliyetlerinde elimine edilmesini sağlayacaktır.

Bu kapsamda doğal dilde yazılmış Türkçe gereksinimlerin otomatik olarak gözden geçirilebilmesi için tez içerisinde ortaya konulan yaklaşım Şekil 3.1'de özetlenen temel dört yapıtaşından oluşmaktadır. Otomatik gözden geçirme sürecine temel girdi olan Türkçe gereksinim dokümanları, en popüler gereksinim yönetim araçlarından kolayca XML dosya formatı ile dışarı aktarılabilen veri setleridir. Bu veriler gözden geçirme yazılımı tarafından çözümlenerek bir modelde tutulur ve üzerinde gözden geçirme algoritmaları çalıştırılmaktadır. Bu algoritmaları ve diğer bir deyiş ile gereksinimlerini tanımlayan yapı ise otomatik gözden geçirme kontrol listesidir (OGGKL). Yaklaşımın son parçası ve temel çıktısı, yazılım tarafından otomatik olarak üretilen ve resmi gözden geçirme delil ve kaydı olarak sunulabilecek sonuç raporlarıdır.



Şekil 3.1: Otomatik gereksinim gözden geçirme yaklaşımı

### 3.1 Veri Seti

#### 3.1.1 Gereksinimler

Ortaya konulan OGGKL için geliştirilen algoritmaların eksiksiz test edilmesi ve gözden geçirme performansının objektif bir şekilde ölçülebilmesi deneylerde kullanılan veri seti ve bu setin çeşitliliği ile doğru orantılıdır. Tez için geliştirilen otomatik gözden geçirme yazılımına girdi sağlayacak Türkçe doğal dilde yazılmış gereksinimlerin baştan tez için özel olarak hazırlanması ve belirli hataların gereksinimlere yerleştirilmesi yaklaşımı yapılacak olan deneylerin ve sonucunda ortaya konulacak olan ölçüm ve değerlendirmelerin suni olmasına sebep olacaktır. Diğer taraftan sektörde devam etmekte olan projelerden henüz gözden geçilmemiş, hatalar içeren gereksinim setlerinin yeterli sayı ve çeşitlilikte temin edilmesinin kurumsal süreç ve proje gizliliği gibi faktörler de düşünüldüğünde ne kadar zor olduğu ortadadır. Bu kapsamda deneylerin objektif ve gerçeğe en yakın sonuçlar verebilmesi adına veri setlerinin oluşturulma süreci aşağıda özetlenmiştir.

1. Oluşturulan OGGKL'nin uygulanabilirlik ve geçerliliğini test etmek amacı ile olabildiğince farklı sektörlerdeki projelerden Ek E'de sunulan 526 adet gereksinim kullanılmıştır.
2. Farklı sektör ve firmalardan toplanan bu gereksinimler farklı kurumsal süreç ve yöntemlere göre son şekillerini almışlardır.

3. İngilizce doğal dilde yazılmış olan bu gereksinimler, makine çeviriciler yardımı ile Türkçe 'ye çevrilerek tezde veri seti olarak kullanılmışlardır. Türkçe'ye çevrilen bu gereksinimlerin anlamsal olarak bozulmadığı ve çeviri kaynaklı hataların eklenmediği ayrıca gözden geçirilerek kontrol edilmiştir. Diğer yandan bu gereksinimlerin olgun olması ve pratikte çok fazla hata içeren ve okunabilirliği düşük dokümanların gözden geçirmeye sunulduğu gerçeği düşünüldüğünde, bu çevirilerde gözden kaçabilecek muhtemel bozulmaların da aslında veri setine gerçekçilik anlamında zenginlik katacağı değerlendirilebilir.
4. Daha önce tamamlanmış projelerden elde edilecek gereksinimlerin proje ekipleri tarafından gözden geçirilmiş ve üzerinden geliştirme ve doğrulama aktivelerinin tamamlanmış olması deneyler için olgun ve zorlayıcı veri setleri anlamına gelmektedir. Bu durum tezde sunulan yaklaşım ve algoritmalar için yapılacak olan testler sonucunda elde edilen sonuç ve değerlendirmeler için güvenilirlik ve başarı anlamında olumlu yansımaları olacaktır. Ayrıca bulunan her bir hatanın, proje ekiplerinin gözden kaçırdığı hataların otomatik olarak bulunabildiği anlamına gelecek bu da tez performans ölçümleri açısından önem arz etmektedir.

### **3.1.2 Kullanılan Gereksinim Dokümanları**

Tez kapsamında deneysel çalışmalarda veri seti olarak kullanılmak üzere farklı sektörlerden edinilmiş ve Ek-F'de sunulan gereksinim dokümanlarını özetle tanımlamak gerekirse;

1. Elektronik ve Ev Eğlence Sistemleri geliştiren ve satışı yapan bir firma için internet üzerinden satış yapmak amacı ile geliştirilen 111 adet yazılım gereksinim nesnesinden oluşan dokümanın orijinali İngilizce olarak hazırlanmıştır. Fonksiyonel, kullanılabilirlik, emniyet, performans, güvenilirlik ve uygunluk, bakım ve destek ve tasarım ve ara yüz gereksinimleri gibi ayrı bölümler altında hazırlanan tüm gereksinimler birleştirilerek E-Market gereksinim seti oluşturulmuştur.
2. Amerikan Federal Havacılık İdaresi tarafından hazırlanan Gereksinim Mühendisliği Yönetimi El Kitabı'ndan alınan ve aviyonik sistemler için geliştirilen 51 adet gereksinimden oluşmaktadır.

3. Hava yolu yolcu rezervasyon sistemi için hazırlanan ve 77 adet gereksinim nesnesinden oluşan gereksinim dokümanı kullanılmıştır.
4. Çocuklar için eğlenirken öğrenmelerini sağlamak amacı ile geliştirilen Eğitimsel Oyun Projesi gereksinim dokümanından elde edilen 36 adet gereksinim doküman nesnesinden oluşan gereksinim grubu kullanılmıştır.
5. Proje yönetimini kolaylaştırmak amacı ile proje paydaş rollerini temel alarak geliştirilen ve Versiyon Kontrol Sistemi, Değişiklik ve Hata Takip Sistemi gibi birçok sistem ile ara yüzü bulunan proje için geliştirilen toplam 120 adet gereksinim doküman nesnesi kullanılmıştır.
6. Günümüzün modern dokunmatik teknolojilerinin geleneksel gazete ve mecmualar için uygulanmasını hedefleyen Dokunmatik Gazete Projesi'ne ait toplam 93 adet gereksinim doküman nesnesinden oluşmaktadır.

### **3.1.3 Veri seti gereksinimlerin dosya formatı ve yazım standardı**

Gereksinim yönetim araçlarında ve Ek B- Gereksinim Yönetim Araç Karşılaştırma Kriterleri'nde bahsettiğimiz gibi birçok araç gereksinimlerin farklı dosya formatlarında dışa aktarılmasına olanak sağlamaktadır. Microsoft Office araçlarına uygun dosya tiplerinde (.xlsx, .docx) dışarı aktarabildiğiniz gibi pdf vs gibi popüler dosya formatları kullanılabilir, ayrıca rtf, txt, xml, csv gibi otomatik analize daha uygun standart dosyalar da oluşturabilirsiniz. Tezin temel veri setini oluşturan tüm gereksinimlerde en çok kullanılan formatlardan olan Microsoft Excel üzerinde standart XML elektronik tablosu olarak oluşturulmuştur. Bu sayede doküman Excel üzerinde okunabilirliği sağlanırken, XML formatı farklı script veya araçlar için kolay çözülebilirlik sağlamaktadır. Oluşturulan bu dosyalar, tez ortaya konulan gözden geçirme aracına herhangi bir güncelleme veya düzeltme ihtiyacı olmaksızın girdi olarak verilebilmektedir. Gereksinim dokümanlarında projelerden projeye fark edebilecek birçok alt alan ve veri tipi kullanılabilir. Fakat farklı sektörlerde de olsa gereksinimler için en kritik ve ortak sayılabilecek 4-5 adet doküman objesi yani veri tipi bulunmaktadır. Bunlar gereksinim numarası, gereksinim metni, gereksinim tipi ve izlenebilirlik. Tezde kullanılan veri setleri de bu 4 gereksinim özelliğini içermekte ve yazılım bu alanlar üzerinden analizler yapmaktadır. Tezde kullanılan bir gereksinim dokümanının hem Microsoft Excel'de hem de bir metin düzenleyicisindeki görüntüsü Şekil 3.2 XML formatlı Türkçe gereksinim seti ile özetlenmiştir.



## XML Excel Görünüm

	A	B	C	D
1	Gereksinim No	Gereksinim	Tip	İzlenebilirlik
2	ARS_PRJ_GRK_1234	ARS, Katıplığın / Yöneticinin çalışma	GEREK SINİM	ÜSG_ARS_GRK_4323
3	ARS_PRJ_GRK_1235	ARS ayrıca, genel müşteriler için World	GEREK SINİM	ÜSG_ARS_GRK_4324
4	ARS_PRJ_GRK_1236	Yukarıdaki iki ARS arabirimi kullanıcılar a aşağıdaki işlevleri	GEREK SINİM	ÜSG_ARS_GRK_4325
5	ARS_PRJ_GRK_1237	ARS ayrıca, müşteriler tarafından telefon veya cep telefonu aracılığıyla her yerden erişilebilen kullanımı	GEREK SINİM	ÜSG_ARS_GRK_4326

```
<Row>
<Cell><Data ss:Type="String">ARS_PRJ_GRK_1235</Data></Cell>
<Cell><Data ss:Type="String">ARS ayrıca, genel müşteriler için World Wide Web'de interaktif bir GUI s
<Cell><Data ss:Type="String">GEREK SINİM</Data></Cell>
<Cell><Data ss:Type="String">ÜSG_ARS_GRK_4324</Data></Cell>
</Row>
<Row ss:Height="30">
<Cell><Data ss:Type="String">ARS_PRJ_GRK_1236</Data></Cell>
<Cell><Data ss:Type="String">Yukarıdaki iki ARS arabirimi kullanıcılar a aşağıdaki işlevleri sağlamaya
<Cell><Data ss:Type="String">GEREK SINİM</Data></Cell>
<Cell><Data ss:Type="String">ÜSG_ARS_GRK_4325</Data></Cell>
...
```

## XML Verisi

Şekil 3.2: XML formatlı Türkçe gereksinim seti

Gereksinim dokümanlarının analizlerinde gereksinim tipi en önemli girdilerden birini oluşturmaktadır. Uygulanan standart ve süreçlere göre değişiklik gösterse de, tez kapsamında DO-178B standardına uygun bir şekilde yazılım gereksinim tipleri kullanılmıştır:

1. Gereksinim: Tanımlı kısıt ve girdiler kapsamında yazılımın üretmesi gereken çıktıları tanımlar. İzlenebilirlikleri olan ve alt veya üst seviye gereksinim olarak tanımlanabilirler [20].
2. Türetilmiş Gereksinim: Üst seviye gereksinimlere izlenebilirlikleri çekilemeyen, yazılım tasarım ve geliştirme sürecinde ortaya çıkmış ek gereksinimlerdir [20].
3. Açıklama: Gereksinim veya türetilmiş gereksinim olmayan diğer tüm bilgilendirici yorum, açıklama, şekil vb. benzeri doküman nesnelere.

### 3.2 Otomatik Türkçe Gereksinim Gözden Geçirme Kontrol Listesi

Gözden geçirme otomasyon süreci için ortaya konulacak teknik altyapı ve çözümlerin hayata geçirilmesi ne kadar önemli ise gözden geçirme algoritmalarına temel olan gözden geçirme kontrol (OGGKL) listesinin de ne kadar kapsamlı, doğru ve standartlara uygun bir şekilde oluşturması da bir o kadar önem arz etmektedir.

Sondan eklemeli bir dil olan Türkçe 'de gözden geçirme sürecinin otomasyonu İngilizce gibi birçok doğal dille kıyaslandığında teknik olarak çok daha zor olması, bu kontrol listenin muhteviyatı ve niteliğine ayrıca önem kazandırmaktadır. Bu amaçla literatürde var olan birçok standart ve çalışma incelenerek bugüne kadar ortaya konulmamış en kapsamlı Türkçe gereksinim gözden geçirme kontrol listesi oluşturulması hedeflenmiş ve ilgili maddeler Tablo 3.1'de sıralanmıştır.

**Tablo 3.1:** Otomatik Gözden Geçirme Kontrol Listesi

<b>OGGKL</b>	
<b>Okunabilir</b>	
<b>K1</b>	Gereksinim yazım dili Türkçedir.
<b>K2</b>	Gereksinimler dilbilgisi kurallarına uygun ve yazım hatalarından arınmış olmalıdır.
<b>Özgün</b>	
<b>K3</b>	Gereksinimler birbirlerinden ayırt edilebilen özgün bir tanımlayıcı numaraya sahip olmalıdır.
<b>Muğlak Olmayan</b>	
<b>K4</b>	Gereksinimlerde işlevi tanımlayan yüklem güçlü bir dil ile <i>-ecek, -acak</i> eklerini içererek tanımlanmalıdır.
<b>K5</b>	Gereksinimler pozitif bir dil ile yazılacak, negatif ifadeler içermeyecektir.
<b>K6</b>	Gereksinimlerde muğlak ifadeler kullanılmamalıdır.
<b>Çokluk</b>	
<b>K7</b>	Bir gereksinimde birden fazla gereksinimi içermemelidir.
<b>Kesinlik</b>	
<b>K8</b>	Gereksinimler okuyanı ikna edici ifadeler içermemelidir.
<b>K9</b>	Gereksinimlerde belirsiz ifadeler kullanılmamalıdır.
<b>K10</b>	Gereksinimlerde ekstra kesinlik belirten ifadeler kullanılmamalıdır.
<b>K11</b>	Gereksinimler genelleyici ifadeler içermemelidir.
<b>K12</b>	Gereksinimler göreceli, öznel ifadeler içermemelidir.
<b>K13</b>	Gereksinimler tavsiye edici ifadeler içermemelidir.
<b>Tamamlanmış</b>	
<b>K14</b>	Gereksinimlerdeki tüm liste ve maddeler eksiksiz olmalıdır.
<b>İzlenebilir</b>	
<b>K15</b>	Gereksinim seviyeleri arasında ileri, geri izlenebilirlikler kurulu olmalıdır.
<b>K16</b>	Türetilmiş gereksinimler için izlenebilirlik oluşturulmamalıdır.
<b>Standartlara Uygunluk</b>	
<b>K17</b>	Gereksinim numara yazım formatı projeye özel belirlenen standarda uygun olmalıdır.
<b>K18</b>	Gereksinim olmayan açıklamalara proje standardına uygun belirleyici numara verilmelidir.
<b>K19</b>	Gereksinim tipleri projede belirlenen tiplere uygun atanmalıdır.
<b>K20</b>	İzlenebilirlik çekilen üst seviye gereksinim dokümanı ve referans gereksinim numara yazım formatı belirlenen standarda uygun olmalıdır.

K1: Gereksinimlerin tüm paydaşlar tarafından okunup ve anlaşılabilmesi gereksinimlerin taşınması gereken en temel özelliktir. Bu amaçla proje plan ve

standartlarına göre başlangıçta kararı verilen gereksinim yazım dili, yazılım ve sistem projelerinde kullanılan İngilizce terimler, tanımlar, teknik ifade ve kısaltmalar sebebiyle genellikle İngilizce ya da İngilizce-Türkçe iç içe geçmiş bir yapıda kullanılmaktadır. Bu sebeple standart ve planlardan gelen dil kararına uyulması ve ilgili denetim otoritelerine de bu uyumun ispatlanması amacı ile K1 maddesi otomatik OGGKL'ne dâhil edilmiştir. Gereksinimler üzerinde benzer bir otomatik dil kontrolüne literatürdeki başka herhangi bir çalışmada rastlanmamıştır.

K2: Okunabilirliği azaltan diğer durumlar, yazım ve dilbilgisi hatalarıdır. Yazım ve dilbilgisi hatalarının yoğunlukta olduğu, dar zamanda hızlı bir şekilde hazırlanmış, okunabilirliği düşük gereksinim dokümanları tüm proje süreçleri için olduğu gibi, manuel veya otomatik gerçekleştirilen gözden geçirme süreçleri için de sorun teşkil etmektedir. Proje ekibi tarafından gerçekleştirilen manuel gözden geçirme sürecinde zaman kaybı ve yanlış anlaşılmalara sebebiyet verebildiği gibi, otomatik gözden geçirme sürecinde de gereksinim analizlerinin doğru bir şekilde gerçekleştirilememesine neden olabilmektedir. Basit bir yazım hatası sebebi ile bozulan kelime yapısı, OGGKL ile uyumsuz kullanılan ifadelerin gözden kaçırılmasına sebep olmaktadır. Bu amaçla otomatik OGGKL'ye dâhil edilen K2 kontrol maddesi okunabilirliği düşük olan gereksinim dokümanlarının öncelikle otomatik olarak düzeltilmesini hedeflemektedir. Bu kontrol maddesi diğer kontrol maddeleri çalıştırılmadan önce bir ön gözden geçirme gibi dokümandaki dilbilgisi ve yazım hatalarını önden düzelterek diğer kontrol maddeleri analizleri için dokümanı daha uygun bir hale getirmektedir.

K3: Gereksinimlerin izlenebilirliklerini sağlayan numaralarının gereksinimleri belirleyici şekilde özgün tanımlanması gerekmektedir. Gereksinim yönetim araçları üzerinde gereksinimler oluşturuluyorsa bu araçlar her bir doküman nesnesine ayrı bir numara verebiliyor. Fakat metin düzenleyiciler ve ofis programları üzerinde gereksinimler yazılıyorsa çok fazla gereksinim içeren, güncellemelerin fazla olduğu ve eklenen ve çıkarılan gereksinimlerin fazlalığı bu numaraların özgünlük takibini zorlaştırıyor. Ayrıca bazı gereksinim yönetim araçları gereksinim numaralarını kullanıcının tanımlanmasını bekleyebilir, aynı zamanda bu numaraların güncellenmesine izin verebilmektedir. Bu sebeple dokümanın son halinde özgün numaraya sahip olmayan gereksinim nesnelere için K3 maddesi kontrol listesine dâhil edilmiştir.

K4: Gereksinimlerdeki fonksiyonların hatasız bir şekilde geliştirilip doğrulanabilmesi ve okuyan tüm paydaşlar tarafından aynı şekilde anlaşılması amacı ile net ve güçlü bir dil ile ifade edilmesi gerekmektedir. Gereksinimde işlevi yüklem tanımlamakta, bu sebeple yanlış yorumlamalara veya eksik ya da fazladan bir işlev çıkarımına sebep olabilecek ifade kullanımları yüklem ile gerçekleştirilebilir. Bu tip hataların engellenmesi amacı ile güçlü ve net dil kullanımı için K4 maddesi ile yüklem analizi kontrol listesine dâhil edilmiştir.

K5: Gereksinimler yazılımın veya sistemin gerçekleştirdiği fonksiyonları tanımlarlar. Sistem tarafından gerçekleştirilmeyen işlemler gereksinim değildir, bu sebeple gereksinimlerde negatif ifadeler de kullanılamaz. Fakat projelerde ekstra açıklama, istenmeyen bir durumun yazılım tarafından engellenmesi veya ekstra koruma amaçlı bu ifadelerle sıklıkla rastlanmaktadır. Bu tip gereksinimlerin tanımlanması, geliştirme ekibi tarafından yazılıma ve sisteme yansıtılması yazılım ve sistem performansına negatif etkilere sebep olabileceği gibi, beklenmedik hatalara da yol açabilirler. Ayrıca bu tip gereksinimlerin tanımlanması gereksiz tasarım, geliştirme ve doğrulama eforuna neden olmaktadır. Bu amaçla negatif ifade analizi için K5 maddesi kontrol listesine dâhil edilmiştir.

K6: Tasarım, geliştirme ve doğrulama aktivitelerinin ana girdisi olan gereksinimlerde muğlak ifadelerin kullanılmasını yasaklamaktadır. Bu tip net olmayan, farklı yorumlamalara ve yanlış anlaşılmalara sebep olabilecek, sistemi istenmeyen bir tasarıma yönlendirebilecek ve doğrulama faaliyetleri açısından test edilebilmeyen her türlü muğlak ifadenin gereksinimlerden ayıklanması hedeflenmektedir.

K7: Gereksinimler belirli koşullar altında belirli bir anda gerçekleştirilen fonksiyonları ifade etmektedir. Tek bir gereksinim aynı çıktıları üreten birden fazla koşul içerebilir, aynı zamanda aynı koşullara bağlı olarak birden fazla çıktı da üretebilir. Fakat aynı gereksinim içerisinde farklı koşullarda, farklı çıktılar üretilmesi birden fazla gereksinim içerildiğine işaret eder, kontrol listesini ihlal eder ve ayrı birer gereksinim olarak tanımlanmalıdırlar. Bu amaçla birden fazla gereksinimin tek bir gereksinim içerisinde ifade edilmesinin engelleyen K7 maddesi kontrol listesine eklenmiştir. Bu maddenin ihlalinin projelerde neden oluşturabileceği sorunlar aşağıda özetlenmiştir;

1. Gereksinim sayıları üzerinden tasarım geliştirme ve doğrulama planlarına, bütçe, işgücü ve benzeri proje metriklerine hatalı yansıtılması.
2. Geliştirme ve doğrulama faaliyetleri sırasında bazı fonksiyonların gözden kaçırılması,
3. Bir gereksinim içerisinde karışık ifade edilen birden fazla koşul ve çıktıların hatalara sebep olabilmesi,
4. İzlenebilirlik yönetimi zorlaşmakta ve eğer üst seviyede de birden fazla gereksinim içeren benzer bir yaklaşım farklı gereksinimler için uygulanmışsa izlenebilirlik takip edilemez bir hal almaktadır.

K8-K13: Gereksinimlerde belirtilen her fonksiyonun yazılım veya sistem tarafından kesin ve net bir şekilde karşılanması gerekmektedir. Tasarım ve geliştirme aktivitelerini zorlaştırabilecek, doğrulanması mümkün olmayan, okuyular tarafından farklı yorumlanabilecek her türlü ifade gereksinimlerin kesinlik özelliğini ihlal etmektedir. Bu amaçla K8 ile K13 arası tanımlanan kontrol maddeleri vasıtasıyla belirsiz, ikna edici, ekstra kesinlik belirten, genelleyici, göreceli, öznel ve tavsiye edici nitelikte ifadelerin ayıklanması hedeflenmiştir.

K14: Gereksinimlerin tamamlanmış olması yani tüm girdi, koşul, çıktı ve diğer tüm detayların eksiksiz bir şekilde tanımlanmaları gerekmektedir. Daha sonra tamamlanacak olması, okuyanın yorumuna veya tahminine bırakılması ya da bazı detayların bilindiği varsayılması uygun değildir. Bu amaçla gereksinimlerde eksik veya tamamlanmamış ifadelerin tespitine yönelik K14 kontrol maddesi kontrol listesine dâhil edilmiştir.

K15-K16: Projelerde son ürünün karşılaması gereken tüm fonksiyonlar kontratlardan başlayarak en üst seviye gereksinimlerden alt seviye yazılım gereksinimlerine kadar kırılarak aktarılmaktadır. Bu seviyeler proje, müşteri, denetim otoritesi, uyulan standart ve süreçlere bağlı olarak değişiklik gösterse de, minimumda gözden geçirmek istediğimiz gereksinim dokümanının bir üst seviye gereksinim veya kontrat gibi proje dokümanlarına izlenebilirliklerinin çekili olması beklenmektedir. Son ürünün beklentileri karşılaması ve eksik ya da hatalı bir fonksiyonunun geliştirilmediğinden emin olunması adına gereksinimler arası izlenebilirlikler takip edilmektedir. Eksik izlenebilirlikler hedeflenen üründen sapmalara, eksik fonksiyonlara ve istenmeyen ekstra fonksiyonların son ürüne girmesine sebep

olabilmektedir. Bu amaçla izlenebilirliklerin otomatik kontrolü amacı ile K15 ve K16 kontrol maddeleri OGGKL'ye eklenmiştir.

K17, K18, K20: Gereksinim dokümanlarında gereksinim değişikliklerinin ve izlenebilirliklerin takibi amacı ile her bir doküman nesnesi özgün birer numara verilmekte ve bu numaralar genellikle proje ismi, alt sistem modül adı veya fonksiyon ismi gibi ek belirteçler de taşımaktadır. Bu belirteçler sayesinde eksik izlenebilirlik olmadığı durumlarda da izlenebilirlikler ile ilgili hatalar ayıklanabilmektedir. Belirteçlerin ifade ettiği proje dokümanlarının kontrolü ile istenmeyen bir dokümandan devşirilen fonksiyonların son ürüne girmediğinden emin olunur. Özellikle benzer ürünlerin geliştirildiği fakat farklı versiyonların farklı müşterilere özel geliştirildiği projeler de muhtemel karışıklıkları önleme adına bu kontrol önem kazanmaktadır. Bu sebeple bu numaralar proje standartları ve süreçleri tarafından tanımlanmakta ve gözden geçirme sürecinde de standartlar ile uyumluluğu kontrol edilmektedir. Projelerdeki gereksinim dokümanları ve içerdikleri gereksinim sayıları düşünülerek bu doğrulamanın otomatik olarak yapılması amacı ile K17, K18 ve K20 numaralı kontrol maddeleri tanımlanmıştır.

K19: Gereksinim dokümanları farklı tipte gereksinimler ve gereksinimler ile birlikte açıklama, yorum, ek bilgi, tablo ve şekil gibi gereksinim olmayan bilgilendirici doküman nesnelere de içermektedir. Bu tiplerin doğru olarak tanımlanması tasarım, geliştirme ve doğrulama faaliyetlerinde gereksinimlerin atlanmasının, gereksinim olmayan nesnelere gereksinim olarak ele alınmasının ve izlenebilirliklerde oluşabilecek hataların önlenmesine katkı sağlamaktadır. Ayrıca izlenebilirlik çekilmemesi gereken türetilmiş gereksinimler de bu tip tanımları sayesinde ayırt edilebilmektedir. Bu amaçla proje standartlarına uygun belirlenen gereksinim tiplerinin doğru tanımlanmasının kontrol edilebilmesi adına K19 kontrol maddesi tanımlanmıştır.

### **3.3 Otomatik Gereksinim Gözden Geçirme**

Tez kapsamında Türkçe gereksinimlerin hazırlanan OGGKL'ye uygun olarak otomatik olarak gözden geçirilmesi amacı ile Java programlama dilinde bir yazılım geliştirilmiştir. Otomatik olarak gözden geçirme sürecine alt yapı sağlayan ve kontrol listelerine uygun algoritmaları sunan bu yazılımın geliştirme sürecinde Türkçe dil

yapısına bađlı olarak dođal dil iřleme yaklařımları detaylı incelenerek sũrece en uygun, gũncel ve bařarılı yaklařımlar deđerlendirilmiřtir.

### **3.3.1 Tez kapsamında Tũrkçe Dođal Dil İřleme**

#### **3.3.1.1 Tũrkçe Dil Yapısının Etkisi**

Tũrkçe yazılmıř gereksinimlerin otomatik olarak analiz edilerek gũzden geãirebilmesi ncelikle Tũrkçe dilinin szcũksel, biãim bilimsel ve szdizimsel zelliklerinin incelenerek algoritmalara temel olacak yaklařımların bařarı ile belirlenmesine bađlıdır. Ural Altay Dil ailesinden ve yapısal olarak bitiřken (*agglutinative*) bir dil olan Tũrkçe 'de szcũkler, kk szcũklere eklerin tespah taneleri gibi arkada eklenirken nlũ uyumu, nsũz deđermeesi ve nlũ nsũz dũřmesi gibi kurallara ile deđererek oluřurlar [22]. Szcũk oluřumuna etki eden ekim eklerin bolluđu, yapım eklerinin retkenliđi ve son eklerin szcũk oluřumuna direk etkisi Tũrkçe dil iřlemeyi zorlařtıran biãim bilimsel faktrlerdendir. Szcũk dizimi genel olarak zne-nesne-yũklem kuralı ile oluřsa da, vurgulama ve anlam bilimsel amaãlar dođrultusunda farklı farklı sz dizim kombinasyonlarına izin veren Tũrkçe'nin bu esnekliđi dil iřleme aãısından zor diller arasında yer almasını sađlamaktadır. Bu tũr zellikleri sebebi ile en geliřmiř ve yaygınlařmıř İngilizce gereksinim analiz yntemlerinin de Tũrkçe zerindeki bařarısızlıđının temel sebeplerini oluřurmaktadır.

#### **3.3.1.2 ITũ Tũrkçe Dođal Dil İřleme Yazılım Zinciri**

Tũrkçe dođal dil iřleme alıřma ve yaklařımları gũncellik ve kapsam anlamında incelendiđinde İstanbul Teknik niversitesi (İTũ) Dođal dil iřleme grubu tarafından ortaya konulan Tũrkçe Dođal Dil İřleme Yazılım Zinciri'nin belirgin řekilde n plana ıktıđı grũlmektedir. Geliřtirilen dođal dil iřleme algoritmaları ve bu algoritmaların gerãeklemede kullanılan veri tabanları, niversite bũnyesinde gerãekleřtirilen tũm arařtırmalara temel olduđu gibi, diđer niversite ve sektrlerde yapılan alıřma ve uygulamalara destek anlamında paylařıma aãılmıřtır. Ortaya konulan dođal dil iřlem zinciri analiz safhaları bakımında ařađıdaki řekilde sıralanmıřtır [33].

n iřleme:

- i. Blũtleme: kelime ve noktalama iřaretlerine kadar parãalama ayırma

ii. Aksan karakter düzeltmesi: ASCII karakterlerin muhtemel Türkçe karşılığına çevirme ((CRF), Şartlı rastgele alanlar metodu)

iii. Sesli harf tamamlayıcısı

iv. Yazım hata düzelticisi

v. Metin normalizasyonu: (önceki aşamaların birleşimi)

Biçim bilimsel işleme:

i. Biçim bilimsel çözümleme: sözcük yapısal analiz

ii. Biçim bilimsel çözüm netleştirme: sözcük için birden fazla analiz sonucundan kontekste bağlı tek sonuç çıkarma (et: isim mi? fil m?)

iii. Dil Türkçe mi değil mi karar vericisi

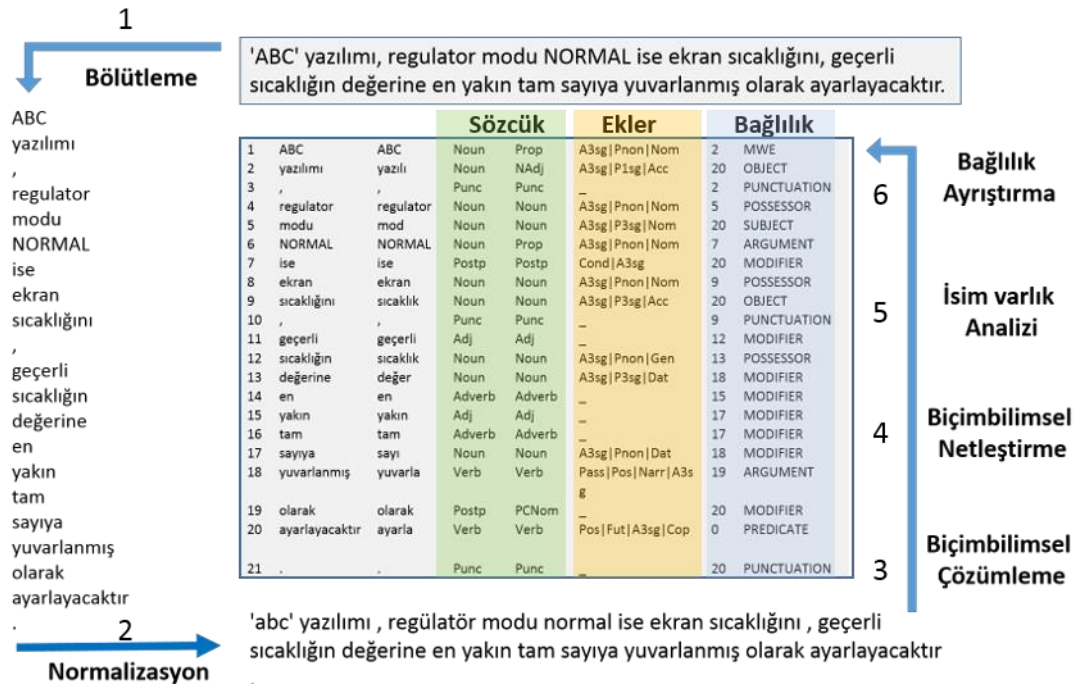
Birden çok sözcüklü ifade işleme:

i. İsim-varlık analizi (kişi, unvan, şehir vs.)

Sözdizimsel işleme:

i. Bağlılık Ayırıştırma (rahat et: iki kelime arasında bağlılık analizi)

Zincirin belirli safhalarının bileşiminden oluşan formal dil zincir algoritması Şekil 3.3’de bir gereksinim örneği üzerinde çalıştırılmıştır. Bölütleme fonksiyonu ile parçalara ayrılan cümle, normalizasyon ile yazım hataları düzeltilmiş devamında biçimbilimsel, ve sözdizimsel işleme adımları ile kelimelerin sözcük sınıfı, kökleri, ve sırası ile aldıkları ekler belirlenerek son halini almıştır.



Şekil 3.3: Formal dil zincir algoritma safhaları

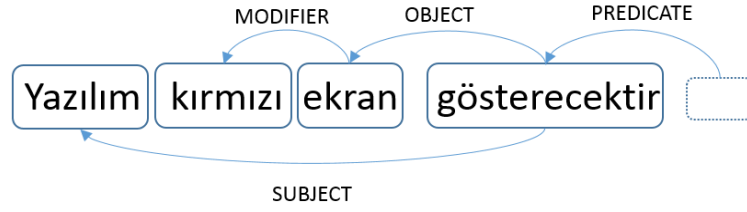


İTÜ formal dil zincir algoritması çıktılarından sözcük sınıfları, ek analizleri ve bağıllık ayrıştırma sonuçları tez kapsamında geliştirilen algoritmalarda analiz verisi olarak kullanılmaktadır. Bağıllık ayrıştırması sözcüklerin dizilişi ile ilgili bilgilerin cümle öğeleri arasındaki yönlü ikili bağıllıklarının ifadesidir [36]. Bağıllık ayrıştırma sonuçları öğeler arasındaki ilişkiyi açıklarken, bağıllık etiketleri bir asil öge ve onu niteleyen ikinci bir öge arasında oluşturulmaktadır. Tablo 3.2’de örneklenen bu bağıllık bilgileri gereksinimler için koşul çıktı analizlerinde kullanılan önemli verilerden birisidir.

**Tablo 3.2:** Sözdizimsel bağıllık etiketleri [36]

PREDICATE	Yüklem bağıllığı
OBJECT	Nesne bağıllığı
SUBJECT	Özne bağıllığı
DETERMINER	Belirtici: (bu, şu, o, bazı, bir, biraz, birçok, birkaç...)
POSSESSOR	Sahiplik (A’nı B’si)
MODIFIER	Diğer sıfat ve zarflar
ARGUMENT	başka, bağlı, beri, dair, doğru, değin, dek, denli, diye, dolayı, evvel...
INTENSIFIER	Pekiştirici: de, daha, bile, dahi, de, ise, ki, ya...
CONJUNCTION	Bağlaç: ama, ancak, çünkü, demek ki, fakat, halbuki, hatta...
COORDINATION	Birbirine göre ayarlama
RELATIVIZER	İlgi zamiri bağı: ki, de, mi..
APPOSITION	Yaklaştırma
PUNCTUATION	Noktalama
VOCATIVE	Seslenme: ah, of, ya, abi, canım, efendim, evet, hayır...
DERIV	Yapım ekleri ile türetilme bağıntısı
MWE	Birden fazla sözcük bağıntısı: kahkahayı bastı, Yeni İzlanda, blöf yapmak...

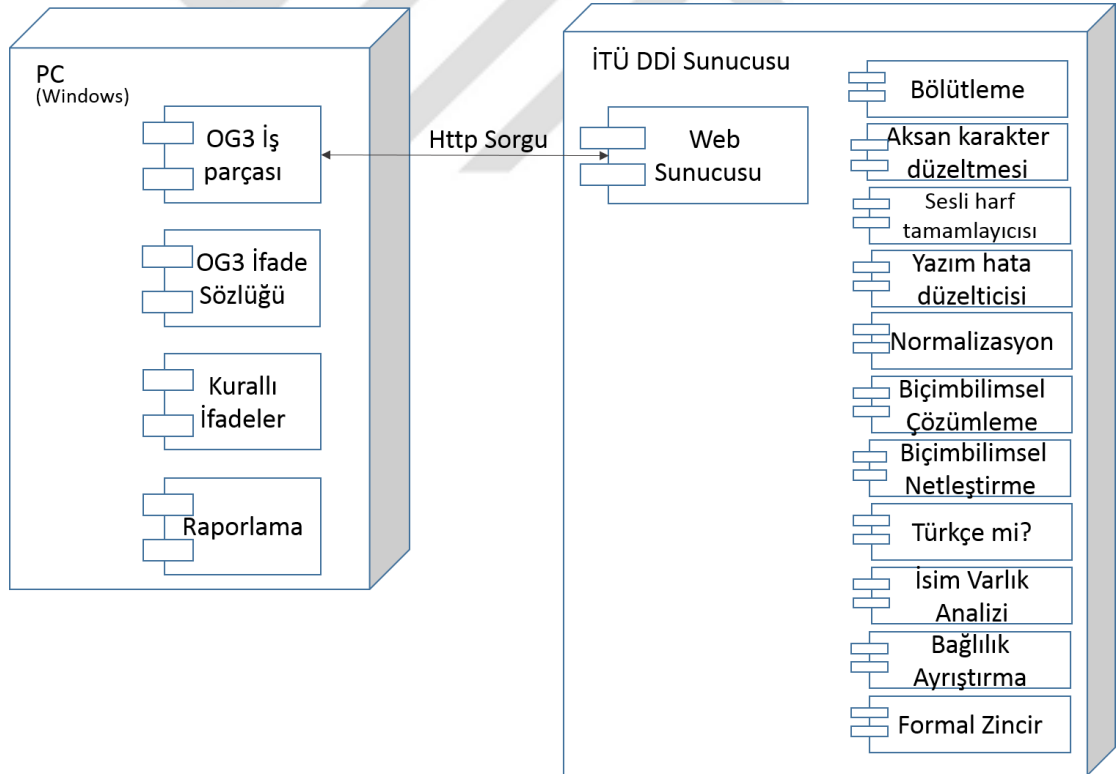
Gereksinimlerde sözdizimsel analizler kapsamında faydalanılan bağıllık ayrıştırma analizleri basit bir örnek ile Şekil 3.4’de açıklanmıştır. En son boş olarak ana tanımlanan düğüm yüklem (PREDICATE) ile arasında bağıllık olan cümlenin temel düğümüdür. Yüklemem sonra OBJECT bağıllığı isim olan nesneyi, MODIFIER bağı niteleyen bir sıfat ile isim arasındaki bağı temsil etmektedir. Son olarak “Yazılım” kelimesi ile yüklem arasında SUBJECT yani özne bağıllığı bulunmaktadır. Bu bağıllık bilgileri sözdizimsel analizlerde kullanıldığı gibi biçim bilimsel analizlerde de faydalanılabilen bilgilerdir.



Şekil 3.4: Örnek bir gereksinim için bağıllık ağacı

### 3.3.2 Gereksinim gözden geçirme algoritmaları

Tez kapsamında Türkçe gereksinimleri kontrol listesi maddelerine dayanarak otomatik olarak gözden geçirmek amacı ile geliştirilen OG3 yazılımı için dağılım şeması Şekil 3.5 ile özetlenmiştir. Genel anlamda Türkçe doğal dil işleme algoritmaları, temel dizi operasyonları ve kurallı ifadelerin bileşimlerinden oluşan algoritmalar bir temel iş parçası (*thread*) tarafından çalıştırılmaktadırlar.



Şekil 3.5: OG3 dağılım şeması

Gözden geçirme iş parçası, çalıştırılan kontrol listesi maddesi algoritmalarına bağlı olarak, İTÜ Doğal Dil İşleme sunucu ile http sorgu-cevap prosedürleri ile

haberleşerek, gerektiğinde iç analiz kütüphanesinden sözlüksel analizler yaparak ya da kurallı ifadeler ve temel dizi operasyonları ile analizleri gerçekleştirmektedir. Kendi kütüphane fonksiyonları ile gerçekleştirilen analizler milisaniyeler seviyesinde gerçekleştirilebilirken, ITÜ sunucusu aracılığı ile yapılan analizler ağ ve sunucu performansına göre saniyeler mertebesinde sürmektedir.

Kontrol listesinde maddeleri için ITÜ DDİ zincirinden bölütleme fonksiyonu, metin düzeltici fonksiyonlar, biçim bilimsel işleme fonksiyonu ve Şekil 3.3’de örneklenen zincirdeki tüm safhaları arka arkaya çalıştırılarak tek çıktı üreten formal dil zincir algoritması kullanılmıştır.

**Tablo 3.3:** Analiz edilen örnek gereksinim

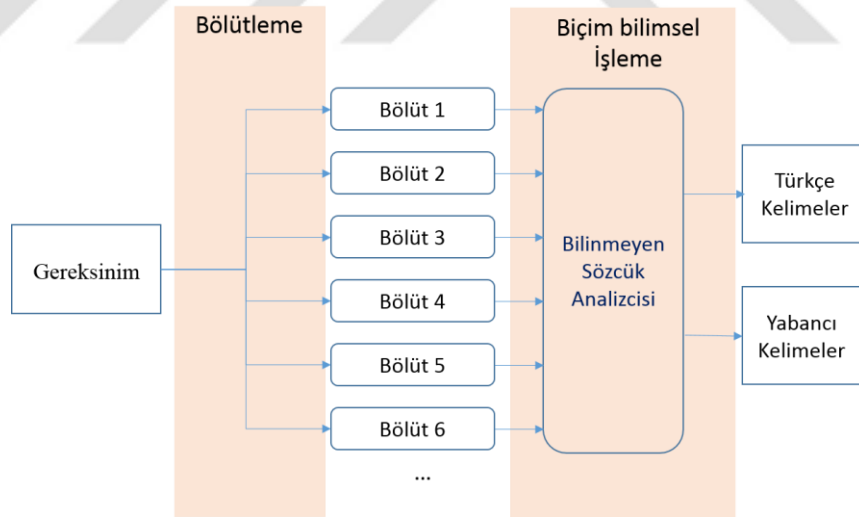
Numarası	Metni	Tip	İzlenebilirlik
ABC_13	ABC yazılımı, regulator modu NORMAL ise ekran sıcaklığını, geçerli sıcaklığın değerine en yakın tam sayıya yuvarlanmış olarak ayarlayacaktır.	GEREKSİNİM	USG_ABC_4

ITÜ DDİ sunucusundan biçim bilimsel sorgu dönüşlerinde elden edilen çıktılar bir sözcük için olası kök ve ek dizilerinin oluşturulmasıdır [34]. Tablo 3.3’de örnek olarak verilen bir gereksinim için bölütleme ve arkasından biçim bilimsel analiz algoritmasının sonuçları Tablo 3.4’de verilmiştir. Analiz dönüşünde kullanılan İngilizce etiket ve kısaltmaların Türkçe karşılıkları Ek C- DDİ Zinciri Sözlük Detayları tablolarında gösterilmektedir. Biçim bilimsel çözümleyici mimarisinde eklerin ifade edildiği Tablo C.3’de “Ekin Kendisi” kolonunda kullanılan bazı karakterlerin hangi harfleri ifade ettiği de Tablo C.2’de belirtilmiştir.

**Tablo 3.4:** DDİ Biçim bilimsel analiz sorgusu dönüşü

ABC	ABC+Guess+Noun+Prop+A3sg+Pnon+Nom
yazılımı	yazılım+Noun+A3sg+Pnon+Acc
	yazılım+Noun+A3sg+P3sg+Nom
	yazılı+Noun+NAdj+A3sg+P1sg+Acc
,	,+Punc
<b>regulator</b>	regulator+Guess+Noun+A3sg+Pnon+Nom
modu	mod+Noun+A3sg+Pnon+Acc
	mod+Noun+A3sg+P3sg+Nom
NORMAL	NORMAL+Guess+Noun+Prop+A3sg+Pnon+Nom
ise	is+Noun+A3sg+Pnon+Dat
	ise+Conj
	ise+Postp+Cond+A3sg
ekran	ekran+Noun+A3sg+Pnon+Nom
sıcaklığımı	sıcaklık+Noun+A3sg+P2sg+Acc
	sıcaklık+Noun+A3sg+P3sg+Acc
,	,+Punc
geçerli	geçerli+Adj
	geçerli+Noun+NAdj+A3sg+Pnon+Nom
	geçer+Noun+NAdj+A3sg+Pnon+Nom^DB+Adj+With
	geçer+Noun+NAdj+A3sg+Pnon+Nom^DB+Adj+With^DB+ Noun+Zero+A3sg+Pnon+Nom
sıcaklığın	sıcaklık+Noun+A3sg+Pnon+Gen
	sıcaklık+Noun+A3sg+P2sg+Nom
değerine	değer+Noun+A3sg+P2sg+Dat
	değer+Noun+A3sg+P3sg+Dat
en	en+Adverb en+Noun+A3sg+Pnon+Nom
yakın	yak+Verb+Pos+Imp+A2pl
	yakın+Verb+Pos+Imp+A2sg
	yakın+Adj
	yakın+Noun+NAdj+A3sg+Pnon+Nom
	yakı+Noun+A3sg+P2sg+Nom
tam	tam+Adj
	tam+Noun+NAdj+A3sg+Pnon+Nom
	tam+Adverb
sayıya	sayı+Noun+A3sg+Pnon+Dat
yuvarlanmış	yuvarlan+Noun+A3sg+Pnon+Nom^DB+Verb+Zero+Narr+A3sg
	yuvarla+Verb+Pass+Pos+Narr+A3sg
	yuvarla+Verb+Pass+Pos^DB+Adj+NarrPart
	yuvarla+Verb+Pass+Pos^DB+Adj+NarrPart^DB+Noun+Zero+A3sg+Pnon+Nom
	yuvar+Noun+A3sg+Pnon+Nom^DB+Verb+Acquire+Pos+Narr+A3sg
	yuvar+Noun+A3sg+Pnon+Nom^DB+Verb+Acquire+Pos^DB+Adj+NarrPart
	yuvar+Noun+A3sg+Pnon+Nom^DB+Verb+Acquire+Pos^DB+ Adj+NarrPart^DB+Noun+Zero+A3sg+Pnon+Nom
olarak	olarak+Postp+PCNom
	ol+Verb+Pos^DB+Adverb+ByDoingSo
ayarlayacaktır	ayarla+Verb+Pos+Fut+A3sg+Cop
.	,+Punc

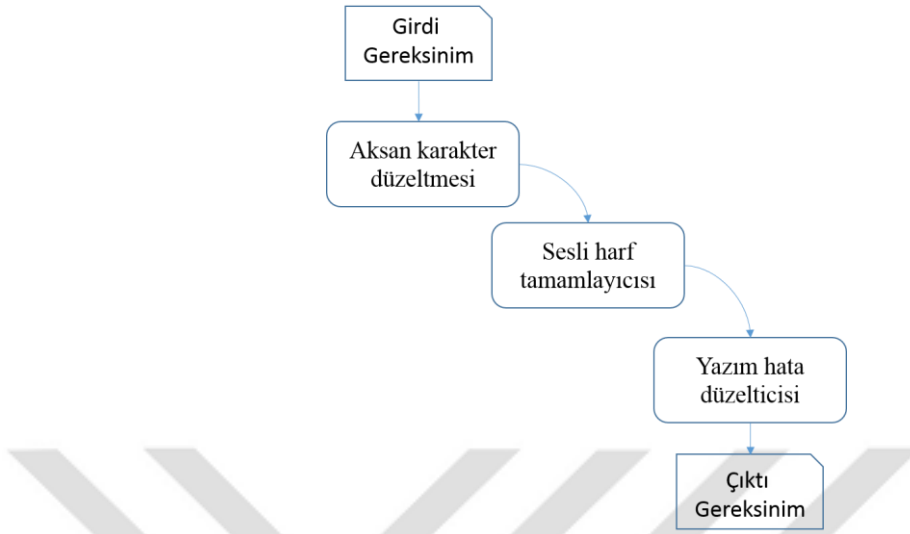
K1 kontrol maddesine göre gereksinim yazım dili kontrolü için (*Türkçe mi?*) gerçekleştirilen algoritmalar ITÜ doğal dil işleme zinciri safhalarından faydalanmaktadır. Zincir üzerinde tam cümle veya birden fazla cümle için topluca yazım dili kararı veren herhangi bir algoritma bulunmamaktadır. Bu sebeple Şekil 3.6'da gösterildiği üzere zincirden faydalanılarak bir veya birden fazla cümle için gereksinimlere özel bir yaklaşım geliştirilmiştir. İlk olarak bölütleme fonksiyonu ile gereksinim metinleri kelime ve noktalama işaretlerine kadar alt bölütlere parçalanırlar. Alt bölütler üzerinden ayrı ayrı dil analizleri yapılmaktadır. İlk tercih olarak bir kelimenin Türkçe olup olmadığı kararına varan ITÜ dil işleme fonksiyonu denenmiş, fakat deneyler sonucunda daha tutarlı sonuçlar elde etmek amacı biçim bilimsel analiz sorgusu kullanılmıştır. "GUESS" veya kelimelere eklenen "+?" etiketleri içeren sorgu dönüşlerinin alınması bilinmeyen sözcüklere işarettir [34]. Tablo 3.4'de "regulator" kelime için verilen analiz sonucunda kelimenin Türkçe olmadığı anlaşılmakta, bu tip kelime sınıflandırılmaları sonucunda Türkçe sözcük olarak tanımlanamayan kelimeler ile gereksinimdeki toplam Türkçe kelime sayısı ile karşılaştırılarak dil kararı verilmiştir.



Şekil 3.6: K1 algoritması ile dil kararı

Dokümanda K2 kontrol maddesine göre yazım ve dilbilgisi hatalarının arındırılması için ITÜ doğal dil işleme zinciri safhalarından faydalanmaktadır. Şekil 3.7'de özetlenen, ASCII karakterlerin muhtemel Türkçe karşılıklarına şartlı rastgele alanlar (CRF) metodu yardımı ile çevrildiği aksan karakter düzeltmesi, sesli harf tamamlayıcısı ve yazım hata düzelticisi gibi fonksiyonların birleşiminden oluşan

metin normalizasyonu yaklaşımı kullanılmış, düzeltilen gereksinimler ile orijinal gereksinimler arasındaki fark incelenerek hata raporlaması yapılmıştır.



Şekil 3.7: K2 ile metin normalizasyonu

K3 kontrol maddesine göre gereksinimler izlenebilirliklerini sağlayan numaraların gereksinimleri belirleyici bir şekilde özgün tanımlanması gerekmektedir. Bu kapsamda, tüm gereksinim nesnelere için gereksinim numaraları modelden çekilerek temel dizi operasyonları ile eşitlik kontrolü yapılmaktadır. Eğer aynı numara ile tanımlı birden fazla gereksinim nesnesine rastlanırsa, her iki nesne için de hata kaydı oluşturulmaktadır.

DDİ zincirindeki tüm safhaların arka arkaya çalıştırılıp tek bir çıktı üretildiği formal dil zincir fonksiyonunun örnek bir gereksinim üzerinde uygulama sonucu Tablo 3.5’de gösterilmiştir. DDİ zinciri fonksiyonu sorgu dönüşlerinde elde edilen bu veriler K4, K5, K7 ve K13 gibi gözden geçirme algoritmaları içerisinde ilgili kontrollerine bağlı olarak analiz edilmektedir.

**Tablo 3.5:** DDİ formal dil zincir sorgusu dönüşü

1	ABC	ABC	Noun	Prop	A3sg Pnon Nom	2	MWE
2	yazılımı	yazılı	Noun	NAadj	A3sg P1sg Acc	20	OBJECT
3	,	,	Punc	Punc	_	2	PUNCTUATION
4	regulator	regulator	Noun	Noun	A3sg Pnon Nom	5	POSSESSOR
5	modu	mod	Noun	Noun	A3sg P3sg Nom	20	SUBJECT
6	NORMAL	NORMAL	Noun	Prop	A3sg Pnon Nom	7	ARGUMENT
7	ise	ise	Postp	Postp	Cond A3sg	20	MODIFIER
8	ekran	ekran	Noun	Noun	A3sg Pnon Nom	9	POSSESSOR
9	sıcaklığımı	sıcaklık	Noun	Noun	A3sg P3sg Acc	20	OBJECT
10	,	,	Punc	Punc	_	9	PUNCTUATION
11	geçerli	geçerli	Adj	Adj	_	12	MODIFIER
12	sıcaklığın	sıcaklık	Noun	Noun	A3sg Pnon Gen	13	POSSESSOR
13	değerine	değer	Noun	Noun	A3sg P3sg Dat	18	MODIFIER
14	en	en	Adverb	Adverb	_	15	MODIFIER
15	yakın	yakın	Adj	Adj	_	17	MODIFIER
16	tam	tam	Adverb	Adverb	_	17	MODIFIER
17	sayıya	sayı	Noun	Noun	A3sg Pnon Dat	18	MODIFIER
18	yuvarlanmış	yuvarla	Verb	Verb	Pass Pos Narr A3sg	19	ARGUMENT
19	olarak	olarak	Postp	PCNom	_	20	MODIFIER
20	ayarlayacaktır	ayarla	Verb	Verb	Pos Fut A3sg Cop	0	PREDICATE
21	.	.	Punc	Punc	_	20	PUNCTUATION

K4 kontrol maddesine göre gereksinimler net ve güçlü bir dil ile ifade edilmelidir. Bu analiz, İngilizce gereksinimler için basit bir kelime taraması "*shall*" ile yapılabilirken, sondan eklemeli bir dil olan Türkçe’de ise kelime, ek ve sözdizimsel analiz gibi detaylı işlemlere ihtiyaç vardır. Öncelikle bu analizin sadece gereksinimler için yapılması yani yorum ve açıklamalar için gerçekleştirilmemesi gerekmektedir. Bu amaçla gereksinim dokümanından nesne Tipi "GEREKSİNİM" veya "TÜRETİLMİŞ G." olan gereksinim metinleri için arka arkaya çalıştırılan ön işleme algoritmalarından bölütleme, biçim bilimsel işleme algoritmaları ile sözcük analizleri ve tip netleştirme ile yüklem belirleme ve son olarak yüklemelere özel ek taramaları yapılmaktadır.

atama	at	Verb	Verb	Able Neg Imp A2sg	4	OBJECT
yapılacaktır	yap	Verb	Verb	Pass Pos Fut A3sg Cop	0	PREDICATE
yapılmayacaktır	yap	Verb	Verb	Pass Neg Fut A3sg Cop	0	PREDICATE
yapılmalıdır	yap	Verb	Verb	Pass Pos Neces A3sg Cop	0	PREDICATE

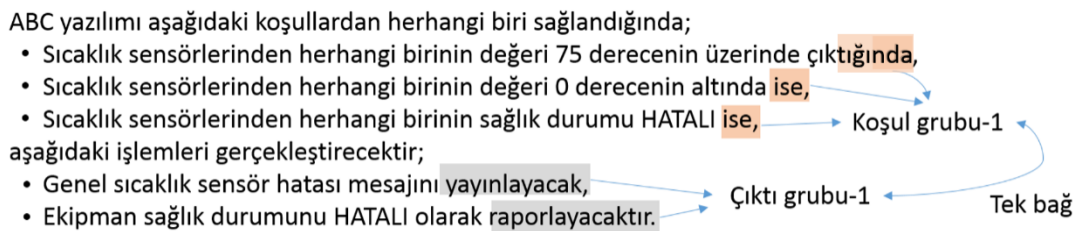
**Şekil 3.8:** Kontrol algoritmalarında yüklem analizi

Tablo 3.5’de verilen “*ayarlayacaktır*” yüklemi için verilen analiz sonuçlarına benzer şekilde farklı farklı yüklemeler için yapılan analiz örnekleri Şekil 3.8’de verilmiştir. Üçüncü ve dördüncü kolonlarda sözcük sınıfları fiil “*Verb*” olan kelimeler için beşinci kolonda ise kelimenin ekleri analiz edilmiştir. Olumsuzluk eki

için “Neg” K5 maddesi için, olumlu ekler için “Pos” etiketi K4 ve gereklilik eki için tanımlanan “Neces” etiketi K13 maddesi için kullanılacak etiketlerdir. Fakat bazı durumlarda “Neces” ekinin “lık”, “lik” gibi eklerde de kullanılması ve “Pos” ve “Neg” in tek başlarına tüm eki tanımlayamadıkları için eklerin kendilerinin de kelimedede taranılıp bulunma ihtiyacı vardır. Son kolonda verilen bağıllık analiz sonucunda yüklemi tarif eden “PREDICATE” etiketi ile birlikte, “yapılacaktır” örneğinde olduğu gibi K4 maddesine göre pozitif, güçlü ve net bir dil kullanılan yüklem analizi *-ecek, -acak* ek taraması yapılarak tamamlanmaktadır.

K5 kontrol maddesine bağlı olarak gereksinimler için negatif ifade analizi birebir K4’de tanımlanan analizleri içermekte, sadece son aşamada yüklemde ek taraması *-meyecek -mayacak* ekleri için yapılmaktadır. Geniş zamanda negatif *-mez -maz* ekleri gibi diğer zamanlara özel bir negatif ek taraması yapılmasına ihtiyaç bulunmamaktadır. Bu ekleri içeren yüklemeler zaten K4 maddesi kontrolüne takılacak ve raporlanacaktır. K5 maddesi için negatif ifade içeren bir yüklem analiz örneği Şekil 3.8’de “*yapılmayacaktır*” kelimesi için verilmiştir. Ayrıca K13 kontrol maddesinde belirtilen tavsiye edici ifadelerin belirlenmesi için yine yüklemeler üzerinde ek taramaları gerçekleştirilmektedir. Gereksinim de gerçekleştirilecek eylemi belirleyen yüklem bulunduğu *-meli -malı* ekleri taranmakta ve sonucuna göre hata raporlanmaktadır.

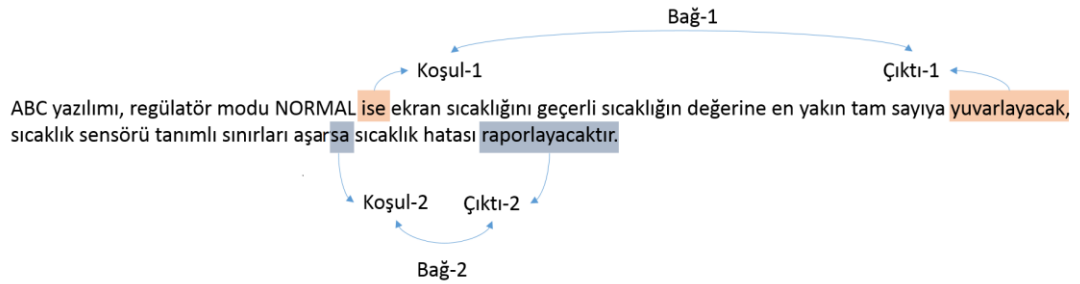
Gereksinimlerin net bir şekilde anlaşılma, geliştirilme ve doğrulanmasına engel olacak muğlak ifadeler kullanılmamalıdır. Bu amaçla Ek D’de verilen K6 maddesi için oluşturulmuş sözlük vasıtasıyla muğlak ifade analizi yapılarak, tipi "GEREKİNİM" veya "TÜRETİLMİŞ G." olan doküman nesneleri için yapılan ihlaller raporlanmaktadır. Benzer şekilde gereksinimlerde belirtilen her fonksiyonu yazılım veya sistemin kesin ve net bir şekilde karşılaması gerekmektedir. Gereksinimlerin kesinlik özelliği analizi için yine Ek D’de K8-K12 kontrol maddeleri için hazırlanan sözlükler ile ifade analizi yapılmaktadır.



Şekil 3.9: Tek bağı gereksinim örneği



Tez kapsamında yapılan deneylerde karşılaştırma amacı ile kullanılan Tiger PRO yazılımında olduğu gibi, birçok gözden geçirme yazılımı birden fazla gereksinim içeren gereksinim taramalarını tam anlamı ile gerçekleştirememekte ve basitçe birden fazla cümle içeren gereksinimleri potansiyel olarak hatalı raporlamaktadır. Şekil 3.9’da örneği verildiği üzere, gereksinimler birden fazla cümle veya paragraf içerebilir, aynı çıktıları üreten birden fazla koşul ve aynı koşullara bağlı olarak birden fazla çıktı da içerebilirler. Fakat aynı gereksinim içerisinde farklı koşullarda, farklı çıktılar üretilmesi birden fazla gereksinim içerildiğine işaretler ve detaylı algoritmalar ile bunların tespit edilmesi gerekmektedir. Şekil 3.9’da verildiği gibi, koşul veya koşullar sağlandığında tüm çıktılar aynı anda gerçekleştiriliyorsa K7 kontrol maddesi yaklaşımında bu tek bağ olarak tanımlanır ve hata olarak raporlanmaz. Koşul veya koşul grubu ile çıktı veya çıktı grubu arasında birden fazla bağ yakalandığında birden fazla gereksinime işaret edilmektedir ve örneği Şekil 3.10 ile verilmiştir.



Şekil 3.10: K7 algoritmasının tespit edeceği birden fazla bağ içeren gereksinim

Gereksinimlerde birden fazla bağ tespiti temelde koşul ve çıktılar tespiti üzerine kurulu olduğu için koşullar için fiil, fiilimsi ve bağlaç analizi, çıktılar için ise yüklem analizleri kullanılmıştır. Koşulları belirleyen fiilimsiler için farklı tip ve durumlar için farklı analiz şekilleri tanımlanmıştır. Şekil 3.11’de verilen analizde sözcük sınıflarında fiilimsiler “Noun” Adj” ve “Pastpart” kombinasyonları ile tespit edilmekte ve bulunduğu sözcük içerisinde *-diğında, -diğinde, -duğunda, -duğünde, -tığında, -tığında, -tuğunda, -tüğünde, -dığı, -diği, -duğu, -duğü, -tığı, -tiğı, -tuğu, -tüğü*, gibi fiilimsi yapan ekler taranmaktadır.

girdiğinde	-	Noun	PastPart	A3sg P3sg Loc	7	MODIFIER
geldiği	-	Adj	PastPart	P3sg	5	MODIFIER

Şekil 3.11: Fiilimsi koşul analiz - Tip 1

Fiilimsiler için ikinci tip koşul analizleri ise Şekil 3.12’de verilen zarf fiil ekleri ile koşul anlamı katan fiilimsi tespiti üzerine kurulmuştur. Burada zarflara işaret eden “*Adverb*” sözcük sınıfı ve diğer kolonda ekin koşula kattığı zamansal anlamları ifade eden “*While*”, “*When*” ve “*WithoutHavingDoneSo*” etiketleri kullanılmıştır. “*While*” etiketi süreç belirterek zarf fiil yapan *-ken* ekini, “*When*” etiketi zaman belirterek zarf fiil yapan *-ınca, -ince, -unca, -ünce* eklerini, “*WithoutHavingDoneSo*” etiketi ise “*yapılmadan önce*” şeklinde koşul anlamı katan *-maksızın, -meksizin* eklerini ifade etmektedir.

seyrederken	_	Adverb	While	_	7	MODIFIER
gidince	_	Adverb	When	_	7	MODIFIER
gelmeden	_	Adverb	WithoutHavingDoneSo	_	4	MODIFIER

Şekil 3.12: Fiilimsi koşul analizi -Tip 2

Gereksinimlerde fiiller kullanılarak da koşullar ifade edilmektedir. Bu kapsamda, Şekil 3.13’de örnekler verilen fiil koşullar için sözcük sınıflarında “*Verb*” bulunduğu, ek analizleri kolonunda koşul şart eki *-se, -sa* ‘yı tanımlayan “*Cond*” etiketi taranmaktadır. Bulduğunda ise fiil koşulun gereksinimdeki yerini ifade eden indeks kayıt edilmektedir.

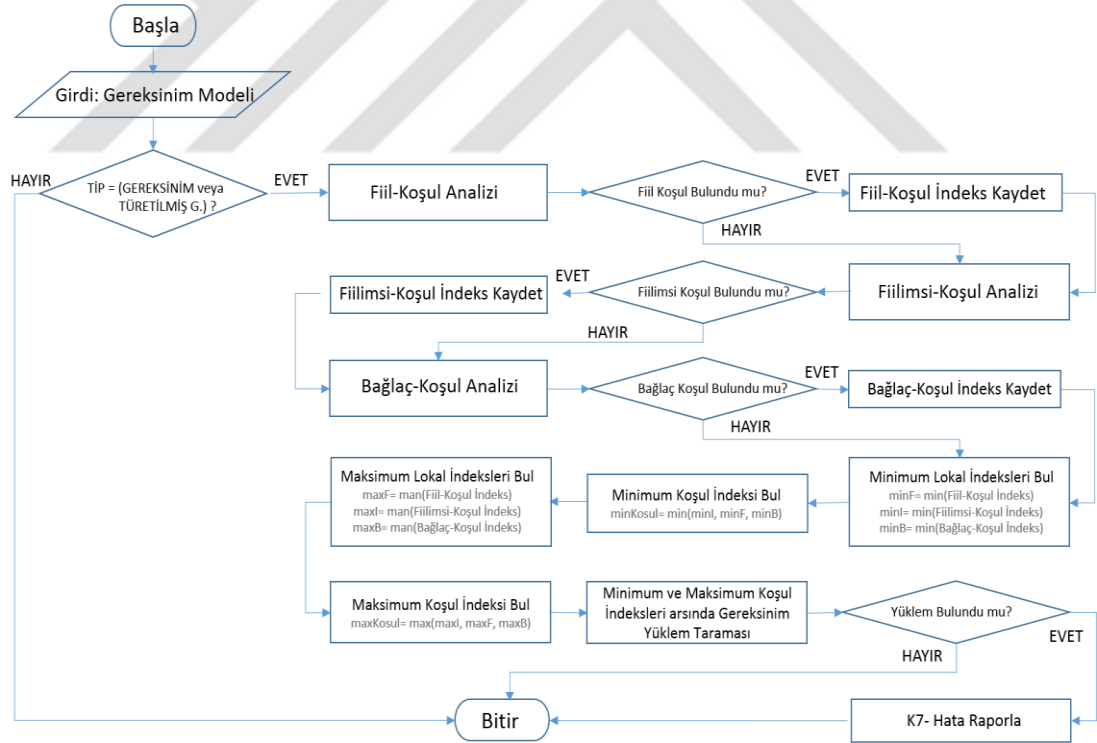
gelirse	gel	Verb	Verb	Pos Aor Cond A3sg	7	MODIFIER
büyükse	_	Verb	Zero	Cond A3sg	8	MODIFIER
aşarsa	aş	Verb	Verb	Pos Aor Cond A3sg	10	MODIFIER

Şekil 3.13: Fiil koşul analizi

Koşul analizlerinin sonuncusu Şekil 3.14’de örneklenen bağlaç olan *-ise* tespittir. Sözcük sınıflarında “*Postp*” etiketi bulunduğu ek analiz kolonunda “*Cond*” sözcüğü ve tam kelime taraması ile birebir *-ise* ‘ye eşitlik kontrolü yapılmaktadır.

Şekil 3.14: Bağlaç koşul analizi

Birden fazla gereksinim analizi için geliştirilen K7 algoritması, içerilen gereksinim sayısı önemsenmeksizin iki farklı koşul da iki farklı çıktı bulma analizi kurulmuştur. Algoritmada ITÜ DDİ işleme sunucusundan dil analizi için tek bir sorgu yapılmakta ve dönülen analiz sonuçları üzerinde Şekil 3.15’de verilen fonksiyonlar çalışmaktadır. Tipi "GEREKSİNİM" veya "TÜRETİLMİŞ G." olan doküman nesneleri üzerinde yukarıda örnekleri verildiği üzere fiil, fiilimsi ve bağlaç olan koşullar taranarak indeksleri kayıt edilir. Her koşul tipinin en küçük ve en büyük indeksleri seçilip, bu indeksler arasında yine en küçük ve en büyük genel koşul indeksleri belirlenir. Genel koşul indeksleri arasında K4’ü ihlal etmeyen bir yüklem bulunduğu, bu yüklem ile bir gereksinim tamamlandıktan sonra yeni bir gereksinimin koşulu geldiği anlaşılır ve K7 hatalı olarak raporlanır.

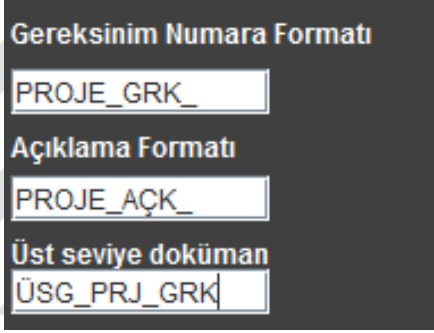


Şekil 3.15: K7 Algoritma Akış diyagramı

K14 kontrol maddesine göre gereksinimlerde eksiklik belirten ifadeler için tipi "GEREKSİNİM" veya "TÜRETİLMİŞ G." olan doküman nesneleri üzerinde Ek D’de verilen K14 Sözlüğü vasıtası ile sözlüksel analizler yapılmaktadır. K15 ve K16

maddeleri gereğince nesne tipi "GEREKSİNİM" olanlar için gereksinim modeli üzerinde izlenebilirlik taraması yapılmakta, tipi "TÜRETİLMİŞ G." VE "AÇIKLAMA" olanlar için tam tersi herhangi bir izlenebilirliğin çekilmediği kontrol edilmektedir. K19 maddesine göre ise bu tanımlı nesne tipleri ile uyumsuz bir nesne tipi kullanımı olup olmadığı gereksinim modeli üzerinde taranmaktadır.

K17, K18 ve K20 kontrolleri için temel dizi operasyonları ile birlikte kurallı ifadelerden (*regex*) faydalanılmıştır. Gereksinim, türetilmiş gereksinim ve açıklamalar için proje standartlarında belirlenen numara ve izlenebilirlik formatları Şekil 3.16'da ki ara yüz ile kullanıcıdan girdi olarak beklenmekte ve tanımlanan kurallı ifadeler ile standart dışı kullanım veya yanlış bir dokümana çekilen izlenebilirlik de bu şekilde belirlenebilmektedir.

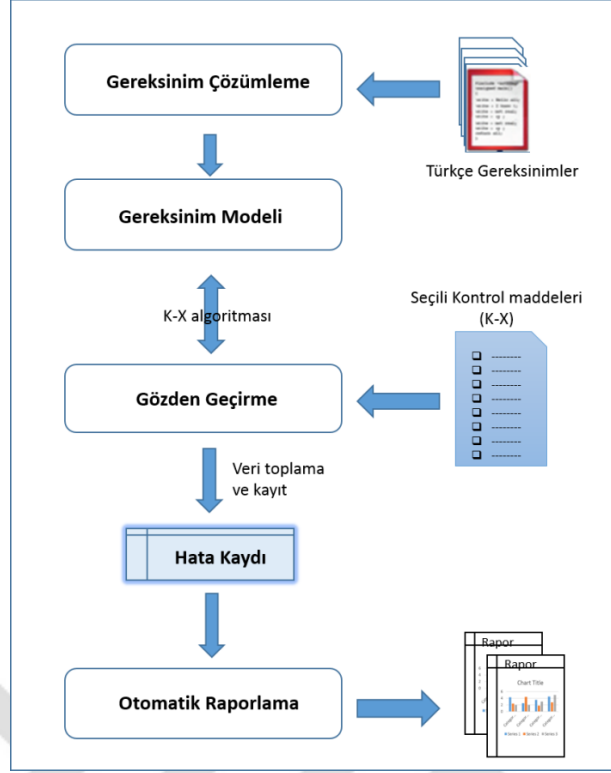


Gereksinim Numara Formatı  
PROJE\_GRK\_  
Açıklama Formatı  
PROJE\_AÇK\_  
Üst seviye doküman  
ÜSG\_PRJ\_GRK|

Şekil 3.16: Kullanıcı arayüzü ile beklenen numara ve izlenebilirlik formatı

### 3.3.3 Otomatik Gereksinim Gözden Geçirme Yazılım Tasarımı (OG3)

Tablo 3.1 OGGKL'de ortaya konulan kuralların proje ekipleri yerine otomatik olarak kontrol edilmesi amacı ile Java programlama dili ile bir gözden geçirme yazılım aracı geliştirilmiştir. Geliştirilen Otomatik Gereksinim Gözden Geçirme Yazılımı (OG3) ile gerçekleştirilen gözden geçirme süreci girdi ve çıktıları ile birlikte Şekil 3.17'de özetlenmiştir.



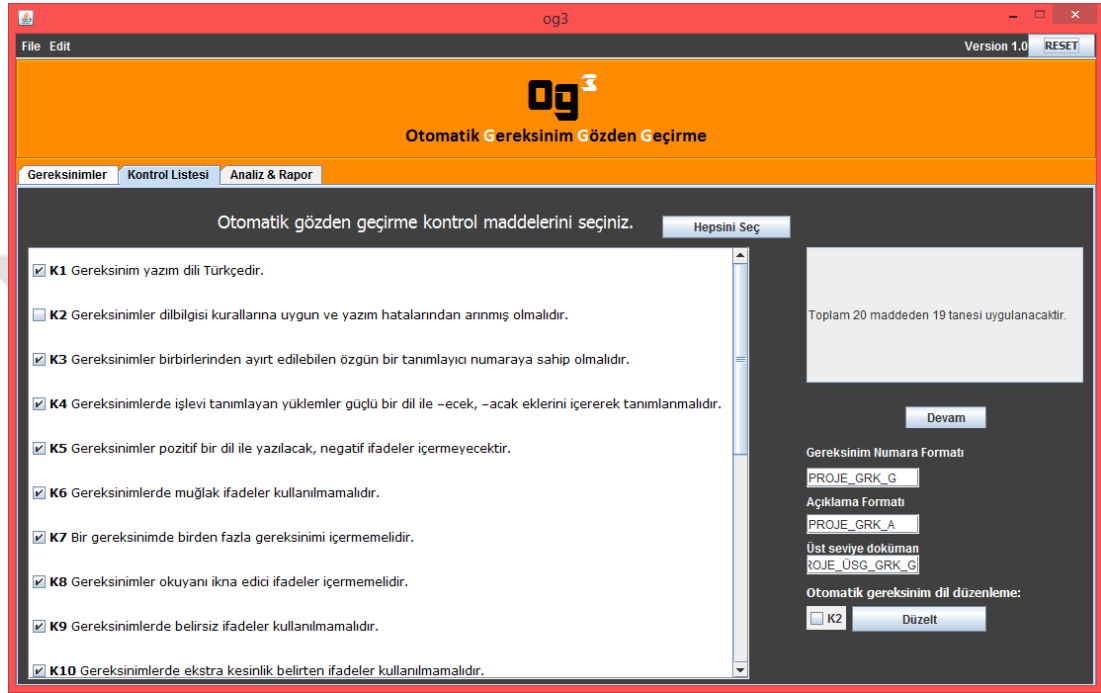
Şekil 3.17: OG3 İle Otomatik Gereksinim Gözden Geçirme Süreci

Yazılımda gözden geçirme süreci öncelikle gözden geçirilmek istenen gereksinim dokümanını sisteme yüklenmesi ile başlamaktadır. Gözden geçirme yazılımına yüklenmiş bir gereksinim dokümanı Şekil 3.18'deki ekran görüntüsü ile verilmiştir.

Gereksinim No.	Gereksinim	Tip	İzlenebilirlik
EDU_PRJ_GRK_1	Sistem, oyunu başlatmadan önce temel bilgisayar özelliklerini deneyecek ve gerekli eğitimleri verecektir.	GEREKŞİNİM	
EDU_PRJ_AÇK_2	Not: Temel bilgisayar özellikleri arasında fare ve klavye girişi bulunur, bilgisayar ekranlarını ve geri beslemeleri anlar (...)	AÇIKLAMA	
EDU_PRJ_GRK_3	Sistem oyun içi dersler ve yardımları rastgele sağlayacaktır.	GEREKŞİNİM	
EDU_PRJ_AÇK_3	Değil: Sistem sınıflarda kullanılmak üzere tasarlanmıştır, bu nedenle oyun yardımı ihtiyaç duyan öğrencilere destek s...	AÇIKLAMA	
EDU_PRJ_GRK_5	Oyunun başlatılacağı kadar oynanabilir maksimum süre 5 dakikadır.	GEREKŞİNİM	USG_EDU_GRK_5
EDU_PRJ_AÇK_4	Uygulama başlatıldıktan sonra, oynatıcı verileri yüklemek ve oyundan son kaldığı yerden oynamaya başlaması 5 daki...	AÇIKLAMA	
EDU_PRJ_GRK_7	Çocuklar için tanıdık kullanıcı arabirimleri sağlanmıştır.	TURETİLMİŞ G.	USG_EDU_GRK_7
EDU_PRJ_AÇK_5	Oyunun kendisi için kullanıcı arabirimleri (yük ekranları ve oyun içi menüler dahil) diğer çocuk oyunlarında bulunanlar...	AÇIKLAMA	
EDU_PRJ_GRK_9	Öğretmenlere tanıdık kullanıcı arayüzü	TURETİLMİŞ G.	
EDU_PRJ_AÇK_6A	Sorgular için arayüz, gezinme ve dosya erişimi açısından bir web tarayıcısına benzeyecektir. Navigasyon, ilerleme ve ger...	AÇIKLAMA	
EDU_PRJ_GEK_11	Long Stretch Software, sistemin tüm taraflar tarafından kullanılmasını sağlamak için KOBİler ile birlikte çalışacaktır.	TURETİLMİŞ G.	
EDU_PRJ_AÇK_2	Sistemin tüm deneyim düzeyindeki insanlar için kullanılabilirliğini sağlamak için öğretmenler, çocuk psikologları ve ...	AÇIKLAMA	USG_EDU_GRK_12
EDU_PRJ_GRK_13	Yazılım başlatıldığında % 99 ihtimalle çalışacaktır. İşletim sisteminin oyunun başlatılmasını engelleyebilecek durum...	GEREKŞİNİM	USG_EDU_GRK_13
EDU_PRJ_GRK_14	Beklenen sistem çalışma süresi % 95 olacaktır. Oyunun kendisi, zamanın % 95'inde en az 3 ardışık saatte çalışabilir. ...	GEREKŞİNİM	USG_EDU_GRK_14
EDU_PRJ_GRK_15	Sistem tarafından çevrimiçi yedekleme her zaman sağlanacaktır. Yazılım, oyun çökerse veri kaybı olmasını sağlam...	GEREKŞİNİM	USG_EDU_GRK_15
EDU_PRJ_GRK_16	Sistem, beklenmedik girdilerin neden olduğu hatalara eğilimli olmayacaktır. Yazılım, her türlü girdiyi idare edebilecek ...	TURETİLMİŞ G.	USG_EDU_GRK_16
EDU_PRJ_GRK_17	Sistem ağ güvenliğini korur. Yazılım, ağ güvenliği sorunları yaratmamasını sağlamak için uygun ağ güvenlik protokolle...	GEREKŞİNİM	USG_EDU_GRK_17
EDU_PRJ_GRK_18	Sistem, Office dosyası biçimindeki ithalat ve ihrac hatalarını algılar ve düzeltir. Yazılım, minimum veri bozulmasıyla Off...	GEREKŞİNİM	USG_EDU_GRK_18
EDU_PRJ_GRK_19	Anzalar arasındaki süre, oyunu çalıştıran her işletim sistemi için minimuma indirilecektir.	TURETİLMİŞ G.	
EDU_PRJ_AÇK_2	Anzalar Arası Ortalama Süre (MTBF) - MTBF, herhangi bir onarım veya yeniden kurulum olmaksızın oyun ortlama ç...	AÇIKLAMA	
EDU_PRJ_GRK_21	Bir sistem hatası durumunda, öğrencilerin önceden hakım oldukları materyalleri tekrar etmeleri beklenmektedir. Bu ne...	GEREKŞİNİM	USG_EDU_GRK_21
EDU_PRJ_GRK_22	Oyun sabit disk alanının en fazla 80mb'sini kapacak. Ayrıca daha sonra 60 Mhz işlemci ve 16 MB RAM bulunmayan bir...	GEREKŞİNİM	USG_EDU_GRK_22
EDU_PRJ_GRK_23	Oyun birden fazla bilgisayar platformunda mutlak çalışacaktır. Oyun, bir Windows 95 veya üstü veya bir Mac OS 7.6 v...	GEREKŞİNİM	USG_EDU_GRK_23
EDU_PRJ_GRK_24	Veri taşınabilir olacaktır. Oyun veri dosyaları platformdan bağımsız olacaktır.	GEREKŞİNİM	USG_EDU_GRK_24

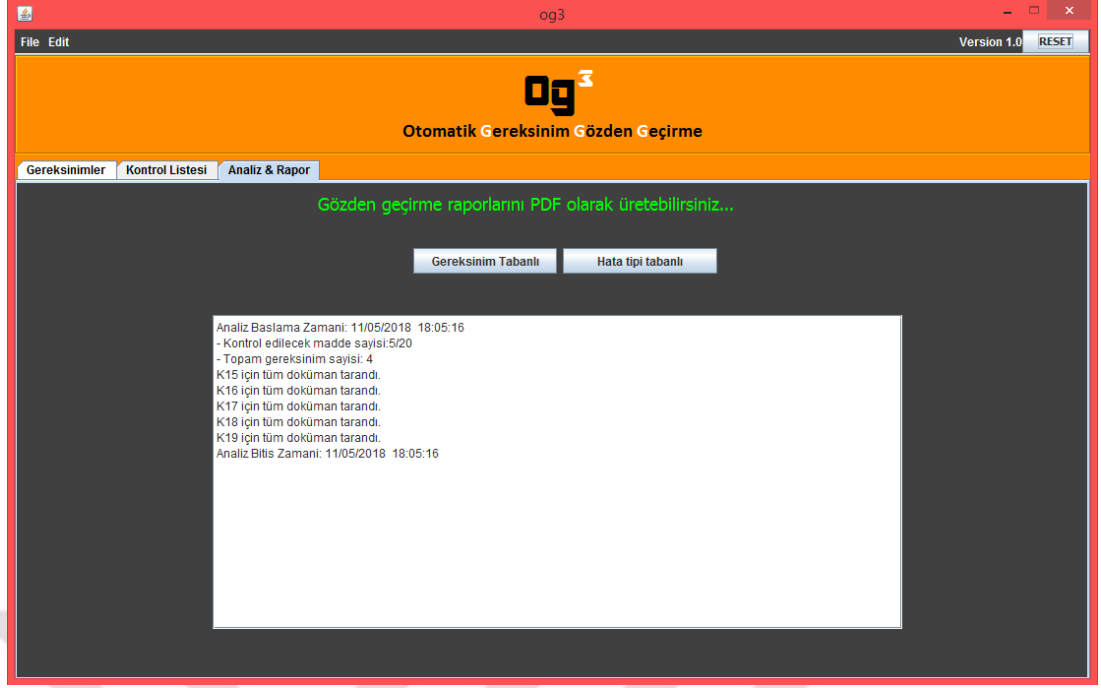
Şekil 3.18: OG3 yazılımı Gereksinimler ekran görüntüsü

Birçok gereksinim yönetim aracından kolayca çıktı olarak alınabilen ve kabul görmüş standart Microsoft XML elektronik tablosu formatında oluşturulan gereksinimler yazılım tarafından çözümlenerek okunur ve gereksinim nesnelarını saklamak üzere oluşturulan gereksinim modeli içerisine aktarılırlar. Şekil 3.19'da görüntüsü verilen kontrol listesi maddelerine karşılık geliştirilen gözden geçirme algoritmalarının tamamı bu model üzerinde çalıştırılır.



Şekil 3.19: OG3 yazılımı Kontrol Listesi ekran görüntüsü

Gereksinim modelinden çekilen gereksinim nesneları ilgili gözden geçirme algoritmaları tarafından analiz edilerek sonuçlar, hem gereksinim nesne numaraları hem de kontrol listesi madde numaraları ile erişilebilecek çiftli dizi şeklinde hata kayıt fonksiyonu tarafından kaydedilirler. Ayrıca gereksinim gözden geçirme boyunca gözden geçirme başlangıç zamanı, o anda kontrol edilen hata tipi ve gözden geçirme tamamlanma zamanı gibi süreç ile ilgili detay durum bilgileri de anlık olarak Şekil 3.20 OG3 yazılımı Analiz & Rapor ekran görüntüsündeki statü alanından takip edilebilir.



Şekil 3.20: OG3 yazılımı Analiz & Rapor ekran görüntüsü

Gözden geçirme sonucunda elde edilen analiz sonuçları, anlaşılır, görsel, istatistiki açıdan zengin, otomatik olarak sınıflandırılmış ve resmi bir gözden geçirme kaydı olarak saklanabilecek bir format ile raporlanmaktadır. Yine Şekil 3.20'de gösterildiği gibi istenirse sonuçlar gereksinim numaraları özelinde, istenirse de kontrol listesi maddeleri özelinde sınıflandırılarak raporlanmaktadır. Ayrıca istatistiki tablolar ve grafiksel özetler içeren bu raporlar Microsoft Word ve Excel dosyası olarak kaydedilebildiği gibi .pdf, .html, .rtf gibi farklı uzantılı dosya formatları ile de kayıt edilebilir.

## DÖRDÜNCÜ BÖLÜM

### DENEYSEL ÇALIŞMALAR

Tezde otomatik gözden geçirme yaklaşımını hayata geçirmek amacı ile ortaya konulan yazılımın performansını ölçmek amacı ile gerçekleştirilecek deneylere öncelikle seçilen gereksinimler üzerinde yapılan manuel gözden geçirmeler ile başlanmıştır. Manuel gözden geçirmeler ile elde edilen sonuçlar daha sonra ortaya konulan yazılım ile elde edilen raporlar ile kıyaslanmıştır. Yapılan manuel gözden geçirmeler, algoritmaları doğrulama anlamında katkı sağlamıştır. Fakat ortaya konulan yaklaşımın başarısını ölçme ve güçlü ve zayıf yönlerini ortaya koyabilme adına yine doğal dile otomatik gözden geçirme yapabilen bir yazılım ile karşılaştırılması ihtiyacı doğmuştur. Bu amaçla literatürdeki otomatik gözden geçirme yaklaşım ve yazılımları incelenerek teorikte ve pratikte yapılabilecek kıyaslamalar araştırılmıştır.

İkinci bölümde özetlenen, Türkçe gereksinimler üzerine yapılan iki çalışmada da ortaya konulan yaklaşımların İngilizce araştırmalardan türetilmesi, gözden geçirme kontrol liste ve yaklaşımlarının giriş seviyesinde olmaları sebebiyle karşılaştırma adına tercih edilmemişlerdir. Bu sebeple, gereksinim dili anlamında en sık kullanılan ve bir çok gözden geçirme yaklaşımına konu olmuş İngilizce doğal dilde gereksinim gözden geçirme yazılımları ile kıyaslama daha uygun görülmüştür. Bu çalışmaların özetlendiği ikinci bölümde, birçok akademik ve sektörel çalışmaya temel olmuş Tiger Pro yazılımı tercih edilmiştir. İngilizce doğal dil yapısının Türkçe 'ye kıyasla otomatik gözden geçirmeye daha uygun ve göreceli daha kolay olması, ayrıca ortaya konulan çalışma sayısının fazlalığı karşılaştırmalar için diğer motive edici sebeplerdir.



## 4.1 Test Ortamı

Tiger Pro ve OG3 otomatik gereksinim gözden geçirme yazılımları ile gerçekleştirilecek deneyler sonucunda elde edilen çıktı ve verilerin aynı şartlar altında sağlanması adına Tablo 4.1 Test Ortam Konfigürasyonunda özetlenen test ortamı oluşturulmuştur. Akademik çalışmalara katkı amaçlı ücretsiz *Shareware* ve öğrenci lisansları sağlayan Tiger Pro yazılımına karşılık, tez için sıfırdan geliştirilen ve ilk versiyonu 1.0 olan OG3 yazılımını Windows 8.1 işletim sistemi üzerinde test edilmiştir.

**Tablo 4.1:** Test Ortam Konfigürasyonu

İşletim Sistemi: Windows 8.1 64 bit işletim sistemi
İşlemci: Intel (R) Core (TM) İ5-4200M CPU 2.50GHz
Bellek: 4 GB RAM
Tiger Pro Yazılımı: Shareware Program ID: 300099862
OG3 Yazılımı: Versiyon 1.0

## 4.2 Tiger Pro ile İngilizce gereksinimlerin otomatik gözden geçirilmesi

Tiger Pro yazılımı gereksinim gözden geçirme kapsamında temelde yedi maddelik bir kontrol listesine sahip ve gözden geçirme sonuçlarını bu maddeler üzerinden raporlara yansıtmaktadır. Tez kapsamında hazırlanan 5 ayrı gereksinim dokümanı yazılımın istediği format olan .csv uzantılı dosyaya dönüştürülmüş ve yazılıma girdi olarak sunulmuştur. İngilizce doğal dilde yazılmış gereksinim dokümanlarında Türkçe versiyonları aynı veri tipleri ile 4 ana kolonda; gereksinim numarası, gereksinim metni, gereksinim tipi ve izlenebilirlik alanları ile oluşturulmuşlardır. Yedi maddelik kontrol listesine gerçekleştirilen gözden geçirmeler sonucunda elde edilen veriler, hataların ayıklanma başarısı ve gerçek anlamda resmi bir gereksinim gözden geçirme süreci ihtiyaçlarının karşılanma başarısı deneyler ile tecrübe edilmiştir.

### 4.2.1 İngilizce Havayolu Rezervasyon Sistemi gereksinimlerinin gözden geçirme sonuçları

Toplam 77 adet İngilizce gereksinimlerden oluşan dokümana deney amaçlı eksik olan izlenebilirlik ve gereksinim tipi kolonları da eklenerek Tiger Pro yazılımına kabul ettiği .csv uzantılı dosya formatı ile girdi olarak sunulmuştur. Tiger

Pro yazılımı ile gözden geçirilen gereksinim dokümanının çıktıları yazılımın rapor dosya formatı olan .prn uzantılı dosya olarak kaydedilmiş ve sonuçları aşağıdaki Tablo 4.2’de özetlenmiştir.

**Tablo 4.2:** Tiger Pro ile Havayolu Rezervasyon Sistemi gereksinim gözden geçirme sonuçları

Hata Tipi	Açıklama	Bulgu Sayısı
Hata Tipi 1	Bir satırda birden fazla gereksinim	10
Hata Tipi 2	Bir satırda muhtemel birden fazla gereksinim	120
Hata Tipi 3	Doğrulanamaz ve test edilemez ifadeler	53
Hata Tipi 4	Gereksinim tanımlarken hatalı kelime kullanımı	28
Hata Tipi 5	Kullanıcının tanımladığı hatalı kelimeler	0
Hata Tipi 6	Muhtemel tasarım kısıtları	0
Hata Tipi 7	Tamamlanmamış ifadeler	23
Analiz Başarı Oranı: (Doğru Bulgu/ Toplam Bulgu)		59/234

Elde edilen istatistikler ışığında hem gözden geçirme algoritmaları hem de İngilizce gereksinimler incelenmiş ve Tiger Pro yazılımının otomatik gereksinim gözden geçirme başarısı üzerine aşağıdaki sonuçlara ulaşılmıştır;

1. hata tipi için raporlanan 10 bulgunun mükerrer sayıldığı, toplam 6 gereksinim için birden fazla gereksinim içerdiği raporlandığı gözlemlenmiştir. 6 gereksinimden sadece 4 tanesinin farklı koşullarda farklı işlevleri anlatan birden fazla gereksinim içerdiği doğrulanmıştır. Bu hata tipi için bulgu başarı oranı 4/10’dur.

2. hata tipi temelde bir kontrol listesi maddesi olarak sınıflandırılmaz. Muhtemel birden fazla gereksinim içeren gereksinimlerin raporlanması hedeflenmiş ve bu da “*ve/veya/ikisi de/her hangi biri*” gibi kelimeleri aramaları yapan algoritmalar ile sağlanmıştır. Fakat bu kelimeler birden fazla gereksinim içeren sözcükler olarak sınıflandırılmaz ve gereksinimlerde koşul tanımlamalarında kullanılması kaçınılmaz ifadelerdir. Bu sebeple 2. hata tipi için bulgu başarı oranı 0/120’dir.

3. hata tipinde doğrulanamaz ifadelerin arandığı, bu ifadelerin bir kısmının doğrulanamaz ifade olarak tanımlanmasının hatalı olduğu görülmüştür. Raporlanan 53 bulgudan, 8 tanesinin gerçek anlamda doğrulanmaz ifade olduğu (*vb, also, enough, high, vb, such as, minimize*), diğerlerinin ise hatalı olarak raporlandığı gözlemlenmiştir (*ayrıca, hepsi, hiç biri, içeren, aynı*). Bu hata tipi için bulgu başarı oranı 8/53’dür.

4. hata tipinde, gereksinim tanımlanırken işlevin en güçlü şekilde ifade edilmesini sağlayan ‘*shall*’ yerine kullanılan tüm “*must, should, will*” gibi ifadeleri

tarayarak hata analizi yapılmaktadır. Burada raporlanan 28 kayıttan mükerrer sayılanlar çıkarıldığında 24 tanesinin doğru raporlandığı gözlemlenmiştir. Bu hata tipi için bulgu başarı oranı 24/28'dir.

5 ve 6. hata tipi için bulgu raporlanmamıştır. 7. Hata tipi için tamamlanmamış ifade analizi için 4. hata tipi benzeri cümlede eksik olan güçlü ifade belirteci “*shall*” aranmış ve 23 adet bulgu raporlanmıştır. Bu bulgular tüm dokümanın gereksinimlerden oluştuğu varsayımı ile başarı oranı 23/23 olarak raporlanabilir. Tiger Pro yazılımı gereksinim olmayan nesnelere tanıyamamaktadır.

#### 4.2.2 İngilizce E-Market gereksinimlerinin gözden geçirme sonuçları

**Tablo 4.3:** Tiger Pro ile E-Market Sistemi gereksinim gözden geçirme sonuçları

Hata Tipi	Açıklama	Bulgu Sayısı
Hata Tipi 1	Bir satırda birden fazla gereksinim	5
Hata Tipi 2	Bir satırda muhtemel birden fazla gereksinim	28
Hata Tipi 3	Doğrulanamaz ve test edilemez ifadeler	29
Hata Tipi 4	Gereksinim tanımlarken hatalı kelime kullanımı	8
Hata Tipi 5	Kullanıcının tanımladığı hatalı kelimeler	0
Hata Tipi 6	Muhtemel tasarım kısıtları	0
Hata Tipi 7	Tamamlanmamış ifadeler	2
Analiz Başarı Oranı: (Doğru Bulgu/ Toplam Bulgu)		16/72

Hata tipi 1 için gözlemlenen başarı Oranı 1/5'dir. 2 adet mükerrer sayım tespit edilmiş, diğerleri ise sadece bir den fazla “*shall*” kelimesi sayarak yanlış karar verdiği gözlemlenmiştir.

Hata tipi 2 için gözlemlenen başarı oranı 1/28'dir. Sadece “/”karakteri içeren bir gereksinim bir den fazla fonksiyon içerdiği doğru raporlanmıştır. “*ve/veya/ikisi de/her hangi biri*” taramaları ile bu karar verilmesi uygun değildir.

Hata tipi 3 için gözlemlenen başarı oranı 4/29'dur. “*Örnek olarak, gibi, kolay, birçok*” ifadeleri doğru tespit edilirken “*ayrıca, hepsi, hiç biri, içeren, aynı*” gibi ifadelerin kullanımlarının hata olarak raporlanması uygun değildir.

Hata tipi 4 için gözlemlenen başarı oranı 8/8. ‘*shall*’ yerine kullanılan tüm “*must, should, will*” gibi ifadeler doğru tespit edilmiştir.

Hata tipi 7 gözlemlenen başarı oranı 2/2'dir. Eksik ‘*shall*’ kelimeleri ise doğru raporlanmıştır.

### 4.2.3 İngilizce Uçuş Kontrol Sistemi gereksinimlerinin gözden geçirme sonuçları

Tablo 4.4: Tiger Pro ile Uçuş Kontrol Sistemi gereksinim gözden geçirme sonuçları

Hata Tipi	Açıklama	Bulgu Sayısı
Hata Tipi 1	Bir satırda birden fazla gereksinim	0
Hata Tipi 2	Bir satırda muhtemel birden fazla gereksinim	30
Hata Tipi 3	Doğrulanamaz ve test edilemez ifadeler	8
Hata Tipi 4	Gereksinim tanımlarken hatalı kelime kullanımı	30
Hata Tipi 5	Kullanıcının tanımladığı hatalı kelimeler	0
Hata Tipi 6	Muhtemel tasarım kısıtları	0
Hata Tipi 7	Tamamlanmamış ifadeler	35
Analiz Başarı Oranı: (Doğru Bulgu/ Toplam Bulgu)		65/103

Hata tipi 2 için gözlemlenen başarı oranı 0/30'dır. Bulgularda “ve/veya/ikisi de/her hangi biri” ve “ / “taramaları ile bu kararın verilmesi uygun değildir.

Hata tipi 3 için gözlemlenen başarı oranı 0/8'dir. ‘Ayrıca, hepsi, hiç biri, içeren aynı’ gibi ifadelerin gereksinimlerde kullanılması hata olarak raporlanması uygun değildir. Açıklamalarda geçen “Maksimum” kelimeleri için doğrulanamaz kararı verilmiş, fakat bu rakamlar bir önceki gereksinimde tanımlandığı için yine hatalı bir karar söz konusudur.

Hata tipi 4 için gözlemlenen başarı oranı 30/30'dur. ‘shall’ yerine kullanılan tüm “must, should, will” gibi ifadeler doğru tespit edilmiştir.

Hata tipi 7 için gözlemlenen başarı oranı 35/35. Eksik olan tüm ‘shall’ kelimeleri doğru raporlanmıştır.

### 4.2.4 İngilizce Eğitimsel Oyun Projesi gereksinimlerinin gözden geçirme sonuçları

Tablo 4.5: Tiger Pro ile Eğitimsel Oyun Projesi gereksinim gözden geçirme sonuçları

Hata Tipi	Açıklama	Bulgu Sayısı
Hata Tipi 1	Bir satırda birden fazla gereksinim	0
Hata Tipi 2	Bir satırda muhtemel birden fazla gereksinim	53
Hata Tipi 3	Doğrulanamaz ve test edilemez ifadeler	29
Hata Tipi 4	Gereksinim tanımlarken hatalı kelime kullanımı	74
Hata Tipi 5	Kullanıcının tanımladığı hatalı kelimeler	0
Hata Tipi 6	Muhtemel tasarım kısıtları	1
Hata Tipi 7	Tamamlanmamış ifadeler	4
Analiz Başarı Oranı: (Doğru Bulgu/ Toplam Bulgu)		87/161

Hata tipi 2 için gözlemlenen başarı oranı 0/53'dür. “ve/veya/ikisi de/her hangi biri” ve “ / “taramaları ile bu kararın verilmesi doğru değildir.

Hata tipi 3 için gözlemlenen başarı oranı 9/29'dur. ‘Her zaman, gizli zamirler, uygun, kolay, örnek olarak’ gibi kullanımlar doğru tespit edilmiştir. Fakat “ayrıca, hepsi, hiç biri, içeren aynı’ gibi ifadelerin gereksinimlerde kullanımı hata olarak raporlanamaz. “Maksimum” kelimesi için doğrulanamaz kararı verilmiş, fakat bu değer gereksinimde tanımlandığı için yine hatalı bir karar söz konusudur.

Hata tipi 4 için gözlemlenen başarı oranı 74/74'dür. ‘shall’ yerine kullanılan tüm “must, should, will” gibi ifadeleri doğru raporlanmıştır. Fakat tüm dokümanın gereksinimlerden oluştuğu varsayımı ile ve mükerrer kayıtlarda ayrı ayrı sayılarak bu istatistiğe ulaşılmıştır.

Hata tipi 6 için gözlemlenen başarı oranı 0/1'dir. “Via” kelimesini tasarım kısıtı sayılmış ve yanlış bir varsayımdır.

Hata tipi 7 için gözlemlenen başarı oranı 4/4'dür. Eksik olan tüm ‘shall’ kelimeleri doğru raporlanmıştır.

#### 4.2.5 İngilizce Proje Yönetim Sistemi gereksinimlerinin gözden geçirme sonuçları

Tablo 4.6: Tiger Pro ile Proje Yönetim Sistemi gereksinim gözden geçirme sonuçları

Hata Tipi	Açıklama	Bulgu Sayısı
Hata Tipi 1	Bir satırda birden fazla gereksinim	41
Hata Tipi 2	Bir satırda muhtemel birden fazla gereksinim	253
Hata Tipi 3	Doğrulanamaz ve test edilemez ifadeler	104
Hata Tipi 4	Gereksinim tanımlarken hatalı kelime kullanımı	108
Hata Tipi 5	Kullanıcının tanımladığı hatalı kelimeler	0
Hata Tipi 6	Muhtemel tasarım kısıtları	0
Hata Tipi 7	Tamamlanmamış ifadeler	5
Analiz Başarı Oranı: (Doğru Bulgu/ Toplam Bulgu)		133/511

Hata tipi 1 için sadece “shall” sayısına bakılarak birden fazla gereksinim kararı verilmesi istatistikleri yanıltabilmektedir. İncelendiğinde bulguların 12 tanesi birden fazla gereksinim içerdiği bu sebeple başarı oranı 12/41 olarak yorumlanmıştır.

Hata tipi 2 için gözlemlenen başarı oranı 0/253. Bulunan bulgularda “ve/veya/ikisi de/her hangi biri” ve “ / “taramaları ile bu kararın verilmesi doğru değildir.

Hata tipi 3 için gözlemlenen başarı oranı 8/104'dür. “Elverişli, olası, yeterli, periyodik” gibi doğrulanamaz kullanımlar başarıyla tespit edilmiştir. “Ayrıca, hepsi,

hiç biri, içeren aynı” gibi ifadelerinin gereksinimlerde kullanılmasının hata olarak raporlanması uygun değildir.

Hata tipi 4 için bulgu başarı oranı 108/108’dir. ‘shall’ yerine kullanılan tüm “must, should, will” gibi ifadeleri doğru bulunmuştur. Fakat tüm dokümanın gereksinimlerden oluştuğu varsayımı ile ve mükerrer kayıtlarda ayrı ayrı sayılarak bu istatistiğe ulaşılmıştır.

Hata tipi 7 için Başarı oranı 5/5. Eksik olan tüm ‘shall’ kelimeleri doğru raporlanmıştır.

#### 4.2.6 Tiger Pro ile gözden geçirme özeti

Tiger Pro yazılımı ile tüm proje dokümanları elde edilen istatistikler Tablo 4.7’de özetlenmiştir. Gözden geçirilen 5 gereksinim dokümanından 433 gereksinim için 1081 hata raporlanmış ve bunlardan 360 tanesinin gerçekten hata olduğu gözlemlenmiştir.

Tablo 4.7: Tiger Pro ile gözden geçirme istatistikleri

Proje	Gereksinim Sayısı	Kontrol Madde Sayısı	Bulgu Sayısı	Hatalı Bulgu Sayısı	Doğru Bulgu Sayısı	Doğru Raporlama Yüzdesi
HRS	77	7	234	175	59	25,21
UKS	84	7	103	38	65	63,10
PYS	125	7	511	378	133	26,02
Eğitimsel Oyun	36	7	161	74	87	54,03
E-Market	111	7	72	56	16	22,22
<b>Toplam</b>	<b>433</b>	<b>7</b>	<b>1081</b>	<b>721</b>	<b>360</b>	<b>33,30</b>

Yedi kontrol maddesinin uygulandığı gereksinimlerde doğru hata raporlama yüzdesinin %22,22 ile %63,10 arasında değiştiği ve genel oranın ise %33,30 olarak tespit edildiği görülmektedir. Bir önceki bölümde, bu istatistikler altında yatan nedenler incelenmiş ve her kontrol maddesine özel analizler Bölüm 4.4’de tekrar özetlenecektir.

#### 4.3 Tez yazılımı OG3 ile Otomatik Gözden Geçirme

Tez çalışmaları kapsamında geliştirilen OG3 yazılımı Tablo 3.1’de verilen tüm kontrol maddeleri üzerinden Türkçe gereksinimleri otomatik olarak gözden geçirmektedir. Bu kapsamda Tiger Pro Yazılımına girdi olarak verilen 5 ayrı İngilizce gereksinim dokümanının Türkçe ’ye çevrilmiş karşılıkları ve yazım-dil

bilgisi kontrol deneyleri için ekstra bir doküman XML formatında yazılıma girdi olarak sağlanmış ve gözden geçirme sonuçları aşağıda özetlenmiştir.

#### 4.3.1 Türkçe Havayolu Rezervasyon Sistemi gereksinimlerinin gözden geçirme sonuçları

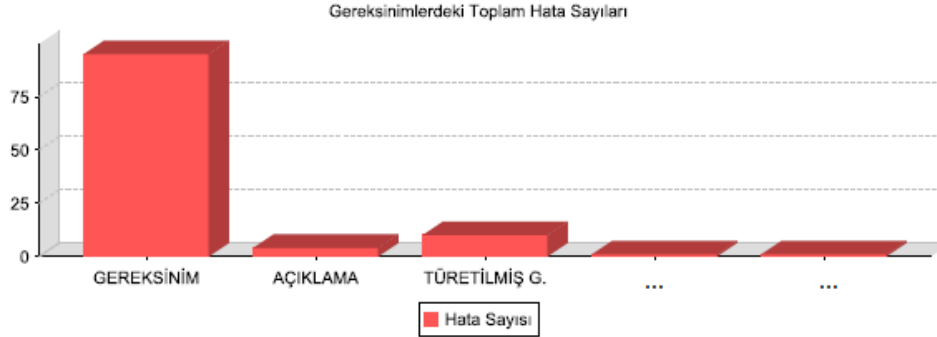
Türkçeye çevrilmiş 77 adet Havayolu Rezervasyon Sistemi gereksinim doküman nesnesinden oluşan XML formatlı proje dokümanı, gereksinim numarası, metni, tipi ve izlenebilirlik kolonları üzerinden OG3 yazılımı ile gözden geçirilmiştir. Ek E’de verilen gereksinim gözden geçirme raporlarına göre Tablo 4.8’de verilen kontrol listesi maddelerinin ihlal edildiği 111 bulgu raporlanmıştır.

**Tablo 4.8:** OG3 Yazılımı ile Havayolu Rezervasyon Sistemi otomatik gözden geçirme sonuçları (Bulgu sayısı 0 olanlar çıkartılmıştır.)

Tip	Açıklama	Bulgu sayısı
K3	Gereksinimler birbirlerinden ayırt edilebilen özgün bir tanımlayıcı numaraya sahip olmalıdır.	4
K4	Gereksinimlerde işlevi tanımlayan yüklem güçlü bir dil ile –ecek, –acak eklerini içererek tanımlanmalıdır.	30
K5	Gereksinimler pozitif bir dil ile yazılacak, negatif ifadeler içermeyecektir.	3
K6	Gereksinimlerde muğlak ifadeler kullanılmamalıdır.	20
K7	Bir gereksinim birden fazla gereksinimi içermemelidir.	4
K8	Gereksinimler okuyanı ikna edici ifadeler içermemelidir.	1
K9	Gereksinimlerde belirsiz ifadeler kullanılmamalıdır.	9
K10	Gereksinimlerde ekstra kesinlik belirten ifadeler kullanılmamalıdır.	2
K11	Gereksinimler genelleyici ifadeler içermemelidir.	3
K12	Gereksinimler göreceli, öznel ifadeler içermemelidir.	7
K13	Gereksinimler tavsiye edici ifadeler içermemelidir.	10
K14	Gereksinimlerdeki tüm liste ve maddeler eksiksiz olmalıdır.	5
K15	Gereksinim seviyeleri arasında ileri, geri izlenebilirlikler kurulu olmalıdır.	3
K16	Türetilmiş gereksinimler ve açıklamalar için izlenebilirlik oluşturulmamalıdır.	3
K17	Gereksinim numara yazım formatı projeye özel belirlenen standartlara uymalıdır.	2
K18	Gereksinim olmayan açıklama ve yorumlara proje standardına uygun belirleyici numara verilmelidir.	2
K19	Gereksinim tipleri projede belirlenen tiplere uygun atanmalıdır.	2
K20	İzlenebilirlik çekilen üst seviye gereksinim numara formatı standarda uygun olmalıdır.	1

Raporlanan hatalar ve gereksinimler detaylı incelendiğinde 111 bulgudan 10 tanesinin hata olmadığı halde hatalı olarak raporlandığı, var olan hatalardan ise 3 tanesinin yakalanamadığı gözlemlenmiştir. Dokümanda var olan diğer 101 hatanın

ise tamamı doğru raporlanmıştır. Şekil 4.1’de verilen gereksinim tipi/bulgu grafiği tüm bulunan hataların gereksinim, açıklama, türetilmiş gereksinim ve hatta hatalı girilen nesne tiplerine göre dağılımlarının gözden geçirme raporundaki gösterimidir.



Şekil 4.1: OG3 ile Havayolu Rezervasyon Sistemi gereksinimleri için gereksinim tipi/bulgu grafiği

Kontrol listesi maddelerine göre bulunan bulgular Şekil 4.2’de hata tipi/bulgu grafiğinde özetlenmiştir. Bu grafikler hata analizleri kapsamında OG3 yazılımı tarafından otomatik olarak üretilmiştir. Gereksinim dokümanında en sık yapılan üç hata; işlevi belirten yüklemelerin hatalı tanımlanması, gereksinimlerde muğlak, göreceli, öznel ifadelerin kullanılması olarak ön plana çıkmaktadır.



Şekil 4.2: OG3 ile Havayolu Rezervasyon Sistemi gereksinimleri için hata tipi/bulgu grafiği

#### 4.3.2 Türkçe E-Market gereksinimlerinin gözden geçirme sonuçları

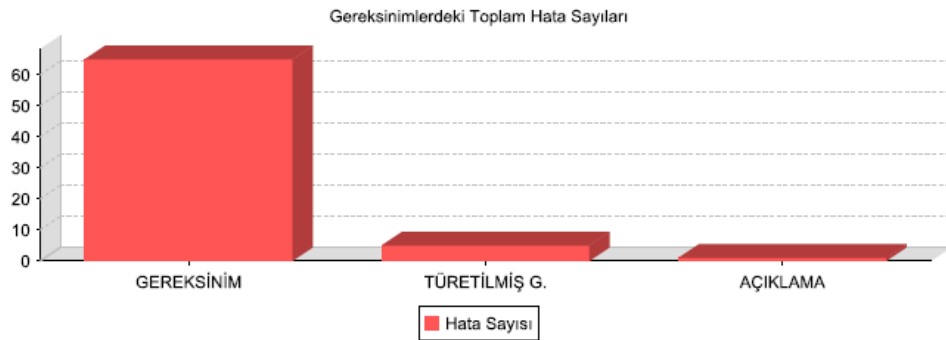
Türkçeye çevrilmiş 111 adet E-Market gereksinim doküman nesnesinden oluşan XML formatlı proje dokümanı, gereksinim numarası, metni, tipi ve izlenebilirlik kolonları üzerinden OG3 yazılımı ile gözden geçirilmiştir. Ek E’de verilen gereksinim gözden geçirme raporlarına göre Tablo 4.9’de verilen kontrol listesi maddelerinin ihlal edildiği 71 bulgu raporlanmıştır.



**Tablo 4.9:** OG3 Yazılımı ile E-Market otomatik gözden geçirme sonuçları (Bulgu sayısı 0 olanlar çıkartılmıştır.)

Tip	Açıklama	Bulgu sayısı
K1	Gereksinim yazım dili Türkçedir.	1
K3	Gereksinimler birbirlerinden ayırt edilebilen özgün bir tanımlayıcı numaraya sahip olmalıdır.	4
K4	Gereksinimlerde işlevi tanımlayan yüklem güçlü bir dil ile –ecek, –acak eklerini içererek tanımlanmalıdır.	32
K5	Gereksinimler pozitif bir dil ile yazılacak, negatif ifadeler içermeyecektir.	3
K6	Gereksinimlerde muğlak ifadeler kullanılmamalıdır.	3
K9	Gereksinimlerde belirsiz ifadeler kullanılmamalıdır.	1
K10	Gereksinimlerde ekstra kesinlik belirten ifadeler kullanılmamalıdır.	3
K11	Gereksinimler genelleyici ifadeler içermemelidir.	1
K12	Gereksinimler göreceli, öznel ifadeler içermemelidir.	2
K13	Gereksinimler tavsiye edici ifadeler içermemelidir.	14
K14	Gereksinimlerdeki tüm liste ve maddeler eksiksiz olmalıdır.	1
K15	Gereksinim seviyeleri arasında ileri, geri izlenebilirlikler kurulu olmalıdır.	2
K16	Türetilmiş gereksinimler ve açıklamalar için izlenebilirlik oluşturulmamalıdır.	1
K17	Gereksinim numara yazım formatı projeye özel belirlenen standartlara uymalıdır.	1
K18	Gereksinim olmayan açıklama ve yorumlara proje standardına uygun belirleyici numara verilmelidir.	1
K20	İzlenebilirlik çekilen üst seviye gereksinim numara formatı standarda uygun olmalıdır.	1

Raporlanan hatalar ve gereksinimler detaylı incelendiğinde 71 bulgun tamamının doğru raporlandığı, hata olmadığı halde hatalı raporlanan ekstra bir bulgu gözlemlenmemiştir. Sadece var olan 3 hatanın yazılım tarafında yakalanamadığı, bunlardan iki tanesinin İngilizce dil kullanımı ve diğerlerinin ise noktalama işareti eksikliği ve fiilimsiler sebebi ile yakalanamadığı tespit edilmiştir. Şekil 4.3 gereksinim tipi/bulgu grafiği tüm bulunan hataların gereksinim, açıklama ve türetilmiş gereksinim tiplerine göre dağılımlarının gözden geçirme raporundaki grafiksel gösterimidir.



**Şekil 4.3:** OG3 ile E-Market gereksinimleri için gereksinim tipi/bulgu grafiği

Kontrol listesi maddelerine göre bulunan bulgular Şekil 4.4 hata tipi/bulgu grafiği ile özetlenmiştir. Bu grafikler hata analizleri kapsamında OG3 yazılımı tarafından otomatik olarak üretilmiştir. Gereksinim dokümanında en sık yapılan üç

hatanın işlevi belirten yüklemelerin hatalı tanımlanması, gereksinimlerde tavsiye edici bir dilin kullanılması ve gereksinim numaralarının ayırt edici bir şekilde kullanılmaması olduğu gözlemlenmiştir.



Şekil 4.4: OG3 ile E-Market gereksinimleri için hata tipi/bulgu grafiği

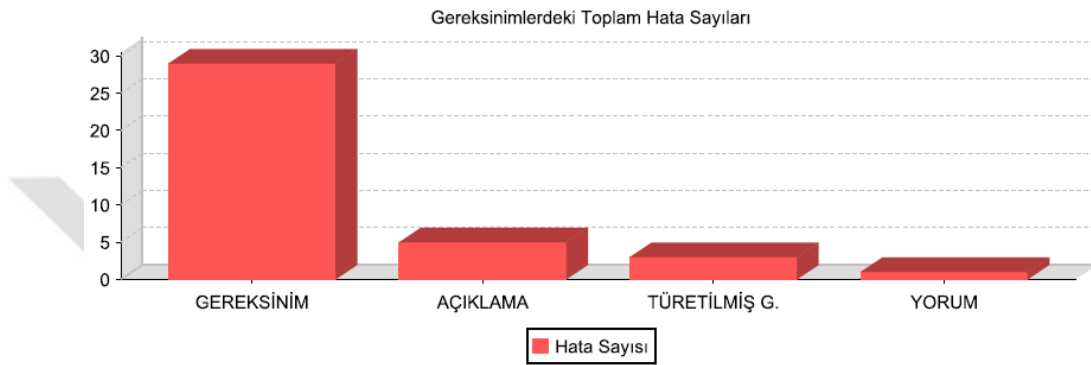
#### 4.3.3 Türkçe Uçuş Kontrol Sistemi gereksinimlerinin gözden geçirme sonuçları

Türkçeye çevrilmiş 84 adet E-Market gereksinim doküman nesnesinden oluşan XML formatlı proje dokümanı, gereksinim numarası, metni, tipi ve izlenebilirlik kolonları üzerinden OG3 yazılımı ile gözden geçirilmiştir. Ek E 'de verilen gereksinim gözden geçirme raporlarına göre Tablo 4.10'da verilen kontrol listesi maddelerinin ihlal edildiği 38 bulgu raporlanmıştır.

**Tablo 4.10:** OG3 Yazılımı ile Uçuş Kontrol Sistemi otomatik gözden geçirme sonuçları (Bulgu sayısı 0 olanlar çıkartılmıştır.)

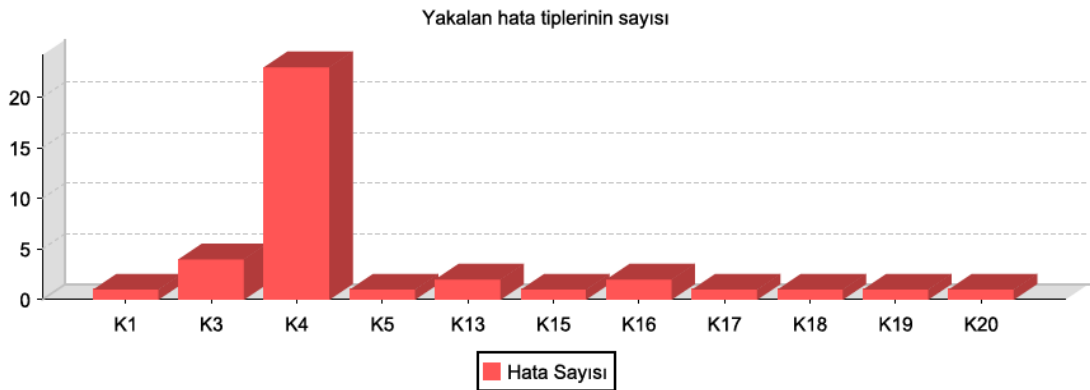
Tip	Açıklama	Bulgu sayısı
K1	Gereksinim yazım dili Türkçedir.	1
K3	Gereksinimler birbirlerinden ayırt edilebilen özgün bir tanımlayıcı numaraya sahip olmalıdır.	4
K4	Gereksinimlerde işlevi tanımlayan yüklem güçlü bir dil ile –ecek, –acak eklerini içererek tanımlanmalıdır.	23
K5	Gereksinimler pozitif bir dil ile yazılacak, negatif ifadeler içermeyecektir.	1
K13	Gereksinimler tavsiye edici ifadeler içermemelidir.	2
K15	Gereksinim seviyeleri arasında ileri, geri izlenebilirlikler kurulu olmalıdır.	1
K16	Türetilmiş gereksinimler ve açıklamalar için izlenebilirlik oluşturulmamalıdır.	2
K17	Gereksinim numara yazım formatı projeye özel belirlenen standartlara uymalıdır.	1
K18	Gereksinim olmayan açıklama ve yorumlara proje standardına uygun belirleyici numara verilmelidir.	1
K19	Gereksinim tipleri projede belirlenen tiplere uygun atanmalıdır.	1
K20	İzlenebilirlik çekilen üst seviye gereksinim numara formatı standarda uygun olmalıdır.	1

Raporlanan hatalar ve gereksinimler detaylı incelendiğinde 38 bulgunun tamamının doğru raporlandığı ve hata olmadığı halde hatalı raporlanan ekstra bir bulgu gözlemlenmemiştir. Sadece var olan 1 hatanın yazılım tarafında yakalanamadığı, bunda fiilimsi olan bir sözcük sebebi ile yüklem analizinde yaşandığı görülmüştür. Şekil 4.5 gereksinim tipi/bulgu grafiği tüm bulunan hataların gereksinim, açıklama ve türetilmiş gereksinim tiplerine göre dağılımlarının gözden geçirme raporundaki grafiksel gösterimidir.



Şekil 4.5: OG3 ile Uçuş Kontrol Sistemi için gereksinim tipi/bulgu grafiği

Kontrol listesi maddelerine göre bulunan bulgular Şekil 4.6 hata tipi/bulgu grafiğinde özetlenmiştir. Bu grafikler hata analizleri kapsamında OG3 yazılımı tarafından otomatik olarak üretilmiştir Gereksinim dokümanında en sık yapılan hataların işlevi belirten yüklemelerin hatalı tanımlanması ve gereksinim numaralarının ayırt edici bir şekilde kullanılmaması olduğu gözlemlenmiştir.



Şekil 4.6: OG3 ile Uçuş Kontrol Sistemi gereksinimleri için hata tipi/bulgu grafiği

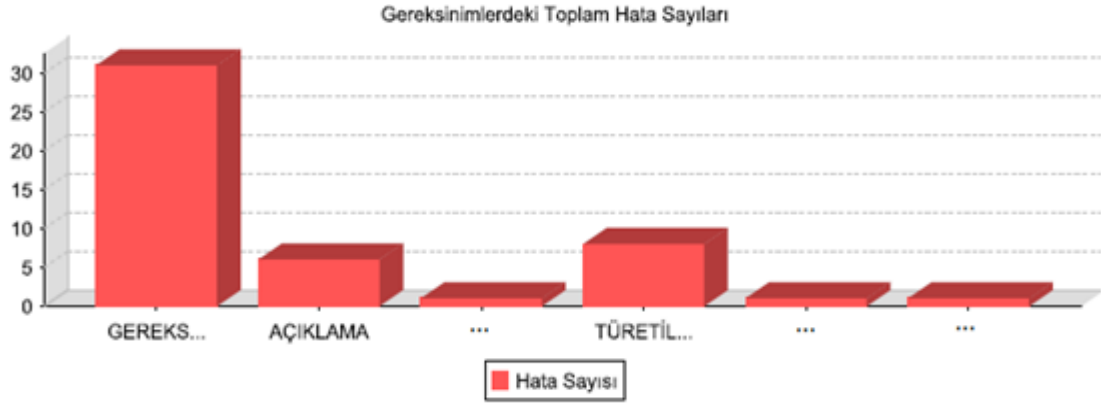
#### 4.3.4 Türkçe Eğitimsel Oyun gereksinimlerinin gözden geçirme sonuçları

Türkçeye çevrilmiş 36 adet Eğitimsel Oyun gereksinim doküman nesnesinden oluşan XML formatlı proje dokümanı, gereksinim numarası, metni, tipi ve izlenebilirlik kolonları üzerinden OG3 yazılımı ile gözden geçirilmiştir. Ek E'de verilen gereksinim gözden geçirme raporlarına göre Tablo 4.11'de verilen kontrol listesi maddelerinin ihlal edildiği 48 bulgu raporlanmıştır.

**Tablo 4.11:** OG3 Yazılımı ile Eğitimsel Oyun otomatik gözden geçirme sonuçları (Bulgu sayısı 0 olanlar çıkartılmıştır.)

Tip	Açıklama	Bulgu sayısı
K1	Gereksinim yazım dili Türkçedir.	1
K3	Gereksinimler birbirlerinden ayırt edilebilen özgün bir tanımlayıcı numaraya sahip olmalıdır.	5
K4	Gereksinimlerde işlevi tanımlayan yüklem güçlü bir dil ile –ecek, –acak eklerini içererek tanımlanmalıdır.	6
K5	Gereksinimler pozitif bir dil ile yazılacak, negatif ifadeler içermeyecektir.	1
K6	Gereksinimlerde muğlak ifadeler kullanılmamalıdır.	4
K7	Bir gereksinim birden fazla gereksinimi içermemelidir.	1
K8	Gereksinimler okuyanı ikna edici ifadeler içermemelidir.	1
K9	Gereksinimlerde belirsiz ifadeler kullanılmamalıdır.	3
K10	Gereksinimlerde ekstra kesinlik belirten ifadeler kullanılmamalıdır.	1
K11	Gereksinimler genelleyici ifadeler içermemelidir.	2
K12	Gereksinimler göreceli, öznel ifadeler içermemelidir.	4
K13	Gereksinimler tavsiye edici ifadeler içermemelidir.	1
K14	Gereksinimlerdeki tüm liste ve maddeler eksiksiz olmalıdır.	1
K15	Gereksinim seviyeleri arasında ileri, geri izlenebilirlikler kurulu olmalıdır.	2
K16	Türetilmiş gereksinimler ve açıklamalar için izlenebilirlik oluşturulmamalıdır.	3
K17	Gereksinim numara yazım formatı projeye özel belirlenen standartlara uymalıdır.	4
K18	Gereksinim olmayan açıklama ve yorumlara proje standardına uygun belirleyici numara verilmelidir.	2
K19	Gereksinim tipleri projede belirlenen tiplere uygun atanmalıdır.	3
K20	İzlenebilirlik çekilen üst seviye gereksinim numara formatı standarda uygun olmalıdır.	3

Raporlanan hatalar ve gereksinimler detaylı incelendiğinde 48 bulgunun 47 tanesinin doğru raporlandığı, sadece 1 bulgunun hatalı raporlandığı gözlemlenmiştir. Şekil 4.7 gereksinim tipi/bulgu grafiği tüm bulunan hataların gereksinim, açıklama ve türetilmiş gereksinim tiplerine göre dağılımlarının gözden geçirme raporundaki grafiksel gösterimidir.



Şekil 4.7: OG3 ile Eğitimsel Oyun gereksinimleri için gereksinim tipi/bulgu grafiği

Kontrol listesi maddelerine göre bulunan bulgular Şekil 4.8 OG3 ile Eğitimsel Oyun gereksinimleri için hata tipi/bulgu grafiğinde özetlenmiştir. Bu grafikler hata analizleri kapsamında OG3 yazılımı tarafından otomatik olarak üretilmiştir. Gereksinim dokümanında en sık yapılan hataların işlevi belirten yüklemelerin hatalı tanımlanması, gereksinim numaralarının ayırt edici bir şekilde kullanılmaması, gereksinimlerde muğlak ve göreceli öznel ifadelerin kullanılması ve gereksinim numarası yazım format hatalarının yapıldığı gözlemlenmiştir.



Şekil 4.8: OG3 ile Eğitimsel Oyun gereksinimleri için hata tipi/bulgu grafiği

#### 4.3.5 Türkçe Proje Yönetim Sistemi gereksinimlerinin gözden geçirme sonuçları

Türkçeye çevrilmiş 125 adet Proje Yönetim Sistemi gereksinim doküman nesnesinden oluşan XML formatlı proje dokümanı, gereksinim numarası, metni, tipi ve izlenebilirlik kolonları üzerinden OG3 yazılımı ile gözden geçirilmiştir. Ek E'de verilen gereksinim gözden geçirme raporlarına göre Tablo 4.12'de verilen kontrol listesi maddelerinin ihlal edildiği 195 bulgu raporlanmıştır.

**Tablo 4.12:** OG3 Yazılımı ile Proje Yönetim Sistemi otomatik gözden geçirme sonuçları (Bulgu sayısı 0 olanlar çıkartılmıştır.)

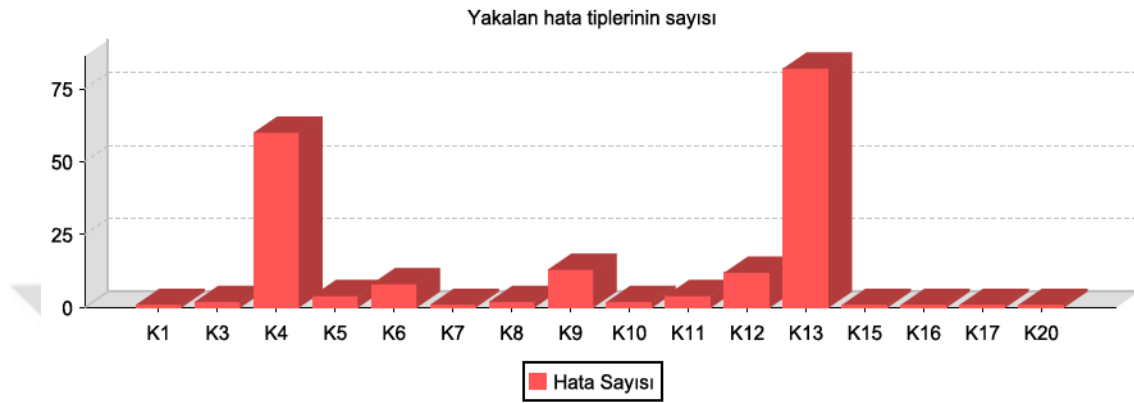
Tip	Açıklama	Bulgu sayısı
K1	Gereksinimler yazım dili Türkçedir.	1
K3	Gereksinimler birbirlerinden ayırt edilebilen özgün bir tanımlayıcı numaraya sahip olmalıdır.	2
K4	Gereksinimlerde işlevi tanımlayan yüklem güçlü bir dil ile –ecek, –acak eklerini içererek tanımlanmalıdır.	60
K5	Gereksinimler pozitif bir dil ile yazılacak, negatif ifadeler içermeyecektir.	4
K6	Gereksinimlerde muğlak ifadeler kullanılmamalıdır.	8
K7	Bir gereksinimde birden fazla gereksinimi içermemelidir.	1
K8	Gereksinimler okuyanı ikna edici ifadeler içermemelidir.	2
K9	Gereksinimlerde belirsiz ifadeler kullanılmamalıdır.	13
K10	Gereksinimlerde ekstra kesinlik belirten ifadeler kullanılmamalıdır.	2
K11	Gereksinimler genelleme ifadeler içermemelidir.	4
K12	Gereksinimler göreceli, öznel ifadeler içermemelidir.	12
K13	Gereksinimler tavsiye edici ifadeler içermemelidir.	82
K15	Gereksinim seviyeleri arasında ileri, geri izlenebilirlikler kurulu olmalıdır.	1
K16	Türetilmiş gereksinimler ve açıklamalar için izlenebilirlik oluşturulmamalıdır.	1
K17	Gereksinim numara yazım formatı projeye özel belirlenen standartlara uymalıdır.	1
K20	İzlenebilirlik çekilen üst seviye gereksinim numara formatı standarda uygun olmalıdır.	1

Raporlanan hatalar ve gereksinimler detaylı incelendiğinde 195 bulgudan 6 tanesinin hata olmadığı halde hatalı olarak raporlandığı, var olan hatalardan ise sadece 3 tanesinin yakalanamadığı gözlemlenmiştir. Dokümanda var olan diğer 189 hatanın ise tamamı doğru raporlanmıştır. Şekil 4.9’da verilen gereksinim tipi/bulgu grafiği tüm bulunan hataların gereksinim, açıklama, türetilmiş gereksinim ve hatta hatalı girilen nesne tiplerine göre dağılımlarının gözden geçirme raporundaki grafiksel gösterimidir.



**Şekil 4.9:** OG3 ile Proje Yönetim Sistemi gereksinimleri için gereksinim tipi/bulgu grafiği

Kontrol listesi maddelerine göre bulunan bulgular Şekil 4.10 hata tipi/bulgu grafiğinde özetlenmiştir. Bu grafikler hata analizleri kapsamında OG3 yazılımı tarafından otomatik olarak üretilmiştir. Gereksinim dokümanında en sık yapılan hataların tavsiye edici dil kullanımı, işlevi belirten yüklemelerin hatalı tanımlanması, belirsiz ve göreceli öznel ifadelerin kullanılması olduğu gözlemlenmiştir.



Şekil 4.10: OG3 ile Proje Yönetim Sistemi gereksinimleri için hata tipi/bulgu grafiği

#### 4.3.6 Türkçe Dokümanik Gazete gereksinimlerinin K2 kontrol maddesine özel gözden geçirmesi

Türkçe dilinde yazılmış dokümanlar üzerinde dilbilgisi kurallarına uygunluk ve yazım hatası kontrolü popüler kelime işleme programları tarafından yapılabilmektedir. Fakat gereksinim yönetim yazılım araçları üzerinde Türkçe 'ye özel bu kontrollerin yapılamaması, ek olarak yazılım ve sistem gereksinimlerinin Türkçe yazılmasına rağmen içerisinde birçok İngilizce ve Türkçe karşılığı olmayan teknik terim ve kısaltma içermesi bu kontrolleri zorlaştırmaktadır. Gereksinim gözden geçirme sürecinde, zaman sıkıntısı sebebi ile hızla hazırlanan, yazım ve dilbilgisi hataları içeren, okunaklılarını kaybeden dokümanların önden otomatik olarak kontrol edilerek düzeltilmesi ve devamında resmi gözden geçirme sürecine dâhil edilmesi önemli bir ihtiyaçtır. Bu kapsamda Türkçe gereksinimlerin dilbilgisi kurallarına uygun olup olmadığının kontrolü ve yazım hatalarının tespiti amacı ile OGGKL'ye K2 maddesi eklenmiştir. Türkçe gereksinimlerde kullanılan özel kısaltmalar, İngilizce özel isim veya teknik kelime kullanımları ve diğer yazım hatalarının otomatik olarak değiştirilmesi diğer kontrol listesi algoritmalarına dolaylı etkisi olabileceğinden bu madde bir ön kontrol maddesi olarak ayrıca

çalıştırılmaktadır. Toplam 93 gereksinim içeren Dokunmatik Gazete proje gereksinimleri K2 maddesi ile kontrol edilmiş ve sonuçları aşağıda özetlenmiştir.

**Tablo 4.13:** OG3 yazılımı ile Dokunmatik Gazete proje gereksinimleri için K2 ile gözden geçirme sonuçları

Proje	Kelime sayısı	Gereksinim Sayısı	Kontrol Madde Sayısı	Toplam Hata Sayısı	Düzeltilme Sayısı	Hatalı Düzeltilme Sayısı	Bozma Sayısı	Bozma %	Başarı %
Dokunmatik Gazete	1426	93	1	37	35	2	7	0,49	94,59

1426 kelimededen oluşan 93 gereksinim içerisinde var olan 37 adet yazım hatasının 35 tanesinin OG3 yazılımı ile doğru olarak düzeltildiği gözlemlenmiştir. Sadece 2 hatanın düzeltilemediği ve hata olmadığı halde 7 kelimenin ise yazılım tarafından bozulduğu gözlemlenmiştir.

#### 4.3.7 OG3 ile gözden geçirme özeti

Tezin için geliştirilen OG3 yazılımı ile 6 gereksinim dokümanından 5'i üzerinde 19 kontrol maddesi için analiz yapılmış, diğer doküman için ise özel olarak K2 kontrol maddesi analizleri gerçekleştirilmiştir. Toplam 433 gereksinim için 463 hata raporlanmış, bunlardan 446 tanesinin gerçekten hata olduğu tespit edilmiştir. 17 gereksinim için hata olmadığı halde hata raporlanmış ve var olan 11 adet gerçek hatanın ise yakalanamamıştır.

**Tablo 4.14:** OG3 yazılımı ile gözden geçirme istatistikleri

Proje	Gereksinim Sayısı	Kontrol Madde Sayısı	Toplam Hata Sayısı	Bulgu Sayısı	Hatalı Bulgu Sayısı	Doğru Bulgu Sayısı	Kaçırma Sayısı	Başarı %	Kaçırma %
HRS	77	19	104	111	10	101	3	97,11	2,88
UKS	84	19	39	38	0	38	1	97,43	2,56
PYS	125	19	192	195	6	189	3	98,43	1,56
Eğitimsel Oyun	36	19	47	48	1	47	0	100	0
E-Market	111	19	74	71	0	71	3	95,94	4,05
<b>Toplam</b>	<b>433</b>	<b>-</b>	<b>456</b>	<b>463</b>	<b>17</b>	<b>446</b>	<b>10</b>	<b>97,80</b>	<b>2,19</b>

Projelere ait başarı ve hata kaçırma istatistiklerinin verildiği Tablo 4.14'de, tüm gözden geçirmeler için hata raporlama başarı oranı % 97,80 ve hata kaçırma oranı ise %2,19 olarak hesaplanmıştır. OG3 yazılımı ile yakalanan başarılı istatistiklerde, standartlar ışığında oluşturulan OGGKL'nin geniş kapsam ve doğruluğu, ayrıca



yazılımda gözden geçirilen gereksinimlerin tiplerinin algılanarak, tiplere uygun farklı analiz algoritmalarının çalıştırılmasının önemli etkileri olduğu görülmüştür.

#### 4.4 Karşılaştırma

Deneyler kapsamında aynı test ortamında çalıştırılan Tiger Pro ve OG3 yazılımı ile gözden geçirmeler sonucunda elde edilen genel istatistikler Tablo 4.15’de özetlenmiştir. Altı gereksinim dokümanından beşi her iki yazılımda da gözden geçirilmiş, Dokunmatik Gazete gereksinimleri ise yazım ve dilbilgisi hata kontrolü Tiger Pro yazılımında bulunmadığı için, K2 maddesine özel şekilde bu kontrol sadece OG3 yazılımı ile gerçekleştirilmiştir.

**Tablo 4.15:** Gözden geçirme istatistikleri

Proje	Gereksinim Sayısı	OG3 Kontrol Madde Sayısı	Tiger Pro Kontrol Madde Sayısı	OG3 Gözden geçirme Süresi	Tiger Pro Gözden geçirme Süresi
HRS	77	19	7	21 dk 3 sn	<10 saniye
UKS	84	19	7	11 dk 21 sn	<10 saniye
PYS	125	19	7	39 dk 13 sn	<10 saniye
Eğitimsel Oyun	36	19	7	8 dk 37 sn	<10 saniye
E-Market	111	19	7	20 dk 16 sn	<10 saniye
Dokunmatik Gazete	93	1	0	27 sn	-

Var olan 433 gereksinim üzerinde OG3 yazılımı 19 madde için uyumsuzluk kontrolü yaparken, Tiger Pro yazılımı 7 madde için analizlerini gerçekleştirmiştir. Herhangi bir başka sunucu veya ağ gibi dış kaynağa erişim ihtiyacı olmayan, lokal ve statik kütüphanelerini kullanan Tiger Pro yazılımı analizlerini 10 saniyeden daha kısa sürelerde tamamlamıştır. OG3 yazılımı ise 20 algoritmadan 6’sı için İTÜ doğal dil işleme web servisi algoritmalarından faydalanmıştır. Bu algoritmalar için kimi zaman tek bir kelime için, kimi zaman da birden fazla gereksinim cümlesi için zincirin tüm algoritmalarının arka arkaya çalıştırıldığı bir zincir şeklinde HTTP sorguları ile çalıştırılmıştır. Her bir sorgu tüm Türkiye’den akademik çalışmalar amacı ile gönderilen sorgulara arasında sıraya girerken, sonuçlar için sunucu yoğunluğuna bağlı olarak saniyeler mertebesinde bekleme süreleri oluşmaktadır. Bu sebeple OG3 yazılımı ile gözden geçirme süreleri dakikalar mertebesinde uzun sürmektedir. Eğer İTÜ doğal dil işleme zincir algoritmaları için statik bir kütüphane kullanabilirse analiz süreleri saniyeler mertebesinde gerçekleşebileceği öngörülmektedir.

**Tablo 4.16:** Tiger Pro yazılımı doğru bulgu istatistikleri

Kontrol Maddesi	Eğ. Oyun	EMarket	HRS	PYS	UKS	Toplam
Hata 1	0	1	4	12	0	17
Hata 2	0	1	0	0	0	1
Hata 3	9	4	8	8	0	29
Hata 4	74	8	24	108	30	244
Hata 5	0	0	0	0	0	0
Hata 6	0	0	0	0	0	0
Hata 7	4	2	23	5	35	69
Toplam	87	16	59	133	65	360

Tablo 4.16’da Tiger Pro yazılımı ile gözden geçirmeler sonucunda elden edilen doğru bulgu istatistikleri verilmiştir. İstatistikler incelendiğinde her doküman için raporlanmış olan ve toplam 360 bulgudan 244 tanesini, hatalı kelime kullanımı olarak tanımlanan fakat temelde kesinlik ifade eden “*shall*” kelime taramalarından oluşan hata tipi 4 oluşturmaktadır. Benzer şekilde, eksik “*shall*” taramaları yapan tamamlanmamış ifade analizi olan hata tipi 7 de en yüksek ikinci raporlanan hata tipi olmuştur. Muğlak ifadeler analizi içeren ve doğrulanamaz kelime kullanımı olarak tanımlanan hata tipi 3 için de üçüncü en yüksek rakam olan 29 adet hata raporlanmıştır.

Bir satırda birden fazla gereksinim analizi yapan ilk hata tipi için 17 adet hata raporlanmıştır. Birden fazla çıktıya sahip olan gereksinimlerde birden fazla “*shall*” kullanılabilceği için bu analizde sadece “*shall*” taramanın hatalı bir varsayım olduğu düşünülmektedir. İkinci hata tipi; bir satırda muhtemelen birden fazla gereksinim analizinde sadece 1 bulgu yakalanmasının temel sebebi, “*and*”, “*or*”, “*both*” ve “*either*” gibi gereksinimlerde kullanılması çok muhtemel olan ifade ve koşul bağlaçlarının aranmasıdır. 5. hata tipi kullanıcı tarafından tanımlanabilecek yasaklı kelimeler listesi olarak ayrılmış, 6. hata tipinde ise tasarım kısıtı veya tasarımla ilgili kelimelerin taranması hedeflenmiş ve herhangi bir bulgu raporlanmamıştır. Fakat teknik ve tasarım detayı içeren terimlerin gereksinimlerin seviyesine bağlı olarak dokümanda içerilmesi ihtiyaç halinde doğaldır ve hata olarak raporlanması uygun değildir. Ayrıca dar bir analiz kontrol listesine sahip olan Tiger Pro yazılımının izlenebilirlik, yazım formatı, dilbilgisi veya yazım hataları gibi analizler yapamadığı, hata ile aynı obje numarasına sahip iki gereksinim içeren bir dokümanı çözümleyemeyip, analizi hiç gerçekleştiremediği de testlerde gözlemlenen diğer durumlardır.

OG3 yazılımı ile OGGKL'ye özel tüm gereksinim dokümanları için raporlanmış hata istatistikleri Tablo 4.17'de verilmiştir. İstatistikler incelendiğinde 446 doğru bulgunun 149 tanesinin gereksinimlerde işlevi tanımlayan yüklemün güçlü bir dil ile ifade edilmesini kontrol eden K4 kontrol maddesinin oluşturduğu görülmüş ve bu veriler Tiger Pro yazılımının İngilizce dokümanlar için en fazla raporlanan hata tipi 4 ve 7 sonuçları ile örtüştüğü gözlemlenmiştir. Dokümanın hem Türkçe hem de İngilizce versiyonlarında işlevi tanımlayan yüklemlerde standartlar ile uyumsuz kullanımlar tespit edilmiştir.

**Tablo 4.17:** OG3 yazılımı doğru bulgu istatistikleri

Kontrol Maddesi	Eğ. Oyun	EMarket	HRS	PYS	UKS	Toplam
K1	1	1	0	1	1	4
K3	5	4	4	2	4	19
K4	6	32	28	60	23	149
K5	1	3	3	4	1	12
K6	4	3	18	7	0	32
K7	1	0	4	1	0	6
K8	1	0	1	2	0	4
K9	3	1	5	9	0	18
K10	1	3	2	2	0	8
K11	2	1	1	4	0	8
K12	3	2	7	11	0	23
K13	1	14	10	82	2	109
K14	1	1	5	0	0	7
K15	2	2	3	1	1	9
K16	3	1	3	1	2	10
K17	4	1	2	1	1	9
K18	2	1	2	0	1	6
K19	3	0	2	0	1	6
K20	3	1	1	1	1	7
Toplam	47	71	101	189	38	446

En sık raporlan diğer madde ise gereksinimlerde tavsiye edici ifade dil kullanımını engelleyen K13 maddesidir. İlgili proje paydaşlarında gereksinimlerin kimi zaman “*olursa iyi olur*” bir özellik gibi algılanmasına sebep olan bu dil kullanımını için 109 hata raporlanmıştır. Üçüncü en sık raporlanan hata ise 32 adet bulgu ile K6 muğlak ifade analiz kontrol maddesidir. Raporlanan diğer maddelere baktığımızda, her madde için hata raporlanabildiği ve literatürdeki diğer çalışmalardan farklı olarak gerçekleştirilen Türkçe dil kontrolü, yazım ve dilbilgisi

hata düzelticisi, birden fazla gereksinim analizi ve izlenebilirlik gibi kontrollerin yapılabildiği gözlemlenmiştir. Yakalanamayan veya hatalı raporlara bulgular incelendiğinde; gereksinim dil kararı veren K1 maddesinde İngilizce Türkçe kelimelerin karışık kullanıldığı gereksinimlerde problem yaşandığı görülmüştür. Birden fazla gereksinim analizi yapılan K7 maddesinde kaçırılan durumlar için, koşulların netçe ifade edilmediği, üst seviyeden genel ifadeler ile yazılan gereksinimler ön plana çıkmıştır. K4 ve K13 gibi yüklem analizleri içeren gereksinimlerde de fiilimsilerin ve eksik noktalama işaretlerinin zaman zaman yanıltıcı olduğu tespit edilmiştir.

Tez kapsamında yapılan deneylerin her iki yazılım için ana hataları ile özetlendiği istatistikler ve yazılım özellikleri Tablo 4.18’de verilmiştir. Gözden geçirilen gereksinim sayısı, uygulanan kontrol maddesi, doğru ve hatalı bulgu sayısının verildiği tabloda, bir gereksinimde birden fazla aynı hata tipi için raporlamayı ifade eden mükerrer bulgu sayma, izlenebilirlik kontrolleri, yazım hata düzelticisi ve raporlama formatlarına kadar yazılım özellikleri karşılaştırılmıştır.

**Tablo 4.18:** Yazılım Karşılaştırma Tablosu

	<b>Tiger Pro</b>	<b>OG3</b>
Gereksinim sayısı	433	433
Kontrol madde sayısı	7	19
Hata raporlanabilen madde sayısı	5	19
Bulgu sayısı	1081	463
Doğru bulgu sayısı	360	446
Mükerrer bulgu sayma	VAR	YOK
Hatalı bulgu sayısı	721	17
Doğru raporlama yüzdesi	33,30	96,32
Gereksinim başına hata bulma	1,20	0,97
En sık raporlanan hata tipi	4, 7, 3	K4, K13, K6
İzlenebilirlik kontrolü	YOK	VAR
Yazım hata düzelticisi	YOK	VAR
Gereksinim tipi tanıma	YOK	VAR
Otomatik raporlama	VAR ( <i>txt</i> )	VAR ( <i>pdf</i> )

Daha önce Tablo 4.15’de özetlenen zamanlama istatistiklerinde OG3 yazılımına kıyasla Tiger Pro yazılımının hızlı analizler gerçekleştirdiği gözlemlenirken, Tablo 4.18’de özetlenen sonuçlarda ise OG3 yazılımının Tiger Pro yazılımına karşı ciddi üstünlükleri ön plana çıkmıştır. Tez kapsamında yapılan

deneysel alıřmalarda performans bir ama ve bařarı kriteri olarak belirlenmediđi gibi, Tablo 4.18’de verilen dođru bulgu istatistikleri, kontrol listesi kapsamı, dođru raporlama istatistikleri ve gereksinim tanıma gibi zellikler ise tezde hedeflenen bařarılı otomatik gereksinim gzden geirme srecini tam anlamıyla belirleyen kriterlerdir. Ayrıca yazılımlar arası farkların listelendiđi Tablo 4.18’de en sık raporlanan hata istatistikleri olarak Tiger Pro yazılımında 4, 7, 13 ve OG3 yazılımında ise K4, K13 ve K6 hataları n plana ıkmıřtır. Bu sonular gereksinimlerde gl ve net dil kullanım ihlallerini ve muđlak ifadeleri iřaret ederken, farklı dillerde de olsa ortak kullanılan veri setlerinin de gvenilirliđi adına nemli birer gstergedir.



## BEŞİNCİ BÖLÜM

### SONUÇ

Yazılım geliştirme yaşam döngüsü içerisinde diğer süreçlere ve projenin başarısına doğrudan etkisi olan gereksinim yönetim süreci, barındırdığı riskler, yapılan hatalar, proje istatistikleri, kullanılan gereksinim yönetim araçları ve sürece yön veren kabul görmüş standartlar incelenerek, gereksinim hatalarının en erken safhada ayıklanabilmesi adına gereksinim gözden geçirme sürecinin otomasyon ile iyileştirilmesi hedeflenmiştir. Bu kapsamda otomatik gereksinim gözden geçirme, doğal dil işleme, İngilizce ve Türkçe gereksinimler üzerine yapılan çalışma ve geliştirilen yazılımlar üzerinde detaylı bir literatür taraması yapılarak Türkçe gereksinimler üzerinde otomatik gözden geçirme sürecinin ihtiyaç, kısıt ve hedefleri belirlenmiştir. Teze özel olarak, gereksinimler ve gözden geçirme süreci üzerine geniş bir literatür taraması yapılarak geniş bir Türkçe otomatik gereksinim gözden geçirme kontrol listesi oluşturulmuş ve bu liste Java dilinde geliştirilmiş olan gözden geçirme yazılımının algoritmalarını tanımlamıştır.

Deneyler kapsamında veri seti olarak farklı sektörlerden 6 ayrı proje gereksinim dokümanı belirlenmiş, İngilizce versiyonları literatürde önemli bir yeri olan Tiger Pro yazılımına, makine çeviriciler ile Türkçe 'ye çevrilmiş versiyonları ise teze özel olarak geliştirilen OG3 Türkçe gereksinim gözden geçirme yazılımına girdi olarak verilmiştir. Farklı sektörlerden toplam 526 adet gereksinimden oluşan veri setinin zenginliği, tezde ortaya konulan OGGKL'nin ve ilgili algoritmaların uygulanabilirliğini de gösterme açısından önemli rol oynamıştır.

Tez kapsamında geliştirilen OG3 yazılımı, deneysel amacı ile karşılaştırılan Tiger Pro yazılımından ve literatürde ortaya konulan diğer çalışmalardan belirli özellikleri ile ayrılmaktadır. Herhangi bir gereksinim yönetim aracında dışa aktarılan XML formatlı gereksinim dokümanlarını girdi olarak alabilen OG3 yazılımı, analiz algoritmalarının temelini oluşturan 20 ayrı maddeden oluşan geniş bir gözden

geçirme kontrol listesine sahiptir. Analizler sırasında diğer yazılımlardan farklı olarak, gereksinim dokümanındaki tüm nesnelere tek tipte algılayamayıp, gereksinim, türetilmiş gereksinim ve her türlü bilgilendirici amaçla oluşturulan açıklama ve yorum nesnelere çözümleyip üzerlerinde farklı kontrol maddelerini çalıştırmaktadır. Ayrıca diğer araçlarda rastlanmayan, Türkçe dil kontrolü, dilbilgisi ve yazım hata düzeltici kontrol maddeleri; dokümanların standartlara uygunluk ve anlaşılabilirlik kontrollerini otomatikleştirmektedir. Türkçe gereksinimlerde muğlak, doğrulanamaz, göreceli, net olmayan, belirsiz veya eksik ifadeler için detaylı sözlüklere sahip olan yazılımın bir diğer güçlü yönü ise en güncel Türkçe doğal dil işleme algoritmalarından olan İTÜ doğal dil işleme yazılım zincirinden faydalanmasıdır. Diğer dillere kıyasla sondan eklemeli bir dil olan Türkçe’de zor olan ve gereksinimlerde işlevi, girdileri ve çıktıları belirten koşul ve yüklem analizleri gereksinimler için başarı ile uygulanmış ve birden fazla gereksinim analizi için yeni bir koşul-yüklem analiz yaklaşımı sunulmuştur. OG3 yazılımında önde çıkan bir diğer özellik ise izlenebilirlik kontrollerinin yanı sıra, gereksinimlerin ve izlenebilirlik çekilen diğer gereksinimlerin nesne numaralarının proje standartlarına uygun ve takip edilebilir özgün bir şekilde oluşturulmasının otomatik kontrolüdür. Projelerde bu tip kontroller için gereksinim yönetim aracı haricinde ekstra yazılım ve araç ihtiyaçları ortaya çıkmakta, kimi zamanda gözden geçirme sırasında manuel olarak kontrol edilmesi beklenmektedir. Gözden geçirme sonucunda, raporları hem gereksinim hem de hata tipleri özelinde PDF formatında üreten OG3 yazılımı, süreçlere uygun şekilde denetim otoritelerine sunulabilecek delil ve kayıtları otomatik olarak hazırlamaktadır.

Tez kapsamında yapılan deneylerde OG3 yazılımı tarafından yakalanamayan veya hatalı raporlanan bulgular incelenerek zayıf ve geliştirilebilecek yönleri de ortaya çıkarılmıştır. Gereksinim dili Türkçe olsa da projelerde sıklıkla kullanılmak durumunda olan İngilizce kelimeler, teknik terimler veya kısaltmalar Türkçe dilinde gereksinim analizini zorlaştıran faktörler olarak ortaya çıkmıştır. Bu tip teknik terim, kısaltma veya İngilizce kelimeler için ek sözlüklerden faydalanılarak bu kelimelerin analiz kapsamı dışında tutulması ve yazım hata düzeltimi sırasında korunması ileride eklenebilecek özellikler olarak ön plana çıkmaktadır. Ayrıca birden fazla gereksinim tespitinde girdi ve çıktıların; koşullar ve gerçekleştirilecek işlevler olarak netçe ayrıldığı, detay verilmeden üst seviyeden ifade edilen gereksinim cümlelerinde de

analizin zorlaştığı tespit edilmiştir. Yükleme analizlerinde filimsilerin ve eksik noktalama işaretlerinin, muğlak ve netliği bozan ifade analizlerinde de birden fazla anlamlı kelimeler veya referans ya da örnek verme amaçlı kullanılan bazı ifadelerin aynı gereksinim içerisinde açıklanması analizlerde yanıltıcı etkileri olmuştur. Fakat bu tip yanıltıcı durumların ileriki çalışmalarda detaylı bir araştırma ile belirlenip, doğal dil işleme algoritmalarında ve kullanılan ifade sözlüklerinde yapılacak iyileştirmeler ile giderilebileceği öngörülmektedir.

Türkçe gereksinimler üzerinde otomatik gözden geçirmenin doğal dil işleme yaklaşımları ile mümkün olabileceğini ortaya koyan bu çalışmadan sonra elde edinilen tecrübe ve kazanımlar ile gereksinim gözden geçirme sürecine otomasyon anlamında katkı yapabilecek yeni hedefler belirlenmiştir. İlk olarak kontrol maddeleri için raporlanan ihlallerin çözümünü kolaylaştıracak düzeltici, yönlendirici ve tavsiye edici aksiyonların otomatik olarak gerçekleştirilmesi hedeflenmiştir. Bu sayede gözden geçirme raporu sonucunda kullanıcıya sadece hata raporlanmayacak, hata çözümü için yol gösterici ve belki de otomatik düzeltici yaklaşımlar sunulacaktır. Ayrıca kullanılan doğal dil işleme algoritmaları için statik kütüphaneler kullanılarak analiz performansının da artırılması hedeflenmektedir. Bu yaklaşımlara ek olarak gereksinimlerin anlamsal ve mantıksal analizleri de yeni çalışmada amaçlardan biri olarak belirlenmiştir. Gereksinim dokümanları üzerinde uyumluluk, benzerlik, kapsama oranları veya çelişen noktalar özelinde otomatik analizler hedeflenmektedir. Bu analizler aynı gereksinim dokümanı içerisinde gereksinimler arasında, farklı gereksinim dokümanları arasında ve gereksinimler ile referans verilen gereksinimler arasında birebir kontroller şeklinde yapılabilir. Doğal dil işleme algoritmaları ile gerçekleştirilebilecek olan bu çalışma sayesinde, gereksinimlerin doğruluğu ve tutarlılığı kontrol edilebilirken, işlevi tanımlayan tasarım ve algoritmalarda bulunan muhtemel çelişki, uyumsuzluk ve hataların ayıklanması mümkün kılınabilir.



## KAYNAKLAR

- [1] Altalbe, A. (2015). Software Requirements Management. *International Journal of Advanced Research in Artificial Intelligence*. 4.10.14569/IJARAI.2015.040410.
- [2] Barki, H., Rivard, S., Talbot, J. (1993). Toward an Assessment of Software Development Risk, *Journal of Management Information Systems*, Vol. 10, No. 2, 1993, pp. 203-225.
- [3] Beatty, J., Ferrari, R., (2011). How to Evaluate and Select a Requirements Management Tool. Erişilebilir adres: [http://www.seilevel.com/wpcontent/uploads/RequirementsManagementToolWhitepaper\\_1.pdf](http://www.seilevel.com/wpcontent/uploads/RequirementsManagementToolWhitepaper_1.pdf)
- [4] Firesmith, D. (2007). Common Requirements Problems, Their Negative Consequences, and the Industry Best Practices to Help Solve Them. *Journal of Object Technology*. 6. 17-33. 10.5381/jot.2007.6.1.c2.
- [5] Hafeez, MS., Farhan, R., Khan MR. (2017). An Improved Model for Requirement Management System. *J Inform Tech Softw Eng* 7: 196. doi: 10.4172/2165-7866.1000196.
- [6] Mathiassen, L., Saarinen, T., Tuunanen, T., Rossi, M. (2008). Managing Requirements Engineering Risks: an Analysis and Synthesis of the Literature All Sprouts Content. 82. [http://aisel.aisnet.org/sprouts\\_all/82](http://aisel.aisnet.org/sprouts_all/82)
- [7] Mogyorodi, G., (2001). Requirements-Based Testing: An Overview, Starbase Corporation, IEEE.
- [8] Nigam, A., Nigam, B., Bhaisare, C., Arya N. (2012). Classifying the bugs using multi-class semi supervised support vector machine. 393-397. 10.1109/ICPRIME.2012.6208378.
- [9] Wiegers, K.E. (2001). Inspecting Requirements, StickyMinds.com Weekly Column, Erişim: <http://www.stickyminds.com>
- [10] Davis, G. B. (1982). Strategies for information requirements determination. *IBM Systems Journal*, Vol 21, no. 1, pp. 4-30.
- [11] Nigam, A., Arya, N., Nigam, B., Jain, D. (2012). Tool for Automatic Discovery of Ambiguity in Requirements. *IJCSI*, Vol 9, Issue 5, No 2.
- [12] Jonette M., et al, (2004). Error Cost Escalation Through the Project Life Cycle, NASA Johnson Space Center and INCOSE.

- [13] Ellis, K. (2008). "Business analysis benchmark. The impact of business requirements on the success of technology projects." *IAG Consulting*. Erişim: <http://www.iag.biz/>
- [14] NASA Requirements Peer Review Checklist, (2012), SWEREF-449, 580-CK-057-02, Software Engineering Division (SED), NASA Goddard Space Flight Center (GSFC), 2012
- [15] NASA SWEHB Software Engineering Handbook, NASA-HDBK-2203. NASA START. Erişim: <https://swehb.nasa.gov>.
- [16] Thayer R.H., Dorfman M., eds., (1977) *Software Requirements Engineering*, Second Edition, pp. 7-22. Los Alamitos, Calif.: IEEE Computer Society Press.
- [17] IEEE Std 830-1993: (1993, 2 Aralık) Recommended Practice for Software Requirements Specifications, *The Institute of Electrical and Electronics Engineering*.
- [18] ISO/IEC 9126: Software Product Evaluation. (1991), *The International Organization for Standardization and The International Electrotechnical Commission*.
- [19] Aydın, A.O., Kutay, F., Gökçen, H. ve Güngör, A. (2006). ISO/IEC 9126 Standardında Anadilin Kullanımı. s 221-237, *Journal of Yasar University*.
- [20] RTCA, "Software Considerations in Airborne Systems and Equipment Certification", 1 Aralık 1992.
- [21] Mich, L., et al, (2004) Market research for requirements analysis using linguistic tools, *Springer, Requirements Engineering*,
- [22] Oflazer, K. and Bozsahin, C., (1994). Türkçe Doğal Dil işleme, Natural Language Processing in Turkish, *Proceedings of Turkish Informatics Society TBD'94*.
- [23] Adalı, E . (2012). Doğal Dil İşleme. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 5 (2), Erişim: <http://dergipark.gov.tr/tbbmd/issue/22245/238797>
- [24] Wilson, W.M., Rosenberg, L.H. and Hyatt, L.E. (1997). Automated Analysis of Requirement Specifications. *Proceedings of the 19th International Conference on Software Engineering*.
- [25] Lami, G., (2005). "QuARS: A Tool for Analyzing Requirements", Technical Report: CMU/SEI-2005-TR-014, ESC-TR-2005-014.
- [26] Leveraging Natural Language Processing in Requirements Analysis, Erişim Tarihi: 20 Aralık 2017, <https://qracorp.com/nlp-requirements-analysis>
- [27] Arendse, B. (2016), A thorough comparison of NLP tools for requirements quality improvement, Master thesis, Faculty of Science Theses,

- [28] RQA Requirements Quality Analyzer, The Reuse Company, Erişim Tarihi: 25 Aralık 2017, [http://www.reusecompany.com/index.php?option=com\\_content&view=category&layout=blog&id=171&Itemid=75&lang=en](http://www.reusecompany.com/index.php?option=com_content&view=category&layout=blog&id=171&Itemid=75&lang=en),
- [29] Kasser, J. (2006), Tool to Ingest and Elucidate Requirements (TIGER) Pro Version.
- [30] Brooks, A. , Laura K., Paulsen, B. (2014), Beta-testing a requirements analysis tool. *ACM SIGSOFT Software Engineering Notes*. 39. 1-6. 10.1145/2659118.2659128.
- [31] Kızılca H., Yılmaz E. (2008), Doğal Dilde Yazılmış Gereksinimlerin Analiz Yöntemleri ve bu Yöntemlerin Türkçe için Uygulanabilirliği, *Yazılım Kalitesi ve Yazılım Geliştirme Araçları Sempozyumu*, Bildiriler Kitabı, s. 69-77, İstanbul.
- [32] Yılmaz, İ.B. (2010), Doğal dilde yazılmış yazılım gereksinimlerinin kalitelerinin dilbilimsel analiz teknikleri ile artırılması, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Ankara Üniversitesi, Ankara.
- [33] Eryigit, G. (2004), ITU Turkish NLP web service. *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Gothenburg, İsveç. Association for Computational Linguistics.
- [34] Şahin M. (2013), Itumorph - Türkçe İçin Daha Geniş Kapsamlı Ve Başarılı Bir Biçimbilimsel Çözümleyici, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, İstanbul Teknik Üniversitesi.
- [35] Liddy, E.D. (2001). *Natural Language Processing*. In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc.
- [36] Sulubacak U. (2015), Yeni bir sözdizimsel işaretleme yönteminin kullanımıyla Türkçe'nin istatistiksel ayrıştırma başarımının artırılması, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, İstanbul Teknik Üniversitesi.

## EKLER

Ek A- Gereksinim Yönetim Araçları Listesi.....**Hata! Yer işareti tanımlanmamış.**

Ek B- Gereksinim Yönetim Araç Karşılaştırma Kriterleri ..... **Hata! Yer işareti tanımlanmamış.**

Ek C- DDİ Zinciri Sözlük Detayları .....**Hata! Yer işareti tanımlanmamış.**

Ek D- OG3 Gözden Geçirme Yazılımı Sözcüksel Analiz Veri tabanı ..... **Hata! Yer işareti tanımlanmamış.**

Ek E- Gereksinim gözden geçirme kayıtları (CD) **Hata! Yer işareti tanımlanmamış.**

## Ek A- Gereksinim Yönetim Araçları Listesi

**Tablo A.1** Gereksinim Yönetim Araçları Listesi [3]

	Yazılım	Sağlayıcı
1	Axure RP	Axure
2	Google Doc Drawings	Google Docs
3	iRise	iRise
4	SmartDraw	SmartDraw
5	CaliberRDM	Micro Focus
6	Focal Point	IBM
7	Requirements Composer	IBM
8	Gravity	Google Chrome
9	Personify Design Team Spec within TFS	TeamSolutions
10	Service-now	Service-now
11	Teamcenter Systems Engineering (TcSE)	Siemens
12	Blueprint Requirements Center	BluePrint
13	CaliberRM	Micro Focus
14	Cockpit	Arcway
15	CORE	Vitech Corp
16	Cradle	ThreeSL (3SL)
17	DOORS	IBM
18	Enterprise Architect	SparxSystems
19	inteGREAT	eDevTechnologies
20	MKS Integrity	MKS
21	Quality Center Requirements Management	HP
22	Rational ReqPo	IBM
23	Raven Professional	Ravenflow
24	Raven Cloud	Ravenflow
25	Raven Express	Ravenflow
26	Team Foundation Server, Visual Studio Team System	Microsoft
27	Cameo Requirements (and plugins)	MagicDraw
28	JIRA	Atlassian
29	MagicDraw	MagicDraw
30	Cognition Cockpit	Cognition
31	Dolphin	Street Light
32	Leap SE	Leap SE
33	ReqDB	Requirements Management
34	Accept360 Requirements	Accept Software
35	Accompa	Accompa
36	Agile Cycle	AccuRev

**Tablo A.1 (devam) Gereksinim Yönetim Araçları Listesi [3]**

37	Agile Manager	Aldon
38	Agile on Demand	Serena
39	Agility	AgileEdge
40	Agilo	Agile42
41	Avenqo PEP	Avenqo
42	Axiom	iConcur software
43	Balsamiq Mockups	Balsamiq
44	Banana Scrum	Banana Scrum
45	BaseCamp	BaseCamp
46	Case Spec	Goda Software
47	CaseComplete	Serlio Software
48	Contour	Jama Software
49	Dimensions RM	Serena
50	EdgeRM	Infogenium
51	Expression SketchFlow	Microsoft
52	FeaturePlan	Ryma
53	FeatureSet	FeatureSet
54	Gatherspace	Gatherspace.com
55	IdeaScale	IdeaScale
56	IRQA	Visure
57	Justinmind Prototyper	Justinmind
58	Kovair Global Lifecycle/ALM Solution	Kovair
59	Lighthouse	Artifact Software
60	LucidChart	Google Chrome
61	Mingle	ThoughtWorks
62	MockupScreens	MockupScreens
63	Objectiver	Respect-IT
64	OnTime	Axosoft
65	Pace RM	Viewset
66	Polarion Requirements	Polarion Software
67	ProjectCards	ProjectCards
68	Projectricity Requirements Tool	Projectricity
69	Prototype Composer	Serena
70	Psoda Requirements Management Module	Psoda
71	Qpack	Orcanos
72	Quality Center Agile Accelerator	HP
73	Rally Community Edition	Rally Software Development
74	Rally Enterprise Edition	Rally Software Development
75	RaQuest	SparxSystems
76	ReMA	Accord Software
77	ReqSheet	STP Soft
78	Requirements Definer	STP Soft

**Tablo A.1 (devam)** Gereksinim Yönetim Araçları Listesi [3]

79	Rommana	Rommana Software
80	RQMAN	Rednotes Technologies
81	Rtime	QAVantage
82	ScenarioPlus (Ian's) for Doors	ScenarioPlus
83	Scrumworks	CollabNet
84	SecTro	SecTro
85	Silver Catalyst	Silver Stripe Software
86	Statestep	Statestep
87	Storyboarding	STP Soft
88	TargetProcess	TargetProcess
89	TeamPulse	Telerik
90	TestTrack RM	Seapine
91	Together	Micro Focus
92	TopTeam Analyst	TechnoSolutions
93	TraceCloud	TraceCloud
94	TrackStudio	TrackStudio LTD
95	Verametric	Verametric
96	VersionOne Team, Enterprise and Ultimate Edition	VersionOne
97	VisibleThread	VisibleThread
98	Visual Requirements	Lucid Models Software
99	Visual Use Case	TechnoSolutions
100	workspace.com	workspace.com
101	Xtie (Cross tie)	Teledyne Brown Engineering
102	Acclaro DFSS	Axiomatic Design Solutions
103	Aligned Elements	Aligned
104	Envision VIP	Future Tech Systems
105	Reqtify	Greensoft
106	What to do next	4SQ Solutions
107	ReqMan	RequirementOne
108	PivotalTracker	Pivotal Labs

## Ek B- Gereksinim Yönetim Araç Karşılaştırma Kriterleri

Tablo B.1 Gereksinim Yönetim Araç Karşılaştırma Kriterleri [3]

Sıra	Kriter	Öncelik Puanı
1	Yeni gereksinim ekleme	3
2	Gereksinimleri düzenleme	3
3	Anahtar kelimelere, yapıya vb. göre harici metin dokümanlarından gelen gereksinimleri otomatik olarak tespit etme	3
4	Yapısal verileri, Excel'den yeni gereksinimler olarak toplu olarak içe aktarma	3
5	Aynı tür gereksinimler arasında bağlantılar oluşturma	3
6	Kriterlere göre gereksinim görünüm filtreleme	3
7	İzlenebilirlik Analizi (Gereksinimler arasında eksik bağlantıları bulma)	3
8	Gereksinimleri Word'e export etme (doc veya rtf)	3
9	Her bir gereksinim türü için detaylı arama (search)	3
10	Projeye göre gereksinim gruplama	3
11	Gereksinimler için ana çizgi oluşturma	3
12	Gereksinim değişiklikleri için otomatik denetim mekanizmaları (kullanıcı, saat / tarih, değişiklik notu ve ayrıntı değiştirme)	1
13	Harici API kullanma	1
14	Offline çalışabilme	1
15	Dokümantasyon yeteneği (çevrimiçi, basılı kopya vs.)	1
16	Belgeleri birden fazla gereksinime bağlama	1
17	Gereksinimler arasında hiyerarşik ilişki tanımlama	1
18	Anahtar kelimelere göre gereksinimleri arama	1
19	Verileri toplu halde Word'den yeni gereksinimler olarak alma	1
20	Birden fazla gereksinimi aynı anda kolayca taramak ve alanları doğrudan listede değiştirmek için gereksinimleri bir liste olarak görüntüleme	1
21	Gereksinim hiyerarşisinde yukarı ve aşağı gidip branchler arası gezinme	1
22	Gereksinimleri toplu olarak doğrudan girebilme	1
23	Minimum kullanıcı etkileşimi ile gereksinimleri girme (kısa yollar vs.)	1
24	Gereksinimleri bir yerel sürücüye depolanmayan dokümanlara bağlayabilme	1
25	Gereksinimleri bir hiyerarşide gösterebilme	1
26	Her bir gereksinim veya nesne için benzersiz bir kimlik atmasına izin verme, manuel atama	1
27	Farklı türdeki gereksinimler arasında bağlantılar oluşturma	1
28	Yazılımda süreç akışlarını modelleyebilme	1
29	Farklı gereksinim türleri ekleyebilme	1
30	İçgüdüsel olarak kullanımı kolaylığı	1
31	Öğrenmesi kolay	1



**Tablo B.1 (devam) Gereksinim Yönetim Araç Karşılaştırma Kriterleri [3]**

32	Her gereksinim için otomatik olarak benzersiz kimlik oluşturma	3
33	Zengin metin biçimlendirmesi	1
34	Doğrudan veri tabanında bir görüntü ile gereksinim tanımlama	3
35	Veri tabanında gömülü bir belge olan gereksinim tanımlama	2
36	Gereksinimleri bir Sharepoint konumundaki gerçek belgelerle ilişkilendirme	2
37	Bir use-case'i görsel olarak modelleyebilme ve otomatik olarak metin gereksinimlerine çevirebilme	1
38	Geri alma işleviyle gereksinimleri düzenleme	3
39	Gereksinimleri gruplara göre düzenleme	3
40	Gereksinim görünümünü bir kerede birden çok kritere göre sıralama	3
41	Özel görünümleri kaydedin	3
42	Herkese açık özel görünümleri kaydetme	1
43	Yazılı gereksinimlerden otomatik olarak diyagramlar üretme	1
44	Yazım hatalarını otomatik olarak kontrol etme	3
45	Dilbilgisi kullanımını otomatik olarak kontrol etme	2
46	Belirsiz kelimelerin kullanımı için otomatik kontrol	1
47	Gereksinimleri silme	2
48	Her türlü izlenebilirlik sonucunu bir tabloda gösterme	2
49	İzlenebilirlik sonuçlarını bir diyagramda görüntüleme (ör. Ağaç)	2
50	Çeşitli önceden tanımlanmış askeri ve ticari standart formatlarda gereksinim dokümanları üretme	1
51	Dokümantasyon çıktısı için özel biçimi tanımlama	3
52	Gereksinimlerin ön izlemesi	1
53	Gereksinimleri HTML'ye aktarma	1
54	Gereksinimleri sınırlandırılmış biçime (ör. CSV) aktarma	2
55	Gereksinimleri XML'e aktarma	1
56	Farklı gereksinim türlerini tanımlama ve yakalama	3
57	Gereksinim numaraları için özel biçim tanımlama	1
58	Her bir gereksinim türü için özel iş akışı tanımlama	1
59	Yeni projeler için şablon desteği	2
60	Gereksinimlerin ana çizgilerini karşılaştırma	3
61	Gereksinimlerin belirli çizgileri üzerinde çalışabilme	3
62	Tek tek kullanıcılar için erişim izinlerini yönetme	1
63	Kullanıcı grupları için erişim izinlerini yönetme	1
64	Etkilenen proje katılımcılarına gereksinim değişikliklerini e-posta ile bildirme	1
65	Gereksinimlerle ilgili görüşmeleri takip etme, konuşmalarınızı gereksinim nesnelere ekleme	1

**Tablo B.1 (devam) Gereksinim Yönetim Araç Karşılaştırma Kriterleri [3]**

66	Metrik raporlama	1
67	Metrik grafikleri	1
68	Verileri verileri Excel'e aktarma	1
69	Gereksinim verileri için geçici sorgular gerçekleştirme	1
70	Tasarım araçları ile otomatik entegrasyon (ör. Rose, XDE, Enterprise Architect, vb.)	1
71	Geliştirme ortamlarıyla otomatik entegrasyon (ör. Eclipse)	1
72	Bağlantılı dokümanlar için revizyon kontrol sistemi ile otomatik entegrasyon (ör. Sharepoint, CVS, SourceSafe, vb.)	1
73	Test yönetim sistemi ile otomatik entegrasyon (ör. QualityCenter)	1
74	Açık veri tabanı sistemini destekleme (standart sorgu erişimi)	1
75	Sistem içinde içeriğe duyarlı yardımın mevcudiyeti	1
76	Telefon desteğinin mevcudiyeti	1
77	Çevrimiçi destek (örneğin, mesaj panosu, bilgi tabanı, kullanıcı grubu, wiki, vb.)	1
78	Öğrenme yardımcılarının kullanılabilirliği (ör. Örnek gereksinimler, iş akışı öğreticisi, vb.)	1
79	Eğitim sınıflarının mevcudiyeti	1
80	Toplantı notları ve bireysel notlara bağlantı ekleme vs.	1
81	Her bir gereksinim için bir paydaş tanımlama	1
82	Gereksinimlerin izini takip etme / imzalama	1
83	E-posta yoluyla gereksinimlerin onaylanması / imzalanması talep edilmesi	1
84	Gereksinim hiyerarşisinden bağımsız olarak gereksinimler arasındaki izlenebilirlik tanımlama	1
85	Gereksinim bağlantılarıyla ilgili herhangi bir işlem maddesini göstermek için rapor verme	1
86	Kullanıcı ara yüzü gereksinimlerini bir kullanıcı ara yüzü prototipiyle birleştirme	1
87	Farklı analistlere tamamlanması için farklı gereksinim grupları atama	1
88	Bir dizi gereksinim üzerinde farklı analistlerden çalışma arasındaki değişiklikleri birleştirme	1
89	Bir dizi gereksinim için bir sözlük oluşturma	1
90	İzlenebilirlik verisinin özelleştirilmiş görünümünü, kullanıcının hangi sütun türlerini gösterebileceğini seçebildiği bir tabloda görüntüleme	1
91	Tartışmalara katıldığımızda (örneğin, toplantıdaki bir projekte canlı düzenlemeler yaparken) gereksinim listesinde bir takım değişiklikleri hızlı ve gerçek zamanlı olarak yapma	1
92	Bir gereksinim gözden geçirme oturumunda olduğu gibi gereksinim özneliği değişikliklerinizi hızlı bir şekilde yapabilme (not alınması ve insanlar gittikçe yeniden atama)	1
93	Filtrelenmiş görünüm hızlı yansıtma	1
94	İhtiyaçları birbirine bağlamak için sürükleyip bırakma	1
95	İzlenebilirlik hiyerarşisine kolayca gitme	1
96	Gereksinim durumuna ve gereksinim başına gözden geçirme sayısına göre olgunluk raporu	1

**Tablo B.1 (devam) Gereksinim Yönetim Araç Karşılaştırma Kriterleri [3]**

97	Bir dizi gereksinimin gözden geçirilip değerlendirilmediğine ilişkin rapor sunma	1
98	Gereksinimin önceki bir sürümüne geri dönme	1
99	Gereksinimlerin önceki bir sürümünü aynen olduğu gibi yeniden üretme	1
100	Gereksinimler temel alınarak bir regresyon testi paketi oluşturma	1
101	Gereksinimler temel alınarak bir sistem test paketi oluşturma	1
102	Tek bir gereksinim için, gereksinimler hiyerarşisinde nerede oturduğunu görsel olarak vurgulama	1
103	Gereksinimlerin bazı özelliklerine dayalı olarak müşteri gereksinimlerini kısıtlama	1
104	Gereksinimlere ilişkin mevcut tüm açık konular hakkında rapor hazırlama	1
105	Hangi gereksinimler üzerinde çalıştığını gösteren rapor	1
106	Bir gereksinimi başka birinin kopyası olarak işaretleme ve bir araya getirme	1
107	Yeni / değişen gereksinimlerin bir listesi için filtre uygulama ve değişiklikleri gözden geçirme	1
108	Gözden geçirme planlarına bağlı olarak gereksinimleri gruplayabilme ve set halinde kabul ve iptale izin verme	1
109	Verilen gereksinimlerin amacının ne olduğu veya diğer açıklama türleri için gereksinimlerin etrafında kabarcık metinler saklayabilme	1
110	Kutu özniteliklerine ek olarak gereksinim nesnelere özel niteliklerine filtreler uygulama	1
111	Gereksinimlerin listesinde, listeyi tarayabilme ve gereksinimdeki eski sürümleri listede görüntüleyebilme	1
112	Özgün bir gereksinimi kimin istediğini takip etme	1
113	E-postaları belirli gereksinimler ile ilişkilendirme	1
114	Yazılımdan gözden geçirebileceğiniz çıktı oluşturabilme ve çıktıda değişiklikler yaparak tekrar yazılıma girdi olarak verme	1
115	Yeni çizgiler alabilme; sürümler arası yenilikleri gösterebilme	1
116	Eşlenmiş ekran görüntüleri ile süreç akışlarına arasında gezinme ve ilgili ekranları görebilme	1
117	Var olan visio belgelerine yönelik gereksinimleri takip etme	1
118	Gereksinimleri tam bir modele bağlama	1
119	Gereksinimler içinde tabloların girilmesine destekleme	1
120	Özellikleri toplu olarak düzenleme	1
121	Hiyerarşi içinde gereksinimleri taşımak için sürükleyip bırakma yeteneği	1
122	Gerçek zamanlı güncelleme (ör. DB vs. bir doküman)	1
123	Önceden tanımlanmış değer seçenekleri ile karmaşık nitelikleri ayarlama	1
124	Dışa aktarılan belgedeki değişiklikleri önceki bir taban çizgisine göre kıyaslayıp farklı renkle vurgulama	1
125	Gereksinimlerle ilgili video dosyalarını bir veya daha fazla gereksinimle ilişkilendirebilme	1
126	Gereksinimlerle ilgili video dosyaları yükleyebilme ve saklama	1
127	Gereksinimlerle ilgili ses dosyalarını bir veya daha fazla gereksinime bağlama	1

**Tablo B.1 (devam) Gereksinim Yönetim Araç Karşılaştırma Kriterleri [3]**

128	Gereksinimlerle ilgili ses dosyaları yükleyebilme ve saklama	1
129	Kontrol panel raporlarını görüntüleme	1
130	Gösterge tablosu raporlarını özelleştirme	1
131	Gereksinim nesnelerini özelliklerini ekleyerek raporları özelleştirme	1
132	Gereksinimlere oy verme	1
133	Diğer akışlardaki adımlara akışları bağlama	1
134	Gereksinim durumu hakkında burndown raporları oluşturma	1
135	OneNote entegrasyonuna izin verme	1
136	Çevrimiçi geri geldiğinde otomatik olarak senkronize olma	1
137	Kullanıcıya çevrimdışı çalışmayı senkronize etmesini düzenli olarak hatırlatma	1
138	Word veya Excel'den alanlara kopyala / yapıştırırken, mevcut alan formatını koruma	1
139	Akışları işlemek için ekran görüntülerini eşleştirme	1
140	Gereksinimlerle ilgili ayrıntılı bilgileri görüntülemek için yalnızca birkaç tıklama (excel gibi, çok sayfanın derinlerine gömülmüş web tabanlı bir araç değil)	1
141	Az sayıda kullanıcı etkileşimi ile hızlıca birçok gereksinimi okuma (ör. Fare tıklamaları, klavye basakları vb.)	1
142	İzleme aracını (tercihen Sharepoint) yayınlamak için otomatik entegrasyon	1
143	Gereksinimlere öncelik koyma	1
144	Gereksinim kaynağını yakalama, gösterme	1
145	Girilen yeni gereksinimler incelenmek üzere yeni olarak işaretleme	1
146	Hangi gereksiniminin beklendiğini gösterme	1
147	Gereksinimler arasındaki ilişki bağının niteliğini tanımlama (ister "izler", "bağımlılık" vb.)	1
148	Tanımlanmış gereksinim grupları arasında bağlantılar oluşturma	1
149	Gereksinimler ile bağlantılı rasyonelleri gösterme	1
150	İzlenebilirlik raporları üretme	1
151	Gereksinimlere doğrulama yöntemleri atama	1
152	Gereksinimi ürün doğrulama statüsüne bağlama	1
153	Gereksinimler ile ilgili açıktaki maddeleri takip etme	1
154	Gereksinim gözden geçirme istatistikleri sunma	1
155	İş hedefleri gibi başka bir gereksinim türüne gereksinimleri bağlama	1
156	Gereksinimi tamamen kaybetmeyen bir silme seçeneğine izin verme	1
157	Gereksinim üzerinde madde/hata yaratılmasına izin verme ve bunu gereksinime bağlama	1
158	Düzenlenebilir bir görünüme erişmek için birden çok tıklamaya gerek duymadan bireysel gereksinimleri hızlı bir şekilde düzenleme	1
159	Gereksinim için, başka bir araçtaki bir konuya URL olarak bağlantı oluşturabilme	1
160	Sorun/hata izleme takip özelliği (işlem maddesi takibi)	1

**Tablo B.1 (devam) Gereksinim Yönetim Araç Karşılaştırma Kriterleri [3]**

161	Açılan Sorun ve hatalar için çözümleri de takip edebilme	1
162	Gereksinim setinin bir alt kümesini görünümüne dayalı uygulanan bir filtre ile dışa aktarma	1
163	Gereksinim değişikliklerinin geçmişini görüntüleme	1
164	Kullanıcı Arabirimi gereksinimlerini bir UI prototipinin belirli bölümleriyle eşleştirebilme	1
165	Modelleri diğer modellerdeki alt nesnelere bağlayabilme	1
166	Modellerde alt nesnelere gereksinimleri bağlama	1
167	Modelleme yapabilme (context diagram)	1
168	Modelleme yapabilme (org chart vs.)	1
169	Belirli bir modelle ilişkili gereksinimlerin listesini çıkarma	1
170	Okunabilir görünümde offline iken çalışabilme	1
171	Kullanıcılara projeye bağlı gereksinim kümelerini kısıtlama	1
172	Gereksinim veya diğer niteliklere göre açık, kapalı, devam etmekte olan vb. işlem maddeleri sayısı hakkında rapor	1
173	E-Postayı gereksinimin belirli bir sürümüne bağlama	1
174	Gereksinim niteliklerinin gerekip gerekmediğini seçme	1
175	Bir belgenin URL'sini bir Sharepoint konumunda saklama	1
176	Bir dizi gereksinim için otomatik olarak taslak UAT komut dosyası oluşturma	1
177	Gereksinimlere bağlı UAT test komut dosyaları oluşturma	1
178	Pek çok gereksinim arasında bir kerede bağlantı oluşturma	1
179	Her tür gereksinim için tek bir özel iş akışı tanımlama	1
180	Önceden tanımlanmış iş akışları	1
181	Çevrimiçi modda İki kişinin aynı gereksinimi eşzamanlı olarak düzenlemesini kısıtlama	1
182	Gereksinimler için açıklayıcı sözlük tanımlama, fareyle click ile tanıma gitme	1
183	İş veya inceleme için gereksinimi sahiplerine atama	1
184	Gereksinimleri, projeler arasında tekrar tekrar kullanılabilme	1
185	Yazılımda düşük kalite ekran görüntülerini kullanma	1
186	Yazılımda yüksek kalite ekran görüntülerini kullanma	1
187	Gereksinimleri başka bir projeden kopyalayabilme	1
188	Gereksinimleri kullanıcı tanımlı izlenebilirlik tipleri ile başak doküman parçalarına bağlayabilme	1
189	Araç, bazı izlenebilirlik bağlantı türlerinin oluşturulmasını ve değiştirilmesini zorunlu kılabilme	1
190	Araç M: N çift yönlü bağlamayı destekleme	1
191	Bağlantılardaki tutarsızlıkları tespit etme	1
192	Değişik karmaşıklığa sahip matematik denklemleri ekleyebilme	1
193	Semboller ve biçimlendirme ekleyebilme (Yunanca Semboller, alt simgeler, üst simgeler)	1

**Tablo B.1 (devam) Gereksinim Yönetim Araç Karşılaştırma Kriterleri [3]**

194	Kullanıcı tanımlı öznelikler için doğrulama denetimleri tanımlama	1
195	Sınırsız sayıda gereksinim nitelikleri tanımlayabilme	1
196	Resmi ve gayri resmi değişiklikler arasında ayırım yapabilme	1
197	Kullanıcı arabirimi değiştirebilme (düğmelere, kısayol tuşlarına ve menülere işlev atama)	1
198	Kolayca makro tanımlama ve düğmelere, tuşlara ve menülere atama	1
199	Kullanıcı tanımlı numaralandırma türleri, nitelikler için kullanıldığında, kullanıcı bir listeden istenen değeri "seçebilir"	1
200	"Tekrar bulun" ve "tekrar bul / değiştir" için sağlanan kısa yol	1
201	Diğer uygulamalara kesme ve yapıştırmaı destekleme	1
202	Yazılım içinde farklı alanlar arasında kesme ve yapıştırmaı destekleme	1
203	Lisans politikası	3
204	Lisans Maliyeti	3
205	Garanti seçenekleri	1
206	Bakım ve Güncelleme desteği	1
207	Bakım Maliyeti	3
208	Hosted Kurulum	3
209	Merkezi repoya kurulum	3
210	İstemci olarak kurulum	1
211	Web ara yüzü istemci platformu	3
212	Windows ara yüzü istemci platformu	3
213	Unix/Linux ara yüzü istemci platformu	1
214	İstemci Bellek, hafıza sistem performans gereksinimleri	3
215	Birden fazla eş zamanlı kullanıcı desteği	3
216	Windows destekli sunucu platformu	3
217	Unix/Linux destekli sunucu platformu	1
218	Sunucu Bellek, hafıza sistem performans gereksinimleri	3
219	Windows uyumlu veri tabanı	3
220	Unix/Linux uyumlu veri tabanı	1
221	MS Access uyumlu veri tabanı	1
222	SQL Server uyumlu veri tabanı	1
223	Oracle uyumlu veri tabanı	1
224	MySQL uyumlu veri tabanı (open source)	1
225	Ticari veri tabanı (tescilli değil)	1
226	Farklı veri tabanlarından gereksinim birleştirebilme	1
227	Kurulum Kolaylığı	1
228	Yönetici panel kullanım kolaylığı	1

**Tablo B.1 (devam)** Gereksinim Yönetim Araç Karşılaştırma Kriterleri [3]

229	Açık kaynak kod	1
230	Yaygın kullanım (müşteri sayısı)	1
231	Firma Agile ve waterfall desteği veriyor mu?	1
232	Gereksinim sayısı arttıkça yazılımın karşılaştırma performansı	1
233	Matrix formatlı bir gözden geçirme ile bir gereksinime birden fazla yazılım sağlayıcısının eşleşmesi yapılabilir mi? (attribute ekleme)	1



## Ek C- DDİ Zinciri Sözlük Detayları

**Tablo C.1** Sözcük Sınıfları

Sözcük	Analizdeki Çıktısı
İsim	+Noun
Fiil	+Verb
Özel isim	+Noun+Prop
Sıfat	+Adj
Teknik sözcükler	+Noun
İkilemeler	+Dup
Edatlar	+Postp+PCNom
Soru	+Ques
Zamirler	+Det
Sayılar	+Num+Card
Bağlaçlar	+Conj
Zamirler	+Pnon

**Tablo C.2** Eklerde kullanılan özel harfler

A	a yada e harfi yerine
D	d yada t yerine
C	c yada ç yerine
H	ı,i,u,ü harfleri yerine
I	ı yada i harfleri yerine
0	Boş geçiş



Tablo C.3 DDİ Sözlükteki Etiketler-1

Ekin etiketi	İngilizcesi	Açıklaması	Ekin kendisi
Ness	Necessitative	Gereklilik eki	+IHk
Agt	Agt	İlgi eki	+cH
Dim	Dimentional	Küçümseme eki	+cHk
A1sg	A1sg	1. tekil şahıs eki	+Hm
A1pl	A1pl	3. çoğul şahıs eki	+Hz
A2sg	A2sg	2. tekil şahıs eki	+sHn
A2pl	A2pl	2. çoğul şahıs eki	+sHnHz
A3pl	A3pl	3. çoğul şahıs eki	+lAr
A3sg	A3sg	3. tekil şahıs eki	0
Become	Become	Olma eki	+lA,s
Acquire	Acquire	Aitlik eki	+lAn
P3pl	Personal	3. çoğul şahıs iyelik eki	+lArH
P1sg	Personal	1. tekil şahıs iyelik eki	+Hm
P2sg	Personal	2. tekil şahıs iyelik eki	+Hn
P1pl	Personal	1. çoğul şahıs iyelik eki	+HmHz
P2pl	Personal	2. çoğul şahıs iyelik eki	+Hmnz
P3sg	Personal	3. tekil şahıs iyelik eki	+sH
With	With	Berberlik eki	+lH
Without	Without	Berber olmama eki	+sHz
Dat	Dative	İsmin yönelme (e) hali	+yA
Loc	Locative	İsmin bulunma (de) hali	+DA
Abl	Ablative	İsmin ayrılma (den) hali	+DAn
Gen	Genitive	İsmin ilgi (in) hali	+nHn
Ins	Ins	Birliktelik eki	+(y)lA
Acc	Accusative	İsmin yüklenme (i) hali	+(y)H
Rel	Relation	İlgi eki	+ki
While	While	Zaman zarfı	+yken
Cond	Condition	Koşul eki	+ysA
Adverb+AsIf	Adverb	Fiilden zarf yapma eki	+cAsHnA
Narr	Narrative	Rivayet geçmiş zaman eki	+ymH,s
Reflex	Reflexive	Dönüşlülük eki	+Hn
Recip	Reciprocal	İşteşlik eki	+H,s
Caus	Causative	Ettirgen fiil eki	+dHr
Pass	Passive	Pasiflik eki	+Hl
Fut	Future	Gelecek zaman eki	+yAcAk
Prog2	Progressive	Devamlılık eki	+mAktA
Neces	Necessitative	Gereklilik eki	+mAlH
Prog1	Progressive	Şimdiki zaman eki	+Hyor
Adj+Aor	Adjective	Fiilden sıfat yapma eki	+Ar
Aor	Aorist	Geniş zaman eki	+Hr
Able	Able	Yeterlilik eki	+yAbHl
Hastily	Hastily	Tezlik eki	+yHver
Repeat	Repeat	Süreklilik eki	+yAdHr
Neg	Negative	Olumsuzluk eki	+mA
Caus	Causative	Ettirgenlik eki	+t
Inf1	Infinitive	Fiilden isim yapma eki	+mA
Inf2	Infinitive	Fiilden isim yapma eki	+mAk
Inf3	Infinitive	Fiilden isim yapma eki	+H,s
Past	Past	Geçmiş zaman eki	+yDH
Cop	Cop	Kesinlik eki	+DHr
Narr	Narrative	Rivayet geçmiş zaman eki	+mH,s
Cond	Condition	Koşul eki	+ysA
Pos-Aor	Positive aorist	Olumlu geniş zaman eki	+Hr
Pos-Fut	Positive future	Olumlu gelecek zaman eki	+yAcAk
Pos-Narr	Positive narrative	Olumlu rivayet zaman eki	+ymH,s
Pos-Opt	Positive optative	Olumlu dilek şart eki	+yA
Pos-Desr	Positive desire	Olumlu istek eki	+sA

**Tablo C.4 DDİ Sözlükteki Etiketler-2**

Ekin etiketi	İngilizcesi	Açıklaması	Ekin kendisi
AsLongAs	As long as	Uzun süren zaman eki	+dHkçA
SinceDoingSo	Since doing so	Süregelen zamanı bildirir ek	+yAlH
AfterDoingSo	After doing so	Sonraki zamanı bildirir ek	+yHp
ByDoingSo	By Doing So	Devam eden zaman eki	+yArAk
When	When	Olduğu zamanı gösteren ek	+yHncA
Feellike	Feel like	Gibi hissetme eki	+yAsH
PastPart	Past participle	Sıfat Fiil eki	+dHk
FutPart	Future participle	Sıfat Fiil eki	+yAcAk
PresPart	Present participle	Sıfat Fiil eki	+yAn
Adj+NarrPart	Narrative participle	Sıfat Fiil eki	+ymH_s
NotState	Not State	Olumsuzluk bildiren fiilden isim yapma eki	+mAzLHk
WithoutHavingDoneSo	Without having done so	Yapmaksızın eki	+mAksHzHn
WithoutHavingDoneSo	Without having done so	Yapmadan eki	+mAdAn
Adj+Agt	Adjective Agt	İlgi eki	+yHcH
Adamantly	Adamantly	Kıyasıya eki	+yAsHyA
Aor-A1sg	Aorist A1sg	Geniş zaman 1. tekil şahıs eki	+m
Aor-A2sg	Aorist A2sg	Geniş zaman 2. tekil şahıs eki	+zsHn
Aor-A3sg	Aorist A3sg	Geniş zaman 3. tekil şahıs eki	+z
Aor-A1pl	Aorist A1pl	Geniş zaman 1. çoğul şahıs eki	+yHz
Aor-A2pl	Aorist A2pl	Geniş zaman 2. çoğul şahıs eki	+zsHnHz
Aor-A3pl	Aorist A3pl	Geniş zaman 3. çoğul şahıs eki	+zlAr
Start	Start	Başlama eki	+yAkoy
Stay	Stay	Kalma eki	+yAkAl
Equ	Equal	Eşitlik eki	+cA
Narr+A3sg	Narrative A3 singular	Rivayet 3. tekil şahıs eki	+mH_s
Past+A2sg	Past A2 singular	Geçmiş zaman 2. tekil şahıs eki	+yDHn
Past+A1sg	Past A1 singular	Geçmiş zaman 1. tekil şahıs eki	+yDHm
Past+A3sg	Past A3 singular	Geçmiş zaman 3. tekil şahıs eki	+yDH
Zero-Cond	Condition	Koşul eki	+sA
Zero-Pres-A2pl	Present A2 plural	Şimdiki zaman 2. çoğul şahıs eki	+sHnHz
Zero-Pres-A1pl	Present A1 plural	Şimdiki zaman 1. çoğul şahıs eki	+yHz
Zero-Pres-A2sg	Present A2 singular	Şimdiki zaman 2. tekil şahıs eki	+sHn
Zero-Pres-A1sg	Present A1 singular	Şimdiki zaman 1. tekil şahıs eki	+yHm
+Pres+Zero+Cop	PresZeroCop	Şimdiki zamanda kesinlik eki	+DHr
Adj+Asif	As if	Gibi eki	+cA
Repeat	Repeat	Süreklilik eki	+yAgör
EverSince	Ever since	Süreklilik eki	+yAgel
Almost	Almost	Yaklaşma eki	+yAyAz
DB	Derivational Boundary	Üretim Sınırı	Yapım ekleri için kullanılır

## Ek D- OG3 Gözden Geçirme Yazılımı Sözcüksel Analiz Veri tabanı

<b>K6-Sözlüğü</b> neredeysen, mümkün olursa, mümkün olan, normal şartlar altında, şartlar uygun olduğu sürece, şartlar uygun ise, uygunsa, uygun ise, uygun bir, uygulanabilirse, mümkün ise, mümkün olduğunca, mümkün olduğu sürece, mümkünse, gerekirse, gerekliğinde , gerekli ise, benzer şekilde, benzer , benzeri, birkaç, gibi, gerçeğe yakın, rahatsız edici, belirgin, bazısı, bazı , bazıları, biraz, birkaçı, bir kaçı, çoğu , çoğunluğu, az çok, bir çok, kimi zaman, verimli,	kararlı, istikrarlı  <b>K8-Sözlüğü</b> şüphesiz, elbet , elbette, kuşkusuz, eninde sonunda, er geç, tabi , tabi ki, mutlaka, açıkça , açık bir şekilde, düpedüz, çünkü  <b>K9-Sözlüğü</b> bir şekilde , en kötü ihtimalle, en kötü ihtimal ile, en iyi ihtimalle, bir ihtimal, çok muhtemel, olabildiğince , kısmen, kapsamlı, galiba, muhtemelen, belki, yaklaşık olarak, maksimize , minimize , ilgili , yani, rastgele , sonsuz ,  <b>K10-Sözlüğü</b> kesinlikle, her zaman,	her halükarda, her durumda, daima, hiçbir zaman, kesin olarak, asla, hiç , ne olursa olsun,  <b>K11-Sözlüğü</b> genellikle, sıklıkla, çoğunlukla , genel olarak, genelde, genel , malum, normal olarak, geleneksel, teamüle göre, alışlagelmiş,  <b>K12-Sözlüğü</b> kullanıcı dostu, çok hızlı, aşırı, çok yavaş, yavaşça, hızlı hızlı, yavaş yavaş, hızlıca, gerçekçi, güzel, zayıf, çok az, çok büyük, çok küçük, kolayca, kolaylıkla, kolay , zor , kolaylı,	çok kolay, çok zor, aniden, kötü, iyi , güçlü , hoş, kötü,, güçlü,, yumuşak, sert, çabukça, çabucak, çok fazla, sık sık, yeterince, yeterli, kabul edilebilir, faydalı, sağlam , pek çok, çabuk ,  <b>K14-Sözlüğü</b> vb., ve benzeri, DSB, TBD, TODO, ileride belirtilecek, ileride belirlenecek, ileride açıklanacak, ..., ..., sonra belirtilecek, sonra belirlenecek, sonra açıklanacak
---	--	--	---

**Ek E- Gereksinim gözden geçirme kayıtları (CD)**



## ÖZGEÇMİŞ

### KİŞİSEL BİLGİLER

Adı Soyadı:

Uyruğu:

Doğum Yeri ve Tarihi:

Medeni Hali:

Adres: E-Posta Adresi:

### EĞİTİM

### MESLEKİ DENEYİM