

**YAZILIM ÜRÜN HATTINDAKİ MODÜLLERİN ÖZELLİK  
AĞACI ANALİZİ İLE YENİDEN KULLANILABİLİRLİĞİNİN  
ARTTIRILMASI: OSGİ ÇERÇEVELİ RADAR KULLANICI  
ARAYÜZÜ ÖRNEĞİ**

**IMPROVING REUSABILITY OF MODULES IN SOFTWARE  
PRODUCT LINE BY FEATURE TREE ANALYSIS:  
THE CASE OF AN OSGİ-FRAMED RADAR USER  
INTERFACE**

**EZGİ CANKURTARAN**

**DR. ÖĞRETİM ÜYESİ AYÇA TARHAN**

**Tez Danışmanı**

Hacettepe Üniversitesi  
Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin  
Bilgisayar Mühendisliği Anabilim Dalı için Öngördüğü  
YÜKSEK LİSANS TEZİ olarak hazırlanmıştır.

2018

EZGİ CANKURTARAN'nın hazırladığı "Yazılım Ürün Hattındaki Modüllerin Özellik Ağacı Analizi İle Yeniden Kullanılabilirliğinin Arttırılması: OSGi Çerçevesi Radar Kullanıcı Arayüzü Örneği" adlı bu çalışma aşağıdaki jüri tarafından BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI' nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Doç. Dr. Kayhan İMRE  
Başkan



Dr. Öğretim Üyesi Ayça TARHAN  
Danışman



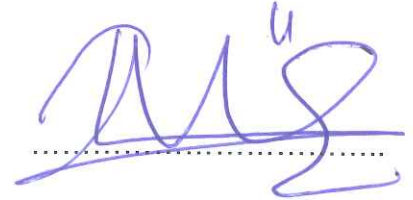
Dr. Öğretim Üyesi Murat AYDOS  
Üye



Dr. Öğretim Üyesi Fuat AKAL  
Üye



Dr. Öğretim Üyesi Tülin ERÇELEBİ AYYILDIZ  
Üye



Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından YÜKSEK LİSANS TEZİ olarak onaylanmıştır.

Prof. Dr. Menemşe GÜMÜŞDERELİOĞLU  
Fen Bilimleri Enstitüsü Müdürü

## YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin/raporumun tamamını veya herhangi bir kısmını, basılı (kağıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe Üniversitesine verdiğimi bildiririm. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alınarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim Kurulu tarafından yayınlanan "**Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge**" kapsamında tezim aşağıda belirtilen koşullar haricince YÖK Ulusal Tez Merkezi / H.Ü. Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

- Enstitü / Fakülte yönetim kurulu kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren 2 yıl ertelenmiştir. <sup>(1)</sup>
- Enstitü / Fakülte yönetim kurulunun gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren ... ay ertelenmiştir. <sup>(2)</sup>
- Tezimle ilgili gizlilik kararı verilmiştir. <sup>(3)</sup>

13/09/2018  
  
Ezgi CANKURTARAN

i

<sup>i</sup>"Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge"

(1) Madde 6. 1. Lisansüstü teze ilgili patent başvurusu yapılması veya patent alma sürecinin devam etmesi durumunda, tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu iki yıl süre ile tezin erişime açılmasının ertelenmesine karar verebilir.

(2) Madde 6. 2. Yeni teknik, materyal ve metotların kullanıldığı, henüz makaleye dönüşmemiş veya patent gibi yöntemlerle korunmamış ve internetten paylaşılması durumunda 3. şahıslara veya kurumlara haksız kazanç imkanı oluşturabilecek bilgi ve bulguları içeren tezler hakkında tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulunun gerekçeli kararı ile altı ayı aşmamak üzere tezin erişime açılması engellenebilir.

(3) Madde 7. 1. Ulusal çıkarları veya güvenliği ilgilendiren, emniyet, istihbarat, savunma ve güvenlik vb. konulara ilişkin lisansüstü tezlerle ilgili gizlilik kararı, tezin yapıldığı kurum tarafından verilir \*. Kurum ve kuruluşlarla yapılan işbirliği protokolü çerçevesinde hazırlanan lisansüstü tezlerle ilişkin gizlilik kararı ise, ilgili kurum ve kuruluşun önerisi ile enstitü veya fakültenin uygun görüşü üzerine üniversite yönetim kurulu tarafından verilir. Gizlilik kararı verilen tezler Yükseköğretim Kuruluna bildirilir.

Madde 7.2. Gizlilik kararı verilen tezler gizlilik süresince enstitü veya fakülte tarafından gizlilik kuralları çerçevesinde muhafaza edilir, gizlilik kararının kaldırılması halinde Tez Otomasyon Sistemine yüklenir

\* Tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu tarafından karar verilir.

## ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada,

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

13/09/2018



EZGİ CANKURTARAN

## ÖZET

# YAZILIM ÜRÜN HATTINDAKİ MODÜLLERİN ÖZELLİK AĞACI ANALİZİ İLE YENİDEN KULLANILABİLİRLİĞİNİN ARTTIRILMASI: OSGi ÇERÇEVELİ RADAR KULLANICI ARAYÜZÜ ÖRNEĞİ

**Ezgi CANKURTARAN**

**Yüksek Lisans, Bilgisayar Mühendisliği Bölümü  
Tez Danışmanı: Dr. Öğretim Üyesi Ayça TARHAN**

**Eylül 2018, 93 sayfa**

Yazılımı yeniden kullanma, kısalttığı geliştirme zamanı ile maliyeti düşürmesi ve yazılım kalitesinde artış sağlaması ile yazılım geliştirmede önemli bir pratik olarak görülmektedir. Yeniden kullanım birçok yazılım geliştirme yaklaşımında uygulansa da Bileşen Tabanlı Yazılım Geliştirme ve Yazılım Ürün Hattı yaklaşımının temelini oluşturmaktadır. Literatürde yeniden kullanılabilirliği değerlendirmek için farklı yöntem ve metrikler sunulmuştur. Bu tez çalışmasında, yazılım geliştirme için servis platformu sağlayan OSGi çerçevesi kullanılarak geliştirilmiş ve Yazılım Ürün Hattı yaklaşımına dayalı mimarisi olan bir radar kullanıcı arayüzü yazılımının, yeniden kullanılabilirliğinin arttırılması hedeflenmiştir. Çalışmada, bileşen tabanlı geliştirmede ve ürün hattı geliştirmede yeniden kullanılabilirliği ölçmek için yapılmış çalışmalar, sistematik yöntemle araştırılmıştır. Araştırma sonucunda yeniden kullanılabilirliği ölçmek için en çok kullanılan yazılım faktörleri ve bu faktörleri sayısallaştıran metrikler belirlenmiştir. Belirlenen metriklerle edinilen ölçüm sonuçlarına göre, yazılım Özellik Ağacı metoduna göre yeniden yapılandırılmıştır. Tez çalışmasında önerilen metot, Java dilinde geliştirilmiş dört kullanıcı arayüzü yazılımının yeniden kullanılabilirliğinin arttırılması için uygulanmıştır. Yapılandırma sonrasında ölçümü alınan LCOM, CC, WMC ve ACD (Ortalama Çevrimsel Bağımlılık) metrik değerlerinin her proje için düştüğü görülmüştür. Yapılandırma

öncesi ve sonrasında ölçülen Ortalama Çevrimsel Bağımlılık metriğinin değerindeki değişimin anlamlılığını değerlendirmek için istatistiksel anlamlılık analizi uygulanmıştır. Elde edilen sonuçlara göre uygulanan metodun Proje-1, Proje-2, Proje-4 için anlamlı Proje-3 için ise anlamlı olmadığı sonucuna ulaşılmıştır.

**Anahtar Kelimeler:** yeniden kullanılabilirlik, yazılım metrikleri, bağlaşım, bileşen tabanlı yazılımlar, ürün hattı, özellik ağacı, OSGi

## **ABSTRACT**

# **IMPROVING THE REUSABILITY OF MODULES IN SOFTWARE PRODUCT LINE BY FEATURE TREE ANALYSIS: THE CASE OF AN OSGi-FRAMED RADAR USER INTERFACE**

**Ezgi CANKURTARAN**

**Master of Science, Computer Engineering Department**

**Supervisor: Assist. Prof. Dr. Ayça TARHAN**

**September 2018, 93 pages**

Software reuse is an important practice in software development because of its potential to reduce costs by shortening development time and increasing software quality. Although software reuse is applied in different development methodologies, it is the main concept in Component Based Software Development and Software Product Line Development approaches. The academia has proposed different methods and metrics to analyze software reusability. This study aims at improving the reusability of a radar user interface software, which has an architecture based on a software product line approach, and which has been developed by using a service based platform for modular software, namely the OSGi Framework. Studies that measure reusability of OSGi and product line based software have been searched systematically. According to the results of systematic research study, software reusability factors and software reusability metrics are determined. According to software reusability measurement results, software is refactored based on Feature Tree method. The method, which is proposed in this study, has been used to refactor for four radar user interface software that have been developed using Java to improve reusability. The method, which is proposed in this study, has been used to refactor for four radar user interface software that have been developed using Java to improve reusability.

After refactoring, it's observed that LCOM, CC, WMC and ACD (Average Cycling Dependency) metrics were dropped down. Statistical significance analysis is used to evaluate change of significance of ACD metrics that were measured before refactoring and after refactoring. Accordingt to results, it's observed that method proposed is significance for Project-1, Project-2 and Project-4 while it's insignificant for Project-3.

**Keywords:** reusability, software metrics, coupling, component based software, software product line, feature tree, OSGi



## TEŐEKKÜR

Öncelikle bu tez alıřmam sırasında kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösteren ve beni her zaman güler yüz ile karşılayan değerli danışman hocam Dr. Öğretim Üyesi Aya TARHAN'a ve sayın jüriye sonsuz teşekkür ve saygılarımı sunarım. Eğitim hayatım boyunca bilgi ve tecrübeleriyle bana destek olan başta değerli annem Gül CANKURTARAN ve babam Tuncay CANKURTARAN olmak üzere desteklerini esirgemeyen değerli Ekip Liderim Evren İLDEN'e teşekkürü bor bilirim.

# İÇİNDEKİLER

## Sayfa

ÖZET .....	i
ABSTRACT .....	iii
TEŞEKKÜR .....	v
1. GİRİŞ.....	1
1.1. Problemin Tanımı.....	1
1.2. Önerilen Metot.....	2
1.3. Özellik Ağacı Oluşturma Kriteri .....	4
1.4. Tez Organizasyonu .....	5
2. ÖN BİLGİ.....	6
2.1. Yeniden Kullanım ve Yeniden Kullanılabilirlik .....	6
2.2. OSGi Çerçevesi (İng. Open Services Gateway initiative Framework).....	7
2.2.1. Java ve Modülerlik .....	7
2.2.2. OSGi Teknolojisi .....	7
2.2.3. OSGi Çerçevesi Alternatifleri .....	8
2.3. Ürün Hattı Yaklaşımı .....	9
2.4. Yeniden Kullanılabilir Yazılım Bileşenlerinin Yaşam Döngüsü .....	11
2.5. Hedef-Soru-Metrik Yaklaşımı .....	16
2.6. Tez Kapsamında Kullanılan Metrikler ve Ölçüm Araçları .....	17
2.6.1. Sınıfın Ağırlıklı Metot Sayısı (İng. Weighted Methods Per Class - WMC) ...	19
2.6.2. Metotlardaki Uyum Eksikliği (İng. Lack of Cohesion Of Methods - LCOM) .....	19
2.6.3. Çevrimsel Karmaşıklık (İng. Cyclomatic Complexity- CC) .....	19
2.6.4. Ortalama Döngüsel Bağımlılık (İng. Average Cycling Dependency - ACD) .....	21
2.6.5. Ölçüm Araçları .....	21

2.7.	Nesneye Yönelik Yazılım Metriklerinin Yorumlanması .....	21
3.	İLİŞKİLİ ÇALIŞMALAR .....	24
3.1.	Yeniden Kullanılabilirlik Faktör ve Metrikleri .....	24
3.1.1.	Araştırma Hedefi ve Soruları .....	24
3.1.2.	Araştırma Stratejisi .....	25
3.1.3.	Seçim Kriteri .....	25
3.1.4.	Makalelerin Kalite Değerlendirmesi .....	26
3.1.5.	Veri Çıkarım .....	26
3.1.6.	Araştırma Sonuçları .....	26
3.2.	Bileşenlerin Yeniden Kullanılabilirliği .....	29
3.3.	Yazılım Ürün Hattının Yeniden Kullanılabilirliği .....	35
4.	DURUM ÇALIŞMALARININ TASARIMI .....	39
4.1.	Yeniden Yapılandırma için Özellik Ağacı Kullanılarak Yetenek Seçimi .....	42
4.2.	Araştırma Soruları .....	44
5.	DURUM ÇALIŞMALARININ BULGULARI .....	46
5.1.	Durum Çalışması-1'in Bulguları .....	45
5.2.	Durum Çalışması-2'nin Bulguları .....	47
5.3.	Durum Çalışması-3'ün Bulguları .....	49
5.4.	Durum Çalışması-4'ün Bulguları .....	50
5.5.	Ortalama Çevrimsel Bağımlılığa Yönelik Bulguların Anlamlılık Analizi .....	51
6.	BULGULARA YÖNELİK DEĞERLENDİRME .....	56
6.1.	Nicel Değerlendirme .....	56
6.2.	Nitel Değerlendirme .....	57
7.	GEÇERLİLİĞE YÖNELİK TEHDİTLER .....	60
7.1.	İçsel Tehditler .....	60
7.2.	Yapısal Tehditler .....	60
7.3.	Sonuçla İlgili Tehditler .....	60

7.4. Dışsal Tehditler .....	61
8. SONUÇLAR .....	62
9. GELECEK ÇALIŞMALAR .....	64
10. EK-1 Sistemik Haritalama ile İncelenen Araştırma Soruları ve Yanıtları .....	65
11. EK-2 Sistemik Haritalama ile İncelenen Çalışmalar ve Metrikler .....	66
14. EK-3 Özellik Ağacı .....	69
15. EK-4 Anket Çalışması .....	81
16. EK-5 Anket Sonuçları .....	83
17. KAYNAKLAR .....	87
18. ÖZGEÇMİŞ .....	92

## ŞEKİLLER

	<u>Sayfa</u>
Şekil 1.1 Metodun süreç şeması .....	3
Şekil 2.1 Ürün hattı yazılım süreci .....	11
Şekil 2.2 Yeniden kullanım için geliştirme süreci .....	15
Şekil 2.3 Yeniden kullanım ile geliştirme süreci .....	16
Şekil 2.4 Kontrol akış grafiği .....	20
Şekil 3.1. Araştırma stratejisi .....	25
Şekil 3.2 Makalelerin yıllara göre dağılımı .....	27
Şekil 4.1 Durum çalışması adımları .....	39
Şekil 4.2 Gömülü durum tasarımı .....	41

## ÇİZELGELER

	<u>Sayfa</u>
Çizelge 2.1. Yeni sistem için değerlendirme kontrol listesi .....	14
Çizelge 2.2. Çevrimsel karmaşıklık ve risk .....	20
Çizelge 3.1. Dahil etme ve hariç tutma kriterleri .....	26
Çizelge 3.2. Yeniden kullanılabilirliği etkileyen özelliklerin frekansı ve çalışmalar	28
Çizelge 4.1. Faktör ve metrikler .....	43
Çizelge 4.2. Proje yetenek eşleşmesi.....	44
Çizelge 5.1. Durum çalışması-1 için yapılandırma öncesi ve sonrası metrikler ....	47
Çizelge 5.2. Durum çalışması-2 için yapılandırma öncesi ve sonrası metrikler ....	48
Çizelge 5.3. Durum çalışması-3 için yapılandırma öncesi ve sonrası metrikler ....	49
Çizelge 5.4. Durum çalışması-4 için yapılandırma öncesi ve sonrası metrikler ....	50
Çizelge 5.5. Yapılandırma öncesi ve sonrası ACD metriği .....	51
Çizelge 5.6. Anlamlılık hesaplama algoritması .....	52
Çizelge 6.1. Çoklu durum çalışmasında incelenen projelerin LCOM, WMC ve CC metriklerinin yapılandırma öncesi ve sonrasındaki değerleri .....	57

## SİMGELER KISALTMALAR

### Simgeler

$\sqrt{\quad}$	Kök
$\Sigma$	Toplam Sembolü
$  \quad  $	Mutlak Değer

### Kısaltmalar

ACD	Average Cycling Dependency
AS	Araştırma Sorusu
CBO	Coupling between Objects Classes
CBSD	Component Based Software Development
CC	Cyclomatic Complexity
CiT	Cihaz İçi Test
CK	Chidamber & Kemerer
DIT	Depth Of Inheritance Tree
GQM	Goal Question Metric
HHF	Hava Hava Füzesi
IFF	Identification Foe or Friend
LCOM	Lack of Cohesion Of Methods
NATO	North Atlantic Treaty Organization
NLP	Neuro Linguistic Programming
NOC	Number Of Children
OSGi	Open Services Gateway initiative Framework
RFC	Response For a Class
RSY	Radar Sistem Yöne
SEI	Software Engineering Instute

# 1. GİRİŞ

## 1.1. Problemin Tanımı

Yazılım kalitesini arttırma, süreç riskini düşürme ve kaynakları verimli kullanma ihtiyaçları, yeniden kullanım kavramını beraberinde getirmektedir. Sürekli test edilen bileşenlerin yeniden kullanılması ile yazılımlar hatalardan arınmakta, projeler zamanında teslim edilerek süreç riski düşürülmektedir [1].

Yeniden kullanım birçok yazılım geliştirme yaklaşımında uygulansa da Yazılım Ürün Hattı ve Bileşen Tabanlı Geliştirme yaklaşımlarının temelini oluşturmaktadır. Literatürde bu yaklaşımlarla geliştirilmiş yazılımların yeniden kullanılabilirliğini değerlendirmek için farklı yöntem ve metrikler sunulmaktadır. Önerilen metrik ve metotların birçoğu sınıf seviyesinde kalmakta olup yazılımın yeniden kullanılabilirliği farklı faktörlere bakılarak farklı metriklerle değerlendirilmektedir. Yeniden kullanılabilirliği ölçmede kullanılan bu yaklaşımların çoğu, yeniden kullanılabilirliği beyaz kutu yaklaşımı ile ele almaktadır. Sonuç olarak yeniden kullanılabilirliğin sayısallaştırılması için kullanılan metot ve yaklaşımlarda tutarlılık sağlanamamakta, yüksek seviyeden ölçüm alma konusunda yapılan çalışmaların eksikliği literatürde görülmektedir [2].

Çalışma, bir sistem geliştirme ve entegrasyon firmasının yazılım müdürlüğünün Kullanıcı Arayüzü Ekibi bünyesinde geliştirilen radar kullanıcı arayüzü yazılımlarında gerçekleştirilmiştir. OSGi servis platformu ve Ürün Hattı Yaklaşımı'na dayalı mimarisi olan radar kullanıcı arayüzü yazılımlarımızda yeniden kullanılabilirliği değerlendirmek ve araştırmak için literatürde var olan çalışmalar sistematik olarak araştırılmıştır. Yapılan araştırmaların sonucuna göre OSGi servis platformu ve Ürün Hattı mimarisine dayalı olarak geliştirilmiş nesneye yönelik bir yazılımın yeniden kullanılabilirliğini değerlendiren metrik ya da metoda rastlanmamıştır [2].

Kendi özelimizdeki bu probleme çözüm getirmek ve literatürdeki bu boşluğu doldurmak için yeniden kullanılabilirliği arttırmak üzere Özellik Ağacı metoduna başvurulmuştur.



## 1.2. Önerilen Metot

Bu tez çalışmasında önerilen metot ile OSGi çerçevesinde geliştirilen bileşenlerin ve bu bileşenlerin oluşturduğu yazılım ürün hattının yeniden kullanılabilirliğinin artırılması hedeflenmektedir.

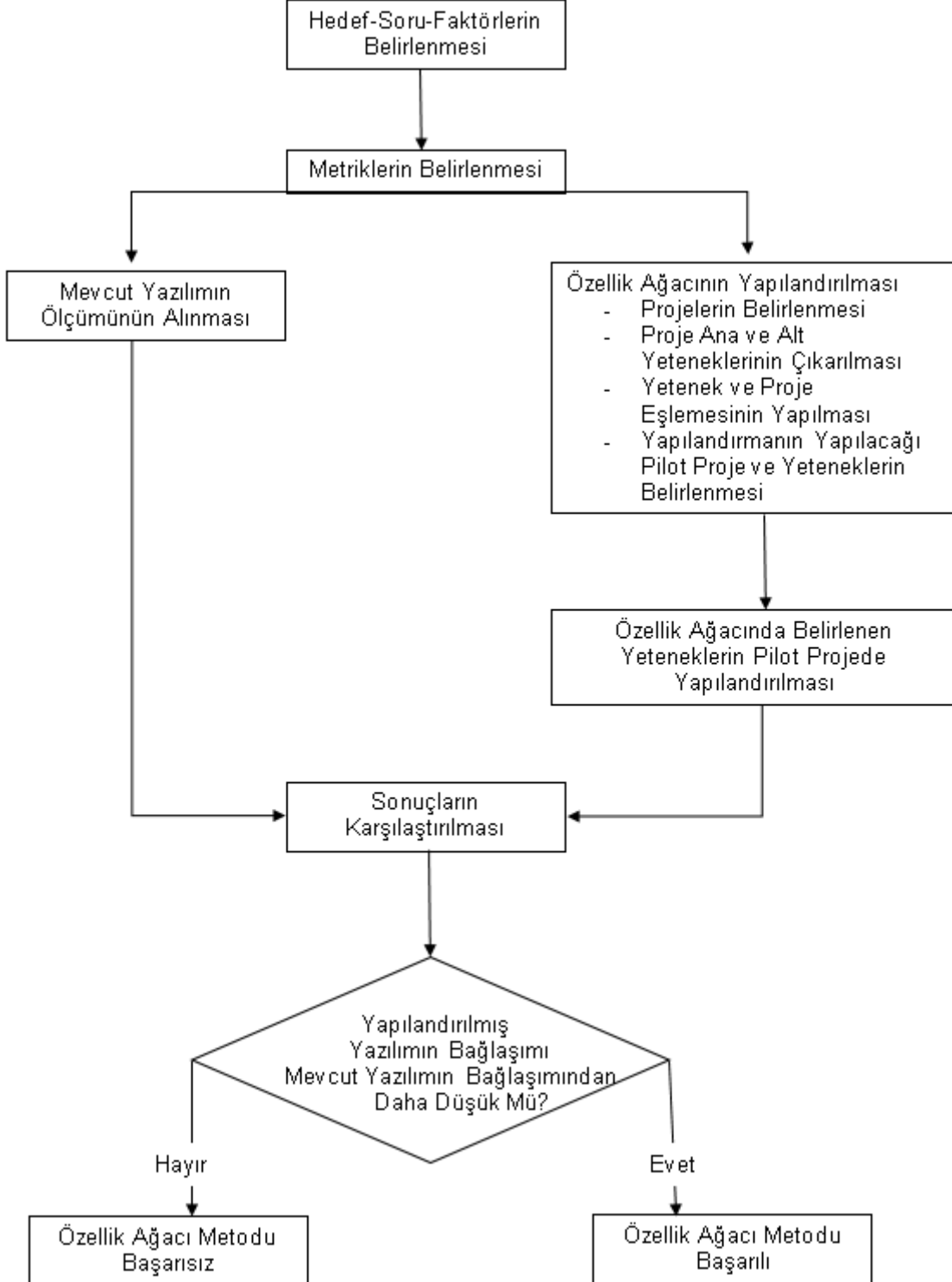
Çalışmada, öncelikle Sistemik Literatür Taraması yapılarak literatürde yeniden kullanılabilirliği etkileyen faktörlerin ve yeniden kullanılabilirliği ölçen metriklerin kullanılma sıklıkları çıkarılmıştır. Frekansı en yüksek olan faktör ve bu faktörü sayısallaştıran metrik seti belirlenmiştir [2].

Önerilen metotta, OSGi alt yapısını kullanan projelerden ve bu projelerin ana-alt yetenek setlerinden çıkarılan Özellik Ağacı kullanılarak yeniden kullanılabilirliğin artırılması hedeflenmiştir. Yazılım Gereksinim Özelliği (İng. Software Requirement Specification) dokümanından özelliklerin tanımlanıp çıkarılması ile özellik ağacı oluşturulmuştur. Çıkarılan özellik ağacı, iki geliştirici tarafından karşılıklı olarak gözden geçirilmiştir. Yetenekler ve projeler eşleştirilmiştir. Özellik Ağacı modelinden, birden çok proje tarafından içerilen özellikler arasından, yeniden yapılandırılacak yeteneklerin ve pilot projenin seçimi yapılmıştır. Seçilen yetenek setlerine göre yazılım yeniden yapılandırılmıştır. Kodu yapılandırmak için Parçala-Yönet (İng. Divide and Conquer) metodundan faydalanılmıştır [3].

Önerilen metodu doğrulamaya yönelik olarak dört farklı radar kullanıcı arayüzü yazılımı üzerinde Çoklu Durum Çalışması (İng. Multiple Case Study) yapılmıştır [82]. Önerilen metodu doğrulamak için durum çalışmaları kapsamında, belirlenen metrik seti kullanılarak yazılımların, metodun uygulanmasından önceki ve metodun uygulanmasından sonraki ölçümleri alınmıştır. Durum çalışmaları üç araştırma sorusu ile adreslenmiştir. İlk araştırma sorusu kapsamında, mevcut yazılımın bağlaşımı, uyumluluğu ve karmaşıklığı ölçülmüştür. İkinci araştırma sorusunda ise özellik ağacı ile bileşenlerdeki ortak ve projeye özel yetenekler çıkarılarak yazılım yeniden yapılandırıldıktan sonra yazılımın bağlaşımı, uyumluluğu ve karmaşıklığı ölçülmüştür. Üçüncü araştırma sorucu ile de araştırma sorusu 1 ve 2'nin sonuçları karşılaştırılarak, yeniden yapılandırma sonrasında yazılımın yeniden kullanılabilirliğinin artıp atmadığı gözlemlenmiştir [3].

Ölçümler, Java programlama dilini destekleyen statik kod analiz aracı olan SonarGraph Explorer aracı ve Understand aracı ile çevrim dışı (İng. offline) olarak alınmıştır [3].

Çalışmada uygulanan metodun süreç şeması Şekil 1.1’de verilmektedir.



Şekil 1.1. Metodun süreç şeması

### 1.3. Özellik Ağacı Oluşturma Kriteri

Bu bölümde, yeniden kullanılabilirliği iyileştirmek için temel alınan özellik ağacı metodunun tekrarlanabilir ve başka çalışmalarda uygulanabilir olması için, hangi kriterlere dayandırılarak çıkarıldığının bilgisi verilmektedir. Ayrıca özellik ağacının detay seviyesi ve kalitesi hakkında da aktarım yapılmaktadır.

Literatürde Yazılım Gereksinim Özelliği dokümanlarından özellik çıkarmak için farklı yaklaşımlar önerilmektedir. Hamza ve Walker [83] yazılım ürün hattında kullanılmak üzere yazılım gereksinim dokümanından özellik çıkarmak için, doğal dil işleme tekniklerinden (İng. Natural Language Processing - NLP) olan FFRE metodunu uygulamıştır. Bu teknik ile dokümandan özellik ve bu özellikler arasındaki ilişkiler çıkarılmıştır. Wang yaptığı çalışmada [84], alan mühendisliğinde kullanılmak üzere yazılım gereksinim özelliği dokümanından otomatik özellik çıkarma için, makine öğrenmesi tekniğini kullanarak semantik bilgi çıkarmıştır. Bakar ve arkadaşları [85], özellik dokümanından özellik çıkarmak için, yarı otomatik bir yaklaşım sunmuştur. Sunulan yaklaşıma göre doğal dil işleme tekniği kullanılarak ürün hattının özellik seti elde edilmiştir. Li ve arkadaşlarının sunduğu çalışmada [86] ise bilimsel yazılımlardaki dokümandan tanımlı örüntüler eşleştirilerek doğal dil işleme metodu ile gereksinimler çıkarılmıştır.

Tez çalışması kapsamında gerçekleştirilen yazılım gereksinimi çıkarma işlemi, yazılımın yeniden kullanılabilirliğini arttırmak için, yazılım gereksinim özelliği dokümanı üzerinden manuel olarak gerçekleştirilmiştir. Uzman görüşünden faydalanılarak çıkarılan ağacın derinliği ana yeteneklerin altındaki alt yeteneklerin derecesi bir olacak şekilde belirlenmiştir. Bu kararın alınması, tez kapsamında gerçekleştirilen yetenek proje eşleşmesinin kolay ve uygulanabilir olması isteğine dayanmaktadır. Manuel olarak uzman görüşü ile dört farklı projenin Yazılım Özellikleri Dokümanlarında yetenek setleri çıkarılarak özellik ağacı oluşturulmuştur. Çıkarım işlemi sonrasında, 46 ana yetenek ve bu ana yeteneklerin altındaki 188 alt yetenek ile özellik ağacı yapılandırılmıştır. Özellik ağacının özellikleri ile proje-özellik (yetenek) eşleşmesi, akran gözden geçirme (İng. peer review) kullanılarak doğrulanmıştır. Böylece eksiksiz ve hatasız bir şekilde özellik ağacının çıkarılması sağlanmıştır.

#### **1.4. Tez Organizasyonu**

Bu tez çalışmasının geri kalan kısmı şu şekilde oluşturulmuştur: 2. Bölümü'nde; OSGi, Yazılım Ürün Hattı gibi temel kavramlar, yeniden kullanılabilir bileşenlerin yaşam döngüsü, Hedef Soru Metrik yaklaşımı, nesneye yönelik metrikler, tez kapsamında kullanılan metrikler ve ölçüm araçları aktarılmaktadır. 3. Bölümü'nde literatürde sunulan yeniden kullanılabilirlik faktör ve metrikleri açıklanmaktadır. Bu bölümde ayrıca, bileşenlerin yeniden kullanılabilirliği ve yazılım ürün hattının yeniden kullanılabilirliği konularında literatürde yapılmış çalışmalar sunulmaktadır.

4. Bölümü'nde yapılan durum çalışmalarının bağlamı ve durum çalışmasını yönlendiren araştırma soruları ve durum çalışmalarının tasarımı açıklanmaktadır.

5. Bölümde 4. Bölümde tasarımı verilen durum çalışmalarına ait bulgular açıklanmaktadır. 6. Bölümde 5. Bölüm 'deki bulguların sonuçları değerlendirilmektedir. Çalışmanın genel sonuçlarına yer verilmekte geçerliliğe yönelik tehditler (İng. validity threats) tartışılmaktadır.

## 2. ÖN BİLGİ

### 2.1. Yeniden Kullanım ve Yeniden Kullanılabilirlik

Yeniden kullanımın tarihsel sürecine bakıldığında, en basit ve ilk yeniden kullanım pratiğinin fonksiyon bazında yeniden kullanım olduğu görülmektedir. İlerleyen süreçte, geliştiricilerin bilgisinden soyutlanmış kütüphanelerin yeniden kullanımı gündeme gelmiştir. Yeniden kullanımda nesneye yönelik geliştirme ise devrim niteliği taşımaktadır [4].

Yazılımda yeniden kullanılabilirlik kavramını anlamak için yeniden kullanım ve yeniden kullanılabilirliği ayırt etmek gerekir. Frakes ve Succi [4] yeniden kullanımı (İng. reuse), “var olan yazılım ve yazılım ürünlerinin yeni yazılım geliştirmede direkt kullanımı” olarak tanımlamıştır. Yeniden kullanım, daha önceden yazılmış yazılım ürünlerinin kullanımudur. Aynı araştırmacılar yeniden kullanılabilirliği (İng. reusability) ise “yeniden kullanım olasılığını gösteren yazılım özelliği” olarak tanımlamaktadır. Yani yeniden kullanılabilirlik, yazılım parçasının başka bir uygulamada yeniden kullanılmasını mümkün kılan kalite özelliğidir. Yeniden kullanılabilirlikte, yazılım parçasının kısmi değiştirilerek ya da tamamen kullanımı söz konusu olabilir. Diğer bir ifade ile yeniden kullanılabilirlik, yazılım ürününün başka bir bağlamda kullanımının kolaylık derecesi olarak tanımlanmaktadır [4].

Yeniden kullanılabilirliğin literatüre kazandırılması, 1968 yılında McIlroy’un yeniden kullanılabilir bileşenleri NATO seminerinde sunmasına dayanmaktadır [5]. Ne var ki pratikte yeniden kullanılabilirliğe ilişkin çalışmalar 1980’lerin sonuna kadar sınırlı kalmıştır. 1990’ların sonunda yeniden kullanılabilirlik ile ilgili yürütülen çalışmalar hız kazanmıştır. Milenyuma giriş, beraberinde birçok bileşenin geniş bir bantta yeniden kullanılmasını beraberinde getirmiştir [5].

Yeniden kullanım tüm yazılım geliştirme yaklaşımlarında esas olsa da bileşen tabanlı yazılım geliştirmenin (İng. Component Based Software Development - CBSD) ve Ürün Hattı Yaklaşımı’nın (İng. Product Line Approach) ana kavramıdır. Bileşen tabanlı yazılım geliştirme, yeniden kullanılabilir bileşenler kullanarak yazılım tasarlamının altını çizmektedir [6]. Yazılım ürün hattı ise belirli bir görevin ihtiyacını karşılamak üzere, ortak yetenek kümelerinden oluşan sistemlerin kullanımını esas almaktadır [7]. Bileşenlerin ve ürün hattı yaklaşımlarının yeniden

kullanılabilirliğini deęerlendirmek için yazılım karakteristikleri, metrikler kullanılarak sayısallaştırılmaktadır.

Üretkenlięin ve kalitenin artması, yeniden kullanımın iki temel amacıdır. Üretkenlik, üretim hızını arttırıp maliyeti ve geliştirme eforunu düşürmektedir. Kalite ise, yeni tasarlanmış, iyi test edilmiş ve iyi dokümente edilmiş yazılım ürünlerini kullanarak anlaşılabilirlięi ve doęruluęu arttırmaktadır.

## **2.2. OSGi Çerçevesi (İng. Open Services Gateway initiative Framework)**

Tez çalışması kapsamında yeniden kullanılabilirlięinin arttırılması hedeflenen radar kullanıcı arayüzü yazılımları, OSGi çerçevesi kullanılarak ve Java programlama dili ile geliştirilmiştir [8].

### **2.2.1. Java ve Modülerlik**

Tek başına Java, modülerlięi tam anlamı ile desteklememektedir. Bartlett'e [8] göre OSGi çerçevesi, Java'daki problemlere ve modülerlięe çözüm sağlamaktadır. İş bölümü, soyutlama, yeniden kullanılabilirlik, kolay bakım ve onarı,m modülerlik ile sağlanan kazançlardır [8].

Yazılımda bir yapının bileşen olabilmesi için bileşenin; bir bütün olarak mantıksal olup atom olmaması, bağımsız bir birim olarak indirilip yüklenebilir olması, mantıksal bir amaca baęlı kalması, modülün etkileşimde olduęu dięer modüllerin gerçekleştirim detaylarına baęlı olmaması gerekmektedir [8].

Örneęin, aynı anda jar dosyasının birden çok versiyonunun yüklenememesi, hangi jar dosyasının hangi versiyona baęımlı olduęunun belirtilememesi, bağımlılıkların belirtildięi standart META-DATA bilgisinin içermemesi, jar dosyalarının bilgi saklama (İng. information hiding) mekanizmasının olmaması, Java jar dosyalarının kullanımında karşılaşılan problemler arasında tanımlanmaktadır [8].

### **2.2.2. OSGi Teknolojisi**

OSGi teknolojisi Java için bir bileşen sistemi sunmaktadır. Gerçek anlamda bir bileşen oluşturulması ve bu bileşenlerin çalışma zamanında birbiri ile etkileşimde bulunabilmesi için yöntem sunar [8].

Java'da birçok sorunun kaynaęı, global ve düz olan sınıf yolundan (İng. classpath) dolaydır. OSGi bu soruna tamamen farklı bir yaklaşım sunmaktadır. OSGi

teknolojisine göre her bileşen kendi sınıf yoluna sahip olup sınıfların bileşenler arasında nasıl paylaşılacağına dair içeri aktarma (İng. import) ve dışarı aktarma (İng. export) mekanizmaları kullanılmaktadır [8].

OSGi'de bileşen, yığın (İng. bundle) olarak da adlandırılır. Yığınlar da aslında birer jar dosyasıdır; yani yeni bir standart değildir. Sadece jar dosyasına META-DATA eklenerek yığın haline gelir [8].

Yığın adı, yığın versiyonu, içeri ve dışarı aktarım bilgileri ve yığının minimum hangi Java versiyonu ile çalışacağı bilgisi, META-DATA bilgilerini içeren MANIFEST.MF dosyasında tutulmaktadır. İçe aktarım ve dışa aktarım bilgilerinin eşlenip eşlenmediğinin kontrolü, OSGi çerçevesi tarafından yapılmaktadır. Bu işleme çözümüleme süreci (İng. re-resolution process) denir [8].

Ayrı sınıf yolu mekanizmasında her bir yığın için ayrı bir sınıf yükleyici (İng. class loader) sağlanmaktadır. OSGi'de ağaç (İng. tree) yapılı değil, çizge (İng. graph) yapılı sınıf yükleyici (İng. Class Loader) tanımlıdır. Yani iki yığın arasındaki bağımlılık hiyerarşik değildir. Klasik olarak Java'da ClassLoader çok uzun bir sınıf listesinde arama yaparken OSGi ClassLoader'ları sınıfı nerede bulacağını bilir [8].

OSGi'de versiyonlama mümkün olmaktadır. Paketin versiyonuna göre bağımlılıklar tanımlanabilmektedir. Dışarı aktarılacak paket adları, versiyon özelliği ile tanımlanırken içeri aktarma, versiyon aralığı ile tanımlanmaktadır [8].

OSGi dinamik bileşen sistemidir. OSGi'de bileşenler uygulamanın tümünü etkilemeden güncellenebilir, kurulabilir ve kaldırılabilir. Bu, sunucudaki bir uygulamanın bir parçasını değiştirmeyi ya da yeni bir parça eklemeyi, diğer parçalar etkilenmeden yapabilme imkanı sağlamaktadır [8].

Şu anki Eclipse, bir OSGi uygulaması olan Equinox tabanlıdır; OSGi spesifikasyonunun 4.1 sürümünün gerçekleşmiş halidir. Knopflerfish, OSGi spesifikasyonunun 3 ve 4.1 sürümlerinin gerçekleşmiş halidir. Felix de OSGi çerçevesine bir örnektir [8].

### **2.2.3. OSGi Çerçevesi Alternatifleri**

Moven and Ivy çerçevesi, OSGi çerçevesine alternatif olarak görülmektedir. Bir yapılandırma zamanı (İng. build time) aracıdır. Bu sebeple, genel sınıf yolu, bilgi

saklama vb. sorunlara çözüm getirememektedir [8]. OSGi temelli uygulama geliřtirmede yardımcı araç olarak kullanılabilir.

Eclipse eklenti sistemi (İng. Eclipse plugin system) OSGi'ye alternatif görülebilecek başka bir sistemdir. Eclipse'nin en son kullandığı alt yapı Equinox'dur. Ancak Eclipse 3.0'dan önce kendi bileşen sistemine sahiptir. Eclipse'de bileşene "eklenti" (İng. plugin) denilmektedir. Her eklenti, OSGi MANIFEST.MF dosyasına çok benzer bir plugin.xml dosyasına sahiptir. OSGi'de bağımlılıklar paket seviyesinde tanımlanırken Eclipse eklentilerdeki bağımlılıklar, tüm eklenti bazında tanımlanmaktadır. Böylece eklentideki tüm paketlere erişilmektedir. Eclipse eklenti sistemindeki en büyük eksiklik, eklentilerin dinamik olarak kurulup kaldırılamamasıdır. Eklenti çizgesi deęişir deęişmez yeniden başlatmak gerekmektedir [8].

Sonuç olarak, Java řu anki yapısıyla dinamik bileşen sistemini desteklememektedir. OSGi teknolojisi ile dinamik bir bileşen sistemi oluşturulmakta, Java'daki sınıf yolu ve jar gibi problemlere çözüm getirilmektedir [8].

### **2.3. Ürün Hattı Yaklaşımı**

Radar kullanıcı arayüzü yazılımının geliřtiriminde faydalanılan bir dięer yaklaşım Yazılım Ürün Hattı'dır. Yazılım Ürün Hattı Mühendislięi gerçekenirken tüm ürün hattı için, deęişken ve ortak olan yetenekler bir araya getirilerek yeniden kullanımın artırılması amaçlanmaktadır [9].

Radar ihtiyacına yönelik özel ihtiyaçları karşılamak üzere geliřtirilen radar kullanıcı arayüzü yazılımlarında, ortak ve yönetilen bir özellik grubunu destekleyen çekirdek bir varlık kümesinin kullanımını esas alan ürün hattı yaklaşımı kullanılmıştır. Radar uygulama alanında geliřtirilen, operatörün radar ile etkileşimini ve kontrol etmesini sağlayan radar kullanıcı arayüzü yazılımları, radar sisteminden elde edilen tespit ve takip verilerinin sunulmasını ve radarın kontrol edilmesini sağlamaktadır. Gerçek zamanlı çalışan radar sisteminde, aynı anda farklı arayüzlere sahip olunması bu yazılımların karmaşıklığını arttırmaktadır. Radar ürün ailesinde yer alan farklı radar sistemlerinin kullanıcı arayüzü yazılımlarının yeniden kullanılabilirlięi, yazılım geliřtirme sürecinin kısaltılarak eforun düşürülmesi ve riskin minimize edilmesi için önemli bir kalite gereksinimi olarak görülmektedir.



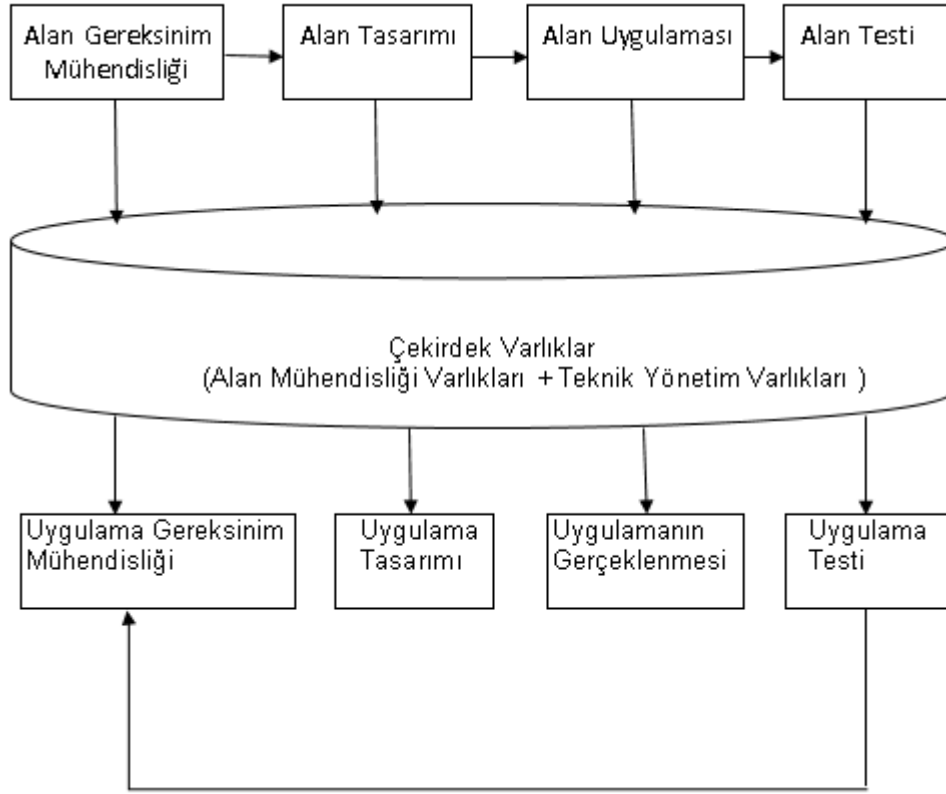
Yazılım Ürün Hattı, “özgün bir pazarın özel ihtiyaçlarını karşılamak üzere, ortak ve değişken yönetilebilir özellik gruplarını destekleyen yazılım yoğun sistemler” olarak tanımlanmaktadır [9]. Şekil 2.1’de, yazılım ürün hattının döngüsel süreci sunulmaktadır.

Yazılım Ürün Hattı yaklaşımı, 1970’lerde Ürün Ailesi kavramı ile ortaya çıkmaya başlamış, 1990’larda ise tamamen tanımlanıp kullanıma geçirilmiş bir yeniden kullanım yaklaşımıdır. Maliyeti ve proje geliştirme süresini düşürürken yazılım kalitesini arttırmayı hedeflemektedir. Olgunlaşmış ve doğrulanmış yetenekler ile daha kaliteli yazılımlar ortaya çıkarılmaktadır. Yaklaşım, yazılım geliştirme süreci boyunca büyük ölçekli yeniden kullanıma dayanmaktadır [10].

Yazılım ürün hattı yaklaşımı, yeniden kullanım için geliştirme ve yeniden kullanımla geliştirme kavramları arasındaki farklılığı temel almaktadır. Yeniden kullanım için yazılım geliştirme, Şekil 2.1’de üst kısımda akışı verilen Alan Mühendisliği paydaşına karşılık gelmektedir. Alan mühendisliği ortak yeteneklerin farklı sistemlerde yeniden kullanımı sürecini kapsamaktadır. Uygulama Mühendisliği ise ürün hattında yeniden kullanım ile yazılım geliştirme paydaşına karşılık gelmektedir. Değişken yeteneklerin yönetimleri ve geliştirilmesi bu aşamada gerçekleşmektedir. Geliştirilen yazılımlarda ortak varlıkların doğrudan ya da değiştirilerek kullanılması söz konusu olduğu için, yeniden kullanılabilirlik üst seviyeye çıkarılmaktadır [9,10].

Özellik Ağacı, Yazılım Ürün Hattı yaklaşımının dayandırıldığı özellik modeli olup nihai ürünün karakteristik özelliklerini göstermektedir. Model, hiyerarşik olarak sunulmaktadır. Özellikler; zorunlu, opsiyonel, “veya”lanmış (İng. ORed) ya da harici “veya”lanmış (İng. XOR) olarak çeşitlenmektedir. Çoğu özellik modeli, farklı özellik tiplerinin kombinasyonları sonucunda oluşturulmaktadır [10].

Şekilde sunulan alan gereksinim mühendisliği sonucu Yapıtaşı, uygulama gereksinim mühendisliği sonucunda Projeye Özel olarak tanımlanan bileşenler geliştirilmektedir [10,11,12,13].



Şekil 2.1. Ürün hattı süreci

#### 2.4. Yeniden Kullanılabilir Yazılım Bileşenlerinin Yaşam Döngüsü

Sistemik yeniden kullanımındaki bilgi yetersizliği, yazılım mühendisliğinde karşılaşılan bir problem olarak tanımlanmaktadır. Tez kapsamında zemin bilgi oluşturması açısından yeniden kullanılabilir bileşenlerin yaşam döngüsü, bu başlık altında aktarılmaktadır [14].

Badareen ve arkadaşının yapmış olduğu çalışmada [14], yeniden kullanılabilir bileşen elde etmede ortaya çıkan yazılım bileşenlerinin yaşam döngüsü ele alınmaktadır. Çalışma, yeniden kullanılabilir bileşenlerin yaşam döngüsünü; yeniden kullanım ile yazılım geliştirme ve yeniden kullanım için yazılım geliştirme açılarından ele alan sistematik bir çerçeve sunmaktadır. Yapılan durum çalışması ile sunulan çerçeve doğrulanmaktadır.

Ramamoorthy'e göre yeniden kullanım kavramı sadece kaynak koddan ibaret olmayıp, ürünün oluşturulma sürecinin, gereksinimlerinin, tasarımının, analiz bilgilerinin, dokümanının yeniden kullanımını da kapsamaktadır [14].

Sommerville'e göre [14] yazılımı yeniden kullanmada karşılaşılan problem araçlardaki ve yeniden kullanılabilir kütüphanelerdeki eksiklik olarak tanımlanmaktadır. Çalışmaya göre planlama fazında, yeniden kullanılabilirliğin amacı tanımlanmaktadır. Gereksinim fazında, hangi bileşenin yeniden kullanılabilir olarak nitelendirileceği ve hangi bileşenin yeniden kullanılabilir olarak geliştirileceği gibi konularda karar verilmektedir. Bu fazla yeniden kullanım için geliştirilecek bileşenler belirlenmektedir.

Sistem analizi ve tasarımı fazında taşınabilirlik, bağlaşım ve uyumluluk konuları dikkate alınmaktadır. Bu karakteristikler, yeniden kullanılabilir bileşenler üretmek için ve bu bileşenlerin diğer bileşenler ile entegrasyonunu çalışmalarını sağlamak için dikkate alınmaktadır. Yani, bu tez kapsamında ölçümlerle değerlendirmeye alınan bağlaşım, uyumluluk ve karmaşıklık gibi yazılım özellikleri, yazılımın sisteminin tasarım fazında da dikkate alınmalıdır. Uygulama fazında, bağımsız alt sistemler geliştirilmekte ve bu alt sistemler birbirlerine entegre edilmektedir. Test fazında alt sistemler ve alt sistemlerin entegrasyonu değerlendirilmektedir. Sürecin sonunda iki yazılım ürünü ortaya çıkarılmaktadır. İlki piyasaya sunulan ürün olarak tanıtılırken, ikincisi yeniden kullanılabilir bileşen olarak kayıt öncesi sürece tabii tutulmaktadır.

Kayıt öncesi süreç belirli standartlara göre yeniden kullanılabilir bileşenlerin değerlendirilip geliştirildiği süreçtir. Sonuç olarak, yeniden kullanılabilirlik testine tabii tutulan bileşen başka sistemlerde kullanılmak üzere kütüphaneye aktarılmaktadır.

Yeniden Kullanım içinden Yeniden Kullanım İle'ye transfer için belli adımlar izlenmelidir. Yeniden kullanım için geliştirme ile yeniden kullanım ile geliştirme arasındaki bağlantıyı üç faz oluşturmaktadır [14]:

- Ön saklama süreci (İng. pre-store process)
- Ön kullanım süreci (ing. pre-use process)
- Yeniden kullanılabilir yazılım bileşenlerin depolaması (İng. reusable software component storage)

Yeniden kullanılabilir bir bileşenin, farklı platformda ve farklı bir sistemde kullanılıp kullanılmayacağı kararı proje yöneticisi tarafından verilmektedir [14].

Yeniden kullanılabilir bileşenlerin kütüphanesi, farklı sistemlerde kullanılan bileşenlerin kaynağı olarak görülmektedir. Depoda tutulan yeniden kullanılabilir bileşenlerin farklı standartları ve şartları sağlaması gerekmektedir. Dolayısı ile yeniden kullanılabilirlik testi yeniden kullanılmaya aday bileşenlerin standartları karşılayıp karşılamadığını test etmektedir. Bu testi geçen bileşen kütüphaneye gönderilmektedir. Aksi takdirde bileşenin standartları karşılaması için modifiye edilmesi gerekecektir [14].

Yazılımın farklı sistemlerde kullanılması için içermesi gereken farklı karakteristikler bulunmalıdır. Bu karakteristiklerin mevcudiyeti, yazılım bileşenlerinin kütüphaneye aktarılıp aktarılmamasında karar verdirici rol oynamaktadır. Yeniden kullanılabilirlik için taşınması gereken karakteristikler [14].

- Bir arada yaşama (İng. co-existence)
  - Sistem bağımsız çalışma (İng. system independence)
  - Makine bağımsız çalışma (İng. machine independence)
- Uygunluk (İng. compliance)
- Birlikte çalışabilirlik / adapte olabilirlik (İng. interoperability / adaptability)
  - Modülerlik (İng. modularity)
  - İletişim ortak noktası (İng. communication commonality )
  - Veri ortak noktası (İng. data commonality)
- Genellik (İng. generality)

Yazılımın genel karakteristiği, yazılımın soyutluk seviyesine karşılık gelmektedir. Yazılımların çok sayıda sınıf problemine, yüksek seviyeden “soyutluk” getirerek çözüm üretmektedir [14].

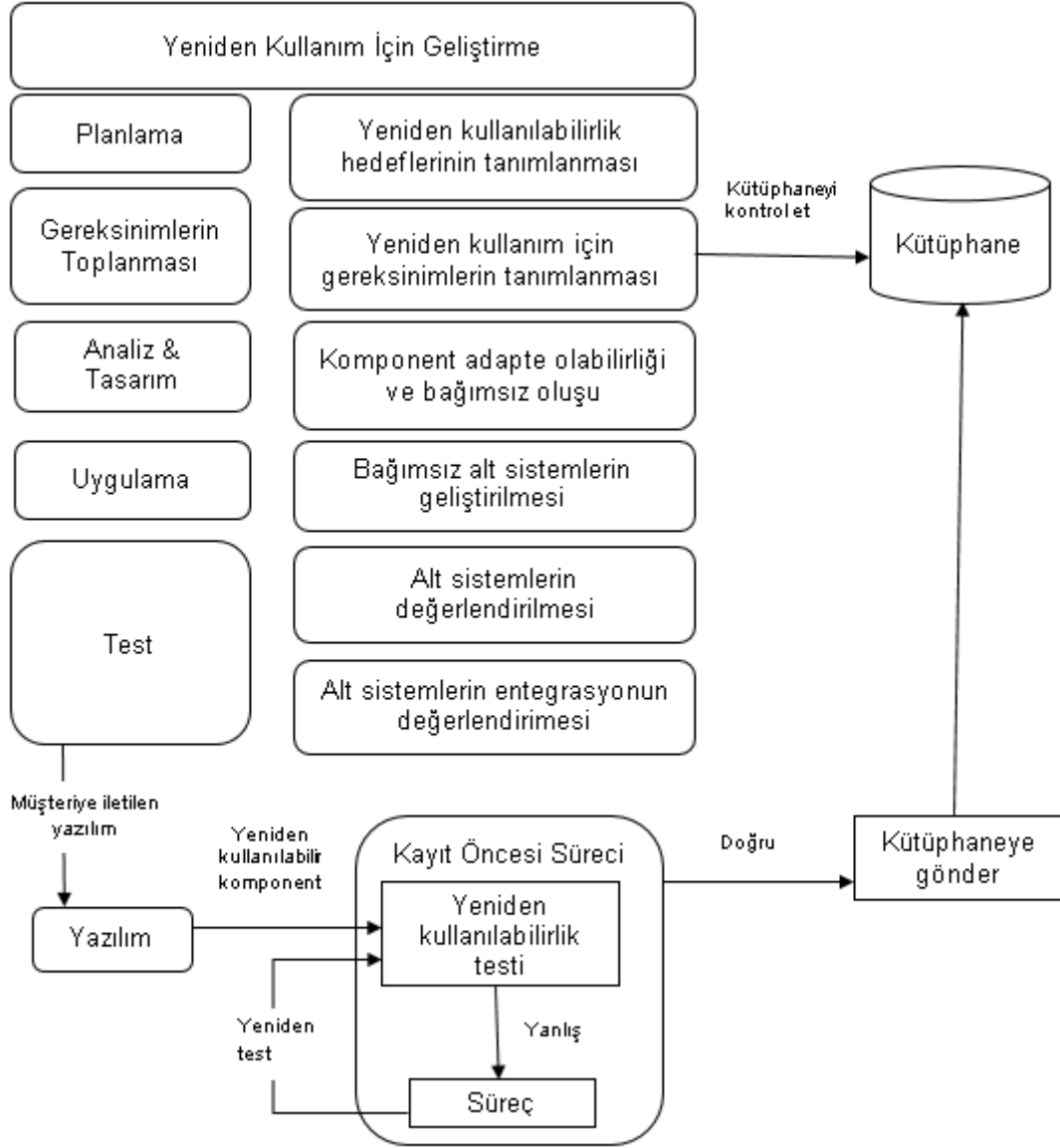
Uygunluk karakteristiği, yeniden kullanılabilir bileşenlerin geliştirilirken uluslararası standartların ya da herhangi bir standardın izlenip izlenmediğini doğrulamaktadır.

Çizelge 2.1, yeni sistem için ihtiyaç duyulan değerlendirmeye temel olan kontrol listesini göstermektedir [14].

Çizelge 2.1. Yeni sistem için değerlendirme kontrol listesi [14]

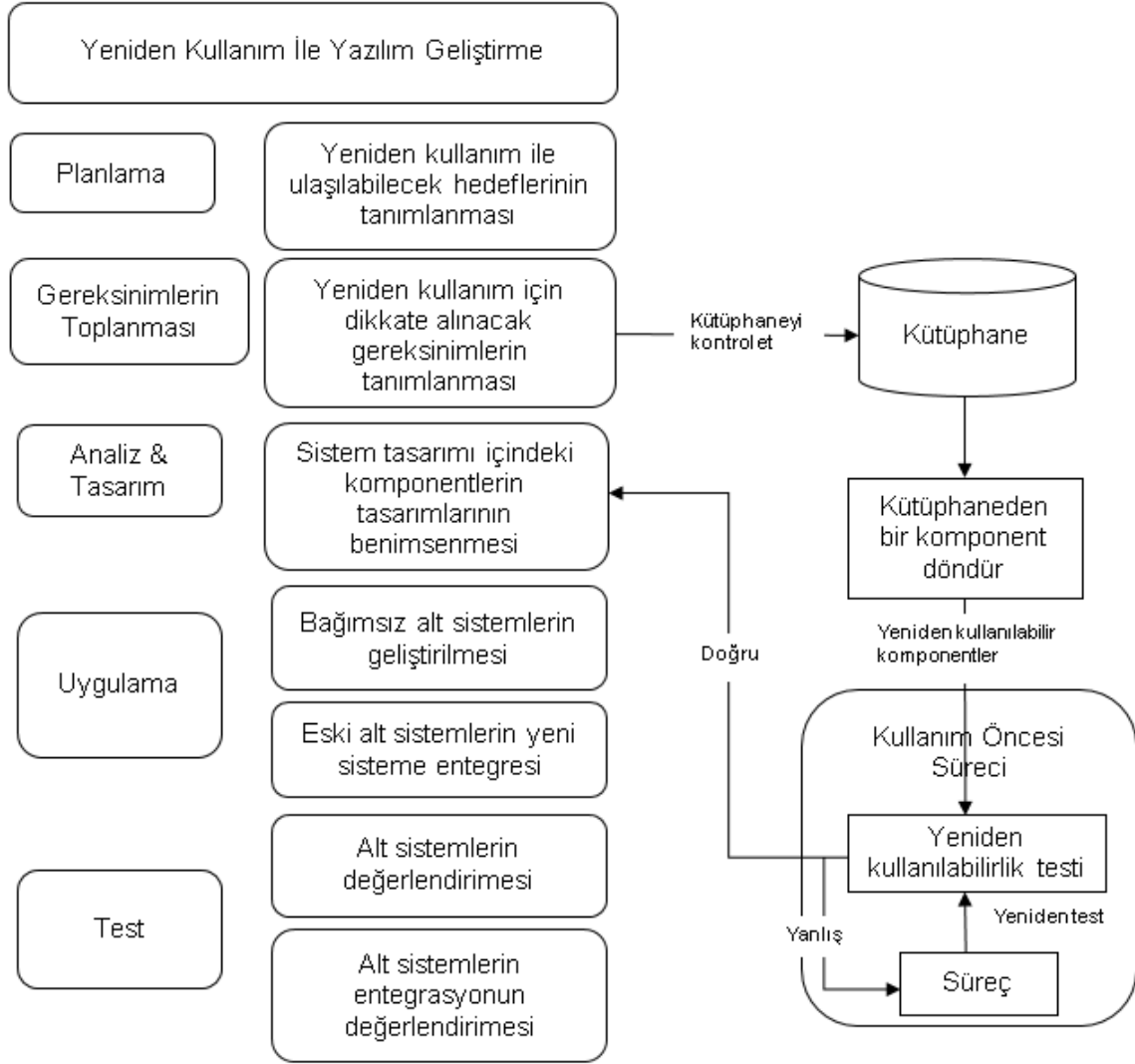
Karakteristik	Kriter	Soru
Fonksiyonellik	Uygunluk	Bileşen kullanıcı gereksinimleri karşılayabilir mi?
	Doğruluk	Bileşen sonuçlarının hassaslığı kullanıcı gereksinimlerini tatmin etmekte mi?
	Uyumluluk	İstenilen fonksiyonları yerine getirmek için bileşen herhangi bir standart izlemekte mi?
Bir arada çalışma	Sistem/Alt Sistem Bağımsızlığı	Bileşen yeni sistem ile çalışabilmekte mi?
	Makine Bağımsızlığı	Bileşen yeni makine ile çalışabilmekte mi?
Uyumluluk		Bileşen yeni sistemde ihtiyaç duyulan standartları izlemekte mi?
Adapte Edilebilirlik	Modülerlik	Bileşen alt sistemden bağımsız mı?
	İletişim Ortaklığı	Bileşen yeni sistem ile haberleşebilmekte mi?
	Veri Ortaklığı	Bileşen yeni sisteme veri gönderip alabilmekte mi?
Genellik		Bileşen yeni sistemin gereksinimlerini karşılamak için değiştirilebilir mi?

Şekil 2.2’de “yeniden kullanım için” geliştirme sürecinin işleyişi gösterilmektedir.



Şekil 2.2. Yeniden kullanım için geliştirme süreci [14]

Şekil 2.3.'te "yeniden kullanım ile" geliştirme sürecinin işleyişi gösterilmektedir.



Şekil 2.3. Yeniden kullanım ile yazılım geliştirme süreci [14]

## 2.5. Hedef-Soru-Metrik Yaklaşımı

Tez çalışmasında, metrikleri ve ilgili çalışmalarını sistematik olarak belirlemek için Hedef-Soru-Metrik (İng. Goal-Question-Metric) yaklaşımı uygulanmıştır. Hedef-Soru-Metrik yaklaşımı organizasyon için amaçlı ölçüm varsayımına dayanmaktadır. Amaçlı ölçümde organizasyon ilk olarak, kendine ve projelerine ait hedeflerini tanımlar. Özetle, organizasyonun elde etmek istediği veya ihtiyaç duyduğu bilgi ne ise bu bilgi sayısallaştırılır [15].

Hedef Soru Metrik uygulamasının sonucu, ölçüm verisini yorumlamak için belirli bir kurallar setini hedefleyen ölçüm sistemi olarak tanımlanmaktadır.

- i. Kavramsal Seviye (İng. Conceptual Level) (GOAL): Çeşitli bakış açılarından, çeşitli kalite modellerine göre, çeşitli nedenler için hedef tanımlanan seviyedir. Ölçülen nesne; x Ürün (dokümanlar, tasarımlar, kaynak kodları), Süreç (gereksinimleri belirleme süreci, tasarım süreç, geliştirme süreci, test süreci), x Kaynak (çalışan, donanım, yazılım, ofis) olabilir [15].
- ii. Operasyonel Seviye (İng. Operational Level) (QUESTION): Soru, ölçüm nesnesini karakterize etmeye çalışır. Hedefleri değerlendirme ve gerçekleştirme yolunu karakterize etmek için kullanılan sorular setidir [15].
- iii. Nicel Seviye (İng. Quantitative Level) (METRIC): Soruları sayısal olarak cevaplamak için, her soru ile ilişkili olan veri setidir. Metrik, objektif ya da sübjektif olabilir. Objektif metrikler bakış açısına bağlı olmaksızın sadece nesnenin kendisine bağlıdır (dokümanın sürüm numarası, program boyutu vb.). Sübjektif metrikler ise hem ölçümü alınan nesneye hem de bakış açısına bağlıdır (Örneğin kullanıcının memnuniyet derecesi) [15].

Hedef-Soru-Metrik mekanizması hedefle başlayan hiyerarşik bir yapıya sahiptir. Hedef, ölçüm alanının bakış açısı gözetilerek ölçümün konusu ve amacının belirlenmiş halidir. Ölçüm hedefi, birden çok soruya inceltir. Sorular, kimi objektif kimi ise sübjektif olan metriklere inceltir. Aynı metrik, farklı soruları yanıtlamak için aynı hedef altında kullanılabilir. Metrikler farklı bakış açılarına göre farklı değerler alabilir. Yöntem genel olarak yazılım kalitesini geliştirmek için kullanılır [15].

## **2.6. Tez Kapsamında Kullanılan Metrikler ve Ölçüm Araçları**

Ölçüm almak, yazılımın kalitesini kontrol altında tutmak için önemlidir. Yazılım metrikleri, ölçüm sonucunda sayısal çıktılar sunarak yazılım kalitesinin ölçümünde nesnel değerlendirme yapılmasını sağlar [16].

Goodman [17] yazılım metriklerini “ölçüm temelli tekniklerin yazılım geliştirme sürecine sürekli uygulanması, süreç ve sürecin ürünlerinin gelişmesi için bu tekniklerin kullanılmasıyla birlikte, anlamlı ve zamanında bilgi yönetimini sağlaması” ve Schulmeyer [18] ise “bir sistemin, bileşenin veya sürecin verilen özelliğe sahip olma derecesinin nicel ölçümü” olarak tanımlamıştır.



1994'te Chidamber ve Kemerer nesneye yönelik programlar için metrikler seti sunmuştur. CK metrikleri, tasarım ve yazılım kalitesini değerlendirmede önemli rol oynamaktadır. CK metrikleri ile sınıf seviyesinde yazılımın kalitesi değerlendirilmektedir [19]. Yazılım dünyasında nesneye yönelik geliştirilen programların ölçümünü yapmak ve öznel değerlendirmelerden bağımsız kalabilmek için birçok metrik önerilmiştir. Bunlardan en önemlisi ve en çok kullanılanı Shyam R. Chidamber ve Chris F. Kemerer tarafından önerilmiş olan C&K metrik setidir. Tez çalışması kapsamında C&K metrik setlerinden LCOM ve WMC nesne yönelimli metrikleri kullanılmıştır. Ayrıca Modified Cyclomatic Complexity, ACD, Fan-in ve Fan-out metrikleri kullanılan diğer metriklerdir [2]. Bu metrikler izleyen alt bölümde detaylı olarak açıklanmıştır.

C&K metrik seti içinde yer alan altı metrik aşağıdadır:

- Weighted Methods per Class (WMC): Bir sınıftaki toplam karmaşıklığı tanımlamaktadır. WMC metriği ile ölçülen sınıfın bakım ve geliştirme eforu hakkında fikir sahibi olunabilir [19].
- Reponse For a Class (RFC): Bir set içindeki büyüyen metotların sayısı olarak tanımlanmaktadır. Belirli bir görevi gerçekleştirmekte yükümlü olan nesneye gönderilen mesaja verilen cevaptır [19].
- Depth Of Inheritance Tree (DIT): Sınıfların hiyerarşisindeki derinliği tahmin etmek için kullanılmaktadır. Yeniden kullanılabilirlik ve bakım yapılabilirlikte harcanacak olası eforu tahmin etmede kullanılabilir [19].
- Number Of Children (NOC): Belirli bir sınıfla ilişkili olan sınıfların sayısı olarak tanımlanmaktadır. Bir sınıfın çok fazla ilişkili olduğu sınıf varsa kötü tasarıma sahip olduğu yorumu çıkarılmaktadır [19].
- Coupling between Objects Classes (CBO): Birbiri ile yapışık ilişki içinde olan sınıfların sayısı olarak tanımlanmaktadır. Yazılımın yeniden kullanılabilirliği ve test edilebilirliği hakkında bilgi vermektedir [19].
- Lack of Cohesion Of Methods (LCOM): Benzerliğin sıfır olduğu metotların sayısı ve benzerliğin sıfır olmadığı metotların sayısı arasındaki farklılık olarak tanımlanmaktadır [19].

### 2.6.1. Sınıfın Ağırlıklı Metot Sayısı (İng. Weigted Methods Per Class - WMC)

WMC metriği, bir sınıftaki metotların karmaşıklık derecesini ölçmektedir. Diğer bir tanımla, bir sınıfın tüm metotlarının karmaşıklığının toplamıdır. WMC metrik değerinin düşük olması beklenmektedir. Çünkü WMC değeri yüksek olan bir sınıfın uygulamaya özgü olma ihtimali yüksektir ve bu nedenle yeniden kullanılabilirliği düşürmektedirler [20].

$M_1, M_2, M_3, \dots M_n$  metotları sınıf C'nin yordamları olsun. Bu metotların karmaşıklığı  $C_1, C_2, C_3, \dots C_n$  ise bu C sınıfının, sınıf başı ağırlıklı metot sayısı Denklem 1'deki gibi hesaplanır [20]:

$$WMC = \sum_{k=1}^n C_k \quad (1)$$

### 2.6.2. Metotlardaki Uyum Eksikliği (İng. Lack of Cohesion Of Methods - LCOM)

Bu metrik n adet metot kümesinin kesişiminden oluşan kümelerdeki uyumsuzlukların sayısıdır. Benzerliğin sıfır olduğu metotların sayısı ve benzerliğin sıfır olmadığı metotların sayısı arasındaki farklılık olarak da tanımlanmaktadır. Sınıfın uyumluluğunun düşük olması, sınıfın alt parçalara bölünmesi gerektiğini göstermektedir. Düşük uyum karmaşıklığı arttıran bir özelliktir. Yüksek uyum ise metotlar arasındaki benzerliği göstermektedir [20].

Bu metrik değeri yüksek ise metotlar arasında uyumun eksik olduğu sonucuna varılmaktadır. Uyumun düşük olması geliştirme sırasında hata yapılma olasılığını artırıp kodun değişebilirliğini azaltmaktadır [20].

### 2.6.3. Çevrimsel Karmaşıklık (İng. Cyclomatic Complexity- CC)

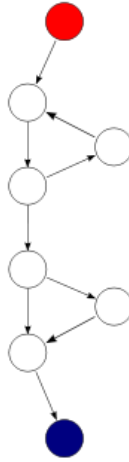
Kodun karmaşıklığını veren metriktir. Ölçüm yapılırken akış diyagramındaki bağımsız bölümler ve karar yapıları göz önünde bulundurulmaktadır. Kod karmaşıklığının düşük olması istenen bir özelliktir. Karmaşık kodun anlaşılabilirliği düşük, bakım yapılabilirliği ve test edilebilirliği zordur [20].

Literatürde bu metriğin sahip olduğu aralığa göre taşıdığı risk seviyeleri çizelge 2.2'deki gibi belirtilmektedir [20].

Çizelge 2.2. Çevrimsel karmaşıklık ve risk

Çevrimsel Karmaşıklık	Risk
1 - 10	Basit, risk az
11 - 20	Daha kompleks, orta derece riskli
21 - 50	Komplek, yüksek dereceli riskli
> 50	Test edilemeyen, çok yüksek riskli

Şekil 2.4'te örnek bir programın kontrol akış grafiği gösterilmektedir. Kırmızı düğüm programın başlangıç noktası olup mavi düğüm bitişi göstermektedir [20].



Şekil 2.4. Kontrol akış grafiği

Şekil 2.4, basit bir programın kontrol akış grafiğini göstermektedir.

K: Grafikteki köşe (İng. edge) sayısı

D: Grafikteki düğüm (İng. node) sayısı

B: Grafikteki bağlantılı bileşenlerin (İng. connected component) sayısı

$$\text{Çevrimsel Karmaşıklık} = K - D + 2B \quad (2)$$

Denklem 2'ye göre şekilde kontrol akış grafiği verilen programın çevrimsel karmaşıklığı  $9-8+2*1=3$  olarak hesaplanır.

#### **2.6.4. Ortalama Döngüsel Bağımlılık (İng. Average Cycling Dependency-ACD)**

John Lakos tarafından önerilen bir metrik olup yazılımın ortalama döngüsel bağımlılığına bakar. Bağımlılıklara hem doğrudan hem de dolaylı bakarak geçişli bağımlılığın ölçümünü sunar. Bu metrik, yazılım bileşenlerinin ortalama bağlaşım değerinin sayısallaştırılması için kullanılabilir [22].

Tez üzerindeki etkisine, bileşenlerde gerçekleştirilen yapılandırmanın sonucunun yazılım ürün hattına yansımaya bu metrik kullanılarak bakılmıştır. ACD metriği, SonarGraph aracı tarafından doğrudan verilen bir metriktir [22].

#### **2.6.5. Ölçüm Araçları**

Tez kapsamında yapılan ölçüm çalışmalarında, Understand ve Sonargraph gibi ölçüm araçları kullanılmıştır.

##### **2.6.5.1. Understand**

Statik kod analiz aracıdır. Metrik sonuçları otomatik olarak çıkartılıp görsel olarak sunulur. Araç, nesneye yönelik metrikleri, hacim metriklerini ve karmaşıklık metrik gruplarını sayısallaştırıp sunar [23].

Çalışma kapsamında Understand aracı ile LCOM, WMC ve CC metrik değerleri elde edilmiştir [23].

##### **2.6.5.2. Sonargraph**

Çalışmada kullanılan bir diğer statik kod analiz aracıdır. Yazılımdaki bağımlılıkları tespit edip görselleştirir. Java ve C# programlama dilleri kullanılarak geliştirilmiş yazılımların analizini yapmaktadır. Metrik ve yazılım yapısını analiz ederek kod kalitesini değerlendirmeyi destekler [22].

Çalışma kapsamında Sonargraph aracı ile giriş çıkış yelpazesi metrikleri ve ACD metriği elde edilmiştir [22].

#### **2.7. Nesneye Yönelik Yazılım Metriklerinin Yorumlanması**

Yöneticiler ve yazılımcıların karar verme, kaynak ayırma ve çizelge yapma ihtiyacı yazılım ölçümü gibi değerli yol göstericileri gündeme getirmektedir [24].

Yazılım ölçümü, yazılım sürecinin ya da yazılım ürününün karakteristiklerinin sayısallaştırılması olarak tanımlanmaktadır. Ölçümdeki temel amaç, planlama aktivitesini desteklemek, tasarımı kontrol etmek, kaliteyi değerlendirmek ve süreci izlemektedir [24].

Yazılım metrikleri karar verdirici ve seçim yaptırıcı bilgi sağlamaktadır. Yazılım metrikleri buldukları aralıklara göre anlamlı kategorilere girmektedir. Tasarım ve kod metrikleri, prosedürel ya da nesne yönelik olarak sınıflandırılmaktadır. Nesneye yönelik metrikle çok biçimlilik, kalıtım, mesaj geçirilmesi gibi nesne yönelim mekanizmalarının etkilerinin gösterilmesinde kullanılmaktadır. Bu tür metrikler, ölçtükleri nesneye yönelik özelliklerine göre sınıflandırılmaktadır [24].

Metrikler sadece kodun yapısına değil aynı zamanda yazılımın yüksek seviye özelliklerine de bağlıdır. Nesneye yönelik metrikler, nesneye yönelik tasarımların kalitesini, sınıf ve tasarım karakteristiklerini değerlendirmede kılavuzluk yapmaktadır [24].

Geçmiş verilerden elde edilen bulgular, çok biçimlilik, kalıtım, bağlaşım gibi nesneye yönelik mekanizmaların; bakım yapılabilirlik, yeniden kullanılabilirlik, test edilebilirlik gibi yazılım karakteristiklerini etkilediğini göstermektedir [24].

Bağlaşım yazılım karakteristiği karmaşıklık, yeniden kullanılabilirlik, anlaşılabilirlik, test edilebilirlik, bakım yapılabilirlik gibi yazılım özelliklerini değerlendirmede kullanılmaktadır [24].

Etkinlik, karmaşıklık, yeniden kullanılabilirlik, anlaşılabilirlik, test edilebilirlik ve bakım yapılabilirlik gibi yazılım özellikleri “kapsülleme” nesneye yönelik karakteristiği ile sayısallaştırılmaktadır. Seçilen tasarım alternatifi, yeniden kullanım kalitesi üzerinde güçlü bir etkiye sahiptir [24].

C&K metriklerinin yorumlanmasında farklılıklar bulunmaktadır. Örneğin yüksek değerli DIT metriği, bakım yapılabilirlik karmaşıklığının yüksek olduğunu gösterdiği gibi yeniden kullanımında yüksek olduğunu gösteren bir indikatördür [24]. Yüksek NOC değeri test için gerekli eforu arttırdığı gibi yeniden kullanımının verimini de arttırmaktadır. Özetle metriklerle ilgili çok sayıda çalışma olsa da bu metriklerin net olarak yorumlanmasını sağlayacak kılavuz bulunmamaktadır [24].

Düşük CBO değeri, sınıfın daha az bağımlı olduğu göstermektedir. Böylece yazılımın yeniden kullanımı ve test edilebilirliği kolaylaşmaktadır. Bağımlılığı az olan sınıfın başka uygulamada yeniden kullanılabilirliği artar. CBO değeri yüksek olan sınıfların bakım yapılabilirlik ve değiştirilebilirlik hassasiyeti de yükselmektedir. Bağımlılığı yüksek olan sınıfın anlaşılabilirliği zorlaşır [24].

LCOM benzerliği ve farklılığı sıfırdan farklı olan metotların sayıları arasındaki farktır. Benzerlik, ortak olarak kullanılan özellikler olarak tanımlanmaktadır. Düşük değerli LCOM metriği sınıfın kohezyonunun yüksek olduğunu göstermektedir. Yüksek LCOM değeri, sınıfların ayrılması gerektiğini işaret etmektedir. Yüksek uyumu olan sınıf iyi bölünmüş sınıf olarak nitelendirilmektedir. LCOM değeri yüksek olan sınıfların karmaşıklığı yüksek olup hataya yatkınlığı daha fazla olarak nitelendirilmektedir [24].

### 3. İLİŞKİLİ ÇALIŞMALAR

#### 3.1. Yeniden Kullanılabilirlik Faktör ve Metrikleri

Bileşen bazında yeniden kullanılabilirlik metrikleri için literatür taraması yapılırken eksiksiz ve tarafsız bir makale havuzu oluşturmak için Peterson ve Kitchenham'ın sunmuş olduğu Sistematik Literatür Tarama (İng. Systematic Literature Review) kılavuzundan faydalanmıştır [25]. Sistematik literatür taraması, spesifik bir araştırma sorusu ile ilgili olarak mümkün olan tüm araştırma sonuçlarının değerlendirilmesi ve yorumlanmasına dayalı bir metodolojidir. Bu yöntemde, araştırmanın titiz, tarafsız ve tekrarlanabilir olarak yapılması hedeflenmektedir. Kılavuza göre ilgili makaleler seçilirken; araştırma sorusu (İng. research question), seçim kriteri (İng. selection criteria), veri çıkarma stratejisi (İng. data extraction strategy) ve çalışmanın kalite değerlendirmesi (İng. study quality assessment) dikkate alınmıştır.

##### 3.1.1. Araştırma Hedefi ve Soruları

Çalışmada, yeniden kullanılabilirliği sayısallaştırmak, yani nicel yöntemlerle değerlendirmek amaçlı yapılan çalışmaları araştırmak için Hedef-Soru-Metrik yaklaşımı uygulanmıştır [26]. Bu yaklaşıma göre araştırma hedefi ve soruları belirlenmiş, türetilen araştırma sorularına cevaben, bileşen tabanlı yazılımlar ve ürün hattı yaklaşımı için yeniden kullanılabilirliği ölçmeye yönelik olarak literatürde sunulmuş metrik setleri elde edilmiştir.

Literatür çalışmasının hedefi, bileşen-tabanlı ve ürün-hattı yazılım geliştirme yaklaşımlarında yeniden kullanılabilirliği ölçmek için yapılan çalışmaları anlamaktır. Bu hedefi adresleyen iki araştırma sorusu (AS) tanımlanmıştır:

- AS-1: Bileşen-tabanlı ve ürün-hattı yazılım geliştirme yaklaşımına dayalı yeniden kullanılabilirliği ölçmek için hangi metrikler önerilmiştir? Bu metrikler hangi yöntemlerle birlikte kullanılmıştır?
- AS-2: Bileşen-tabanlı ve ürün-hattı yazılım geliştirme yaklaşımlarında yeniden kullanılabilirliği ölçmek için yapılan çalışmalarda, yeniden kullanılabilirliği etkileyen faktörler nelerdir?

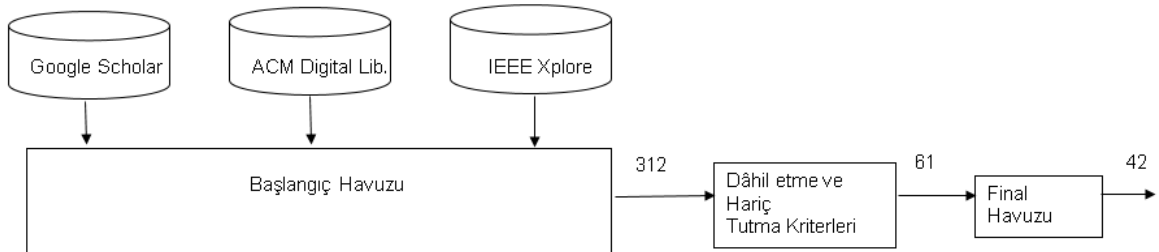
### 3.1.2. Araştırma Stratejisi

Yukarıda tanımı yapılan araştırma soruları için arama yapılmasını sağlayacak anahtar kelimeler (İng. keyword) çıkarıldı. Çıkarılan bu anahtar kelimeler, makalelerin başlıklarında, özetlerinde ve anahtar kelimelerinde arandı. Anahtar kelimeler oluşturulurken mantıksal operatörlerden yararlanıldı ve aşağıdaki sorgu cümlesine dönüştürüldü:

(("software component" OR "product line") AND  
(reuse OR reusable OR reusability) AND metric OR measure))

Makale havuzunun oluşturulması için, verilen sorgu cümlesi; Google Scholar, IEEE Xplore ve ACM Digital Library elektronik kütüphanelerinde arandı.

Araştırma yapılırken Google Scholar diğer akademik veri tabanlarını indekslemesi özelliği ile öncelikli veri tabanı olmuştur. İlgili makalelerin havuza dâhil edilmesi için, makalelerin özet bölümleri, başlıkları, anahtar kelimeleri ve sonuç bölümleri incelenmiştir. Şekil 3.1, nihai makale havuzunun nasıl oluşturulduğunu resmetmektedir. Okların üzerindeki sayısal değerler, her adım sonunda havuzda kalan makale sayısını göstermektedir.



Şekil 3.1. Araştırma stratejisi

### 3.1.3. Seçim Kriteri

Makalelerin seçimi Çizelge 3.1'de tanımlanan dâhil etme ve hariç tutma kriterlerine göre gerçekleştirilmiştir. İlk seçim, sorgu sonucunda veri tabanlarından dönen makalelerin toplanması ile gerçekleştirilmiştir. Ortak makaleler havuzdan elenmiştir. Makaleler, özet ve başlık bilgilerine bakılarak değerlendirilmiştir. Kapsamla ilgili olmadığı görülen makaleler başlangıç havuzundan atılmıştır. Geri kalan makalelerde daha derin inceleme yapılmış, çizelgede belirtilen kriterlere göre çalışma havuzdan uzaklaştırılmış ya da çalışmanın kalması yönünde karar verilmiştir. Örneğin, kapsam ve alan bakımından yeniden kullanılabilirlik ile ilişki



olduğu görülen bir makale, yeniden kullanım seviyesini tanımlayan bir çalışma ise (metrikler üzerinden sayısal değerlendirme içermiyorsa) havuzdan atılmalıdır.

Çizelge 3.1. Dahil etme kriteri ve hariç tutma kriterleri

<b>Dâhil Etme Kriteri (İng. Inclusion Criteria)</b>	<b>Hariç Tutma Kriteri (İng. Exclusion Criteria)</b>
İngilizce yazılmış makaleler	Alanla ve konuyla ilgisi olmayan makaleler
Bilimsel konferanslarda/dergilerde yayınlanmış makaleler	İngilizce yazılmamış makaleler
Yazılım bileşenlerine odaklanmış makaleler	Somut bir metrik/yöntem sunmayan makaleler
Yeniden kullanılabilirlik metriğine odaklanmış makaleler	Yeniden kullanım seviyesini tanımlayan makaleler
Sayısal ölçüm sunan makaleler	Çakışan makaleler
Deneysel çalışma ile doğrulanmış makaleler	Deneysel çalışma ile doğrulanmamış makaleler

#### **3.1.4. Makalelerin Kalite Değerlendirmesi**

Makalelerin kalitesi nerede yayımlandıklarına, hangi araştırma yöntemini kullandıklarına ve deneysel çalışma sunup sunmadıklarına göre değerlendirilmiştir.

Tarama yapılırken makalenin konferans ya da dergide yayınlanmış olmasına ve niceliksel yöntem kullanmış olmasına önem gösterilmiştir.

#### **3.1.5. Veri Çıkarım**

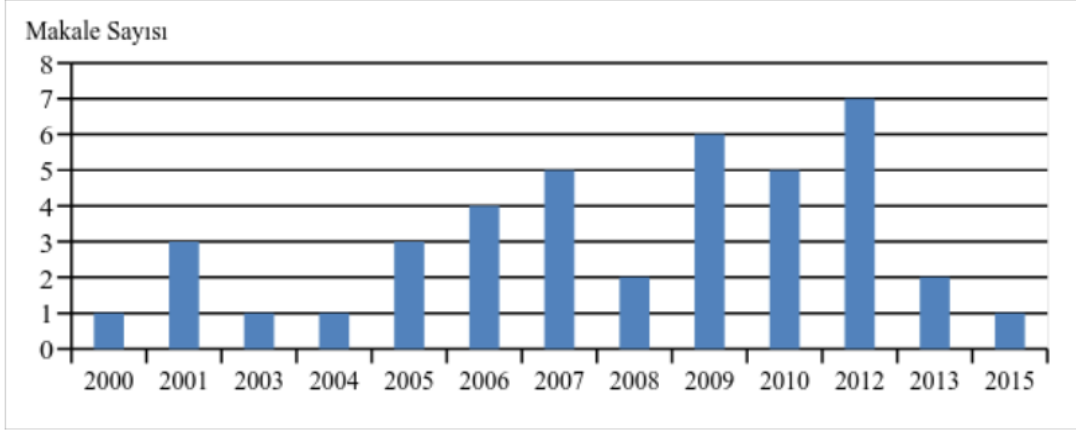
Seçim kriterleri uygulandıktan sonra havuzda kalan nihai makalelerde derin analiz ve inceleme yapılarak veri çıkarma işlemi yapılmıştır. Makaleler okundukça yanlarına önemli görülen noktalar not edilmiştir. Veri çıkarma işlemi Google Drive'da Excel dokümanı üzerinden yönetilmiştir. Nihai havuzda kalan ilgili makalelerden veri çıkarma işlemi yapıldıktan sonra her bir sonuç için ekran gözden geçirme yapılmıştır.

#### **3.1.6. Araştırma Sonuçları**

Bileşen tabanlı ve ürün hattı yazılım geliştirme yaklaşımları ile ilgili olarak yapılmış olan çalışmaların yıllara göre dağılımları Şekil 3.2'de verilmiştir.

Çalışmalar uyguladıkları yöntemler ve odaklandıkları konular açısından çeşitlilik göstermektedir. Yazılımın yeniden kullanılabilirliği birçok faktörden etkilendiği için literatürde farklı yaklaşımlar denenmiştir. Çalışmaların bazıları yazılım içindeki

yeniden kullanılabilirliğe uygun parçaların belirlenmesine yönelik iken bazıları bileşenlerin yeniden kullanılabilirliğini değerlendirmeye çalışmaktadır. Çizelge 3.2’de araştırma sorularına yanıt olarak elde edilen metrik, yöntem ve faktörler verilmiştir.



Şekil 3.2. Makalelerin yıllara göre dağılımı

İlk araştırma sorusu (AS-1) kapsamında, bileşen tabanlı yazılımların yeniden kullanılabilirliğini değerlendirmeye yönelik olarak sunulmuş çalışmalar EK-1’ deki çizelgede sunulmuştur. Çizelgedeki “Yazılım Özellikleri” sütunu, çalışmalarda yeniden kullanılabilirliği etkilediği düşünülen yazılım özelliklerini; “Yazılım Metrikleri” sütunu, yeniden kullanılabilirliği etkileyen yazılım özelliklerini ölçmek için kullanılan metrikleri; “Yeniden Kullanılabilirliği Ölçme Yöntemleri ve Metrikleri” sütunu ise yeniden kullanılabilirliği ölçmek için önerilen yöntem ve metrikleri ifade etmektedir.

Bileşen tabanlı yazılımların yeniden kullanılabilirliğini araştıran çalışmalarda sunulan metot ve/veya metrikler uygulamalarla doğrulanmıştır. Örneğin Sandhu ve Singh çalışmalarında [27], altı sınıf içinden 48 örnek seti çıkaran geliştirilmiş sisteme başvurmuştur. Bulanık arayüz sistemindeki değişkenler, girdi parametrelerin ölçümlerine bağlı olarak, yine girdi parametrelerine atanmıştır. Bu çalışmada yeniden kullanılabilirlik altı farklı değişkene atanmıştır (PERFECT, HIGH, MEDIUM, LOW, VERY LOW ve NIL). Çıktı olarak 200 iterasyon sonunda normalize edilmiş kurallar ile yazılımın yeniden kullanılabilirliğini değerlendiren bir sistem sunmuştur.

Torkamani çalışmasında [28], ürün hatlarının yeniden kullanılabilirliğini değerlendirmeye yönelik olarak özgün metrik takımları önermiştir. Yeniden kullanımdan elde edilen kar ve bir bileşenin yeniden kullanımının bir ürünün elde edilmesine olan etkisi gibi metrikler sunulmaktadır. Yazılım parçalarının ağırlıklı değerleri dikkate alınarak metrik önerilmiştir.

Zhang ve arkadaşları [29], ürün hattı mimarisinin kalitesini belirlemek için yeni metrik takımı önermiştir. Çalışma doğrudan yeniden kullanılabilirliğe odaklanmamıştır. Ürün hattı kalitesini etkileyen faktörler olarak yeniden kullanılabilirlik, değişkenlik, karmaşıklık ve benzerlik gibi özellikler göz önüne alınmıştır. Ürün hattı mimarisi tanımlama dili (İng. Product Line Architecture Description Language) analiz edilerek mimari kalitenin değerlendirilmesi yaklaşımı benimsenmiştir.

Her ve arkadaşlarının yapmış oldukları çalışmada [30], ürün hattındaki değerlerin yeniden kullanılabilirliğini değerlendirmeye yönelik olarak çerçeve sunulmaktadır. Çalışmada, ürün hattının yeniden kullanılabilirliğini etkileyen kalite faktörleri belirlenmiş ve belirlenen bu faktörleri ölçmek için metrikler sunulmuştur. Yazılım özelliklerinin değişkenlik gösterdiği ya da ortaklandığı noktalar göz önüne alınmıştır.

İkinci araştırma sorusu (AS-2) kapsamında, bileşen tabanlı yazılımlarda yeniden kullanılabilirliği en çok etkileyen ilk üç yazılım özelliğinin çalışmalara göre dağılım frekansları, Çizelge 3.2'de sunulmuştur.

Çizelge 3.2. Yeniden kullanılabilirliği etkileyen özelliklerin frekansı ve ilişkili çalışmalar

Yazılım Özelliği	Makale Sayısı	Referans eden makaleler
Bağlaşım	17	[32],[33],[35],[36],[38],[39],[41],[42],[43],[44],[47],[50],[52],[53],[54],[55],[57]
Uyumluluk	11	[38], [39],[41],[42],[44],[52],[53],[54],[55],[56],[62]
Karmaşıklık	7	[33],[35],[36],[42],[43],[50],[63]

Kara Kutu (İng. black box) olarak kabul edilen yazılımların bütün olarak yeniden kullanılabilirliği değerlendirilmektedir. Bileşen tabanlı yazılımların Kara Kutu olarak

ele alındığı yöntemlerde, yazılımın yeniden kullanılabilirliğinin arayüz üzerinden hesaplandığı yaygın olarak görülmektedir.

Beyaz Kutu (İng. white box) yazılımların ise kaynak kodları üzerinden yeniden kullanılabilirliği analiz edilmektedir. Beyaz Kutu ve Kara Kutu olarak değerlendirilen yazılımların tümü için en çok kullanılan kalite özelliğinin Bağlaşım (İng. coupling) olduğu görülmektedir. Yazılımın parçaları arasındaki kullanım ilişkileri, yazılımın yeniden kullanılabilirliği açısından belirleyici bir nitelik olarak ortaya çıkmaktadır. Bileşen tabanlı yazılımların yeniden kullanılabilirliğini ölçmeye yönelik sunulan metrik ve yöntem çalışmalarının büyük kısmı, Beyaz Kutu yaklaşımına dayanmaktadır.

Yapılan araştırma sonucunda, bileşen tabanlı birçok çalışma olduğu gözlenirken ürün hattının yeniden kullanılabilirliğine yönelik olarak sunulmuş çalışmaların az olduğu görülmüştür. Yeniden kullanım ilkesine dayanan yazılım ürün hatlarının yeniden kullanılabilirliğini ölçme çalışmalarında boşluk olup bu alanın çalışmaya ve araştırmaya açık olduğu görülmektedir.

Yazılım ürün hattı yaklaşımı ile geliştirilen yazılımların, yeniden kullanılabilirliğini ölçmeye yönelik sunulan yaklaşımları tarayan herhangi bir literatür tarama çalışmasına rastlanmamıştır. Bu alanda yeniden kullanılabilirliği etkileyen faktörler incelendiğinde bunların, bileşen temelli yazılımlar için yapılan çalışmaları etkileyen faktörlerden ayrıştığı görülmektedir. Geliştirme eforu, geliştirme maliyeti, yenileme gibi faktörler yazılım ürün hatlarında ele alınan faktörlerden birkaçıdır.

Araştırmaya dâhil edilen çalışmaların analizini içeren tablo EK-2'de sunulmaktadır.

### **3.2. Bileşenlerin Yeniden Kullanılabilirliği**

Korra ve arkadaşlarının yaptıkları çalışmada [65], yeniden kullanılabilir yazılım bileşenlerinin tasarlanması ve gerçekleşmesi için literatürde sunulmuş ve kanıtlanmış metot ve stratejilerin neler olduğu aktarılmaktadır. Çalışma, özel bir uygulamanın veri bağımlılığı ile nasıl başa çıkılacağını adreslemektedir. Bileşen Tabanlı Geliştirme, yüksek bileşen yeniden kullanımı ve düşük pazara sürme süresi sağlayan bir geliştirme yaklaşımıdır. Bileşenin oluşturulmasında bileşenin hangi fonksiyonellikleri kapsayacağı önemli bir tasarım kararıdır. Var olan

bileşenlerin yeniden kullanımına dayanır. Gereksinim spesifikasyonları, tasarım dokümanları, kaynak kodlar yeniden kullanılabilir öğeler arasında sayılabilir. Bileşen tabanlı geliştirme, geliştirme zamanı ve eforunu düşürüp kaliteyi yükselten bir yaklaşım olarak tanımlanmaktadır. Değişen sistem gereksinimleri, organizasyonel kültür, yeniden kullanımı destekleyen otomasyon araçlarının kullanımında eksiklik, yatırım gereksinimleri, organizasyonun boyutu, gerçekleştirilen uygulama miktarı, deneyim gereksinimi, yazılım yaşam döngüsünün uzun oluşu ve bileşen indekslemesindeki eksiklik gibi konular, yeniden kullanım konusunda engel ortaya çıkarmaktadır. Bir bileşenin yeniden kullanılabilir olarak görülmesi için o bileşenin kendi içinde tutarlı ve tam olması ve makine ve uygulamadan bağımsız olması, hatalarının ayıklanabilir olması, anlaşılabilir olması, iyi dokümente edilmiş olması, güvenilir olması, test edilebilir olması, bakım yapılabilir olması, taşınabilir olması, standartlara uygun olarak geliştirilmiş olması, veri modelinden bağımsız olması beklenmektedir. Çalışmaya göre yeniden kullanılabilir bileşenleri oluşturmak için; genelleştirme, standardizasyon, bileşenleri otomatikleştirme, kaliteyi sertifikalandırma ve dokümente etme adımları izlenmelidir [65].

Bileşenlerin genelleştirilmesi, bileşenin farklı kullanımlarda ortak ihtiyaca cevap vermesi olarak tanımlanmaktadır. Bunun için parametreler kullanılmalı, daha yüksek derecede soyutlama yapılmalı, bilgi gizleme ve kapsülleme yapılmalı, bileşen tasarlanırken sabit ve değişken parçalar tanımlanmalıdır. Bileşenin standartlaştırılması geliştirilmesinde tek tip yaklaşımı ile sağlanmaktadır. Kodlama stili, birleştirilmiş yapısal programlama teknikleri, fonksiyonların ve değişkenlerin isimlendirilmesi, yüksek bağlaşımlı sınıfların yapılandırılması ve bilgi saklama, standart yazılım geliştirme kullanılan temel yaklaşımlardır. Bileşenlerin otomasyonu, uygulamanın bir kısmının, kodun veya tasarım dokümanının otomatik üretilmesine karşılık gelmektedir. Yeniden kullanımın başarılı olmasındaki bir diğer kriter ise bağımsız kalite sertifikasyon sürecinin olmasıdır. Test, karmaşıklık ölçümü, otomatik söz dizimi (İng. syntax) kontrolü, standartlar ile uyum, dış bağımlılığı olmayan bileşen geliştirme, platformlar arası taşınabilirlik, yeniden kullanım istatistiği ve kod gözden geçirmesi sertifikasyonda izlenecek kontrol listesi (İng. check list) olarak önerilmektedir. Yeniden kullanımda kritik olan diğer konu da dokümantasyondur. Bileşenin adı, bileşenin kategorisi, arayüz

gereksinimleri, bileşenin ne yaptığı, bileşenin özellikleri, yeniden kullanım bilgisi, bileşenin spesifikasyonu, kalitesi, yazarı, bileşen için geliştirilen test, bileşene kimin bakım yaptığı, geliştirme için öneriler, gereksinim spesifikasyonlarına bağlantı, versiyon ve bileşenin diğer bileşenler ile olan ilişkisi dokümantasyonun kontrol listesini oluşturmaktadır [66].

Gill yapmış olduğu çalışmada [67] yazılımın yeniden kullanılabilirliğini, yazılım bileşeninin yeniden kullanım potansiyelini adresleyen nitelik olarak tanımlanmaktadır. Bileşen tabanlı yazılım geliştirme mevcut yazılım geliştirme yaklaşımlarının hem uzantısı hem de alt seti olarak görülmektedir. Bileşen tabanlı geliştirme, bir referans modelidir. Paralel geliştirmeye olanak verir. Farklı teknolojilerde geliştirilmiş bileşenler bağımsız olarak bir araya getirilip entegre edilebilir. Kullanılmadan önce bileşenin davranışı doğrulanabilir. Zaman ve maliyetlerdeki kazanç, endüstrinin yeniden kullanım yönünde neden hareket ettiğini açıklamaktadır. Yazılım bileşeninin yeniden kullanımı, ancak ve ancak yeniden kullanım maliyetinin yeni bileşen geliştirme maliyetinden daha düşük olması durumunda kabul edilebilir [67].

Basili ve Caldiera [68] üç yeniden kullanılabilirlik faktörünü tanımlayan bir model sunmuşlardır. Yeniden kullanım maliyeti, yeniden kullanılabilir bileşenin kullanılabilirliği ve yeniden kullanılabilir bileşen kalitesi kavramları, yeniden kullanımda göz önüne alınması gereken faktörler olarak tanıtılmaktadır. Yeniden kullanılabilirlik kılavuzları, yeniden kullanılabilirliğin aşağıdakileri içermesi gerektiği konusuna dikkat çekmektedir: Kolay anlaşılabilirlik, fonksiyonellik, güvenilirlik, hata işleminin yapılması, bilgi saklama mekanizmasının aktif olması, yüksek uyumluluk, düşük bağlaşım olması, taşınabilirliğin sağlanması ve modülerliğin sağlanmasıdır [68].

Çalışmaya göre bileşen tabanlı yazılımlarda yeniden kullanılabilirlik seviyesini arttırmak için aşağıdaki şu yüksek seviye öneriler sunulmaktadır:

- Yeniden kullanım için potansiyelin ölçülerek yeniden kullanım değerlendirmesinin yapılması,
- Yeniden kullanım için maliyet fayda analizinin gerçekleştirilmesi,
- Bileşenler için standartların benimsenmesi,

- Yeniden kullanım için pilot proje seçilmesi,
- Yeniden kullanılabilirlik metriklerinin değerlendirilmesi.

Yeniden kullanım değerlendirmesi için organizasyonun mevcut yeniden kullanım stratejisi değerlendirilir. Değerlendirme sonucuna göre organizasyonun yeniden kullanım hedefleri belirlenir. Yeniden kullanım hedef ve stratejisine göre aksiyon alınır. Organizasyonda başarılı yeniden kullanım; dikkatli planlama, iş birliği ve iyi yönetim pratiği gerektirmektedir. Yeniden kullanımın değerlendirilmesi organizasyonel ve teknik olarak ayrı ayrı yürütülmelidir. Teknik tarafta yeniden kullanılabilirlikle ilgili bazı önemli konular [68];

- Yeniden kullanılabilir bileşenlerin belirlenmesi,
- Standartların ve kılavuzların tanımlanması,
- Organizasyonel yapının ve sınıflandırmanın tanımlanması olarak belirtilmektedir.

Yeniden kullanım için maliyet fayda analizinin yapılması, organizasyonun yeniden kullanıma yatırım yapıp yapmayacağını kararını vermesinde önemli bilgi sağlamaktadır. Bu analiz tek başına karar verdirici kıstas olmasa bile fikir vermesi açısından kullanılabilirliği yüksek olarak nitelendirilmektedir [68].

Schmidt'in yapmış olduğu çalışmada [69], nesneye yönelik programlamadaki teknikler, kapsülleme ve soyutlama mekanizmaları sağlayarak yeniden kullanılabilirliğe olanak sağlamaktadır. Tasarım örüntüleri, bileşenlerin yapılarının ve aralarındaki ilişkilerin açıklanması yoluyla yeniden kullanılabilir bileşenlerin geliştirilmesine olanak sağlamaktadır. Bu tasarım örüntüleri ile kaynak kod seviyesinden daha üst bir seviyede yapılır. Sonuç olarak tasarım örüntüleri, diğer yeniden kullanım formları uygunsuz olsa bile, mimarinin yeniden kullanılmasını sağlamaktadır. Tasarım örüntüleri, tasarım deneyim pratiklerinin dokümantasyonunda önemli araçlar olarak tanımlanmaktadır. Tasarım örüntüleri, yaygın yazılım problemlerini tanımlar ve tasarım aşamasında başarısı kanıtlanmış mimariler önererek yeniden kullanılabilirliğin artırılmasını sağlar [69].

Sagar ve arkadaşları [70] yeniden kullanılabilir bileşenler geliştirmek için literatürde sunulan çalışmaları sistematik bir şekilde gözden geçirmişlerdir. Çalışma, değişik yeniden kullanılabilirlik kavramlarını ele almaktadır. Konferanslarda ve dergilerde

yayınlanıp önerilmiş yeniden kullanılabilirlik metrikleri sunulmaktadır. Çalışmadaki temel amaç, bileşen tabanlı yazılımların yeniden kullanılabilirlikleri ve yeniden kullanılabilirliğin bağlı olduğu faktörleri ele almaktır. Çalışmanın sonucuna göre yeniden kullanılabilirlik yüksek oranda arayüz karmaşıklığına, dokümantasyon kalitesine ve taşınabilirliğe bağlı çıkmıştır. Ayrıca çalışma sonucunda aşağıdaki sonuçları elde etmiştir [70]:

- DIT azalırse yeniden kullanılabilirlik artar.
- RFC artarsa yeniden kullanılabilirlik düşer.
- NOC artarsa yeniden kullanılabilirlik artar.
- CBO'nun yeniden kullanılabilirlik üzerinde çok etkisi bulunmamaktadır.
- WMC artarsa yeniden kullanılabilirlik düşer.

Kakkar ve arkadaşlarının yapmış oldukları çalışmada [71], yazılım sistemlerinin yeniden kullanılabilirliği ele alınmaktadır. Bir bileşen, belirli bir fonksiyon sağlayan bir uygulamanın kendi başına bağımsız ve yer değiştirilebilir bir parçası olarak dikkate alınabilir. Bileşen, diğer bileşenler ile arayüzler vasıtasıyla bağlantı kuran, bağımsız geliştirilebilen, tek başına bir yeteneğe karşılık gelebilen paketler olarak tanımlanmaktadır. Yazılım geliştiren ancak yeniden kullanım konusunda deneyimsiz olan şirketler için yeniden kullanılabilir bileşenlerin geliştirilmesi ekstra maliyete neden olmaktadır. Sıfırdan geliştirme maliyeti yeniden kullanılabilir bileşenler üretildikçe düşürülmektedir [71].

Younoussi ve Roudies'in yaptıkları çalışmada [72], yeniden kullanılabilirlikle ilgili literatürde sunulan çalışmalar sistematik olarak ele alınmıştır. Çalışma, literatürdeki yeniden kullanılabilirlik konseptlerini sunan sistematik literatür taramasıdır. Çalışmaya, göre yeniden kullanılabilirlik konusunu çalışan araştırmalarda çeşitli yaklaşımlar sunulmaktadır [72].

- Uygulama çerçevesinin kullanımı yaklaşımı,
- Ürün hattı yaklaşımı,
- İlgiliye yönelik (İng. aspect oriented) yazılım geliştirme yaklaşımı,
- Bileşen tabanlı yazılım geliştirme yaklaşımı,
- Konfigüre edilebilir uygulamaların kullanımı yaklaşımı,
- Ticari kullanıma hazır uygulamaların kullanımı yaklaşımı,



- Tasarım örüntüleri yaklaşımı,
- Program üreteçlerinin kullanımı yaklaşımı,
- Kütüphanelerin kullanımı yaklaşımı,
- Servis yönelimli sistemlerin kullanımı yaklaşımı .

Yeniden kullanılabilirliği etkileyen yazılım özellikleri de literatürde de tartışılmaktadır [72]:

- Anlaşılabilirlik: Yazılım bileşeni ne kadar anlaşılırsa yeniden kullanımı o kadar kolaydır. Yazılım ne kadar iyi dokümante edildiye ya da ne kadar yorum satırı kullanıldıysa anlaşılabilirliği o kadar yüksektir.
- Taşınabilirlik: Yazılım bileşeni bir çevreden başka bir yazılım çevresine taşınırken az değişiklik gerekiyorsa taşınabilirliği yüksek olarak tanımlanmaktadır.
- Bakım Yapılabilirlik: Hataların giderilmesi, kalitenin yükselmesi için veya sistemin etkinliğinin artırılması için yazılımın kolay değiştirilme derecesi olarak tanımlanmaktadır.
- Adapte Edilebilirlik: Yazılımın gereksinimleri ne kadar kolay karşıladığının derecesidir.
- Arayüz Karmaşıklığı: Arayüz karmaşıklığı yüksek olan bileşenlerin anlaşılabilirliği düşüktür. Bir bileşenin yeniden kullanılabilirliğinin yüksek olmasının koşullarından biri de arayüz karmaşıklığının mümkün olduğu kadar düşük olmasıdır.
- Esneklik: Yazılımın farklı konfigürasyonlarda kullanımı olarak tanımlanmaktadır. Esnekliği yüksek olan bileşenlerin yeniden kullanılabilirliği yüksek olmaktadır.
- Sabitlik (İng. stability): Yazılımın hatadan yoksun olarak güvenli bir şekilde başka sistemlere adapte edilebileceğini göstermektedir.
- Bağımsızlık (İng. independence): Yazılım bileşenin görevini kendinin yerine getirmesi olarak tanımlanmaktadır. Bağımsızlığı yüksek olan yazılım bileşenin yeniden kullanılabilirliği yüksek olarak derecelendirilmektedir.
- Dokümantasyon: Yazılımın anlaşılabilirliğini arttıran özelliktir. İyi dokümante edilmiş yazılımın yeniden kullanılabilirliği yüksektir.

Babu ve Srivatsa'nın yapmış oldukları çalışmada, yeniden kullanılabilirlik ölçümleri analiz edilmiştir. Çalışmada, yeniden kullanım olasılığı ve var olan yazılımdan ne kadar kodun kullanılabileceğinin ölçümü açıklanmaktadır. Yeniden kullanılabilir yazılım ve bunun ortaya çıkardığı maliyeti tartışılmaktadır [73].

Caldiera ve Basili'nin yapmış oldukları çalışma [74], yeniden kullanılabilir bileşenlerin belirlenmesine yönelik yöntem sunmaktadır. Yeniden kullanımda karşılaşılan problemlerden ilki, yeniden kullanılan objenin doğasından kaynaklı olarak ortaya çıkmaktadır. Kullanılacak nesnenin içeriğinin bilinmesi deneyim gerektirmektedir. Yeni yazılım geliştirmede yeniden kullanımı deneyim mümkün kılmaktadır. İkinci temel problem, yeniden kullanılabilir bileşenlerin setindeki eksiklik olarak tanımlanmaktadır [74].

### **3.3. Yazılım Ürün Hattının Yeniden Kullanılabilirliği**

Torkamani ve arkadaşlarının yapmış oldukları çalışma [75], Yazılım Ürün Hattı yaklaşımı uygulanarak geliştirilmiş yazılımların yeniden kullanılabilirliklerinin değerlendirilmesi yönünde bir çerçeve sunmaktadır. Ürün hattı mühendisliği yazılımı yeniden kullanmak için sunulan etkin bir yaklaşımdır. Yeniden kullanılabilir temel varlıklar (İng. core asset) kullanılarak uygulamalar üretilmektedir. Bu yüzden, temel varlıkların yeniden kullanılabilirliğinin yüksek olması, ürün hattının başarısını doğrudan etkilemektedir. Var olan modeller temel varlıkların yeniden kullanılabilirliğini değerlendirse de temel varlıkların karakteristiklerini adreslemeleri yeterli olmamaktadır. Yeniden kullanılabilirlik, temel varlıkların karşılaması gereken en temel yüksek seviyeli kalite özelliği olarak belirtilmektedir [75].

Çalışma, bu temel varlıkların yeniden kullanılabilirliklerini değerlendirmek üzere kapsamlı bir çerçeve sunmaktadır. Çalışmada önce temel varlıkların karakteristikleri belirlenmiş, sonra temel varlıkların yeniden kullanılabilirliklerini karakterize etmek için kalite özellikleri seti üretilmiştir. Her bir kalite özelliğine karşılık bir metrik tanımlanmış ve finalde, ürün hattını değerlendirmek üzere kılavuz sunulmuştur. ISO 9126 yazılım ürünlerinin kalitesi anlatan ve sınıflandıran uluslararası bir standarttır. ISO 9126'ya göre kalite özellikleri işlevsellik, güvenilirlik, kullanılabilirlik, verimlilik, bakım yapılabilirlik ve taşınabilirliktir [75].

Yazılımın kullanımının kolay olması kullanılabilirlik kalite özelliğine karşılık gelmektedir. Son kullanıcının kolay öğrenmesi ve anlaması, uygulamanın kullanılabilirliğinin yüksek olduğuna işaret etmektedir. İşletilebilirlik ve kullanıcı etkileşimi konuları bu kategori altında incelenmektedir.

ISO 9126 standardı, kalite kavramını geliştirici gözünden daha çok müşteri tarafından ele alan bir standarttır. Ancak, özellikle geliştiricilerin önem gösterdiği yeniden kullanılabilirlik kavramı bu standartta ele alınmamaktadır. Yeniden kullanılabilirlikte ISO 9126 standardı yetersiz gelmektedir [76]. Dolayısıyla çalışma, ISO 9126 standardını üzerine çerçeve geliştirilerek yapılmıştır. Çalışmanın ISO 9126 üzerine yapılandırılmış olmasının nedeni standartlara bağlı kalmaktadır. Geliştirilen çerçevenin hedefi belirlenirken Hedef-Soru-Metrik yaklaşımından faydalanılmıştır. Hedef-Soru-Metrik yaklaşımına göre:

- Çalışmanın amacı: Değerlendirebilmek, yönetebilmek ve iyileştirebilmek için temel varlıkları değerlendirmektir.
- Çalışmanın yapılış bakış açısı: Geliştiricilerin, uygulama mühendislerinin ve ürün hattı yöneticilerinin bakış açısı ile çalışma yapılmıştır.
- Çalışma çevresi: Ürün, değişkenlik noktaları olan yeniden kullanılabilir temel varlıklardan oluşmaktadır. Sunulacak çerçeve temel varlıklar geliştirildikten sonra kullanılacaktır.

Yazılım Mühendisliği Enstitüsü'nün (İng. Software Engineering Institute-SEI) yapmış olduğu çalışmada [77], ürün hattı ölçümü iki grupta toplanmıştır. İlk grupta yönetim rollerini içeren metrikler sunulmaktadır. Bu grupta, toplam geliştirme maliyeti, verimlilik, takvim sapmaları, pazara sürme süresi, ürün miktarı, mimarisel uyumluluk ve hedef odağı gibi yönetsel ölçümler bulunmaktadır. İkinci grup metrikler, temel varlıkları üretmek için gösterilen performans gibi ölçümlerden oluşmaktadır. SEI'nin sunmuş olduğu mimarisel uyumluluk ve piyasa özellik kapsamı gibi ölçümler yeniden kullanılabilirliği ölçse bile bu metriklerin yeniden kullanılabilirliği değerlendirmesinde eksiklikler bulunmaktadır [77].

Benzer şekilde IESE tarafından yapılan çalışmada [78], ürün hattı kalite modeli geliştirme için arka plan bilgisi sunulmaktadır. Bu çalışma herhangi bir kalite modelini desteklememektedir. Ancak bu modellerin dokümantasyonu ve geliştirimi

için sunulan yaygın çerçeveleri desteklemektedir. Bu çerçeve, ürün hattı kalite modelinin genel yapısını tartışmakta ve nasıl özel kalite modeli geliştirilebileceğine ilişkin bilgi vermektedir. Ayrıca sunulan çalışma, meta kalite modeli ile spesifik kalite model kılavuzu arasındaki ilişkiyi tartışmaktadır. Bu çerçeve, ürün hattı literatüründeki ve kalitedeki standartları analiz ederek potansiyel kalite adaylarını listelemektedir. Çalışmada sunulan potansiyel kalite adayları; güvenilirlik, bakım yapılabilirlik, kullanılabilirlik ve taşınabilirliktir. Ve her kalite faktörü bir kalite kriterine karşılık gelmektedir. Bu çerçeve yeniden kullanılabilirlikten daha çok temel varlıkların kalitesine odaklanmaktadır. Sonuç olarak, temel varlıkların yeniden kullanılabilirliğini desteklemek için bu çalışma geliştirilmelidir [78].

Washizaki'nin çalışması [79], kara kutu yaklaşımına dayalı olarak herhangi bir kaynak kod olmaksızın sınırlı bilgi ile yeniden kullanılabilirliği ölçmeyi hedeflemiştir. Çalışma, bileşen kullanıcılarının bakış açısı ile geliştirilmiştir. Yeniden kullanılabilirlik modelinde, yeniden kullanılabilirlik üç farklı faktör ile ilişkilendirilmiştir: Adapte edilebilirlik, anlaşılabilirlik ve taşınabilirlik. Bu faktörler kendi içlerinde dört alt kriter ve beş farklı metriğe bölünmektedir. EMI (İng. Existence of Meta Information), RCO (İng. Rate of Component), RCC (İng. Rate of Component Customizability), SCCr (İng. Self Completeness of Component's Return Value) ve SCCp (İng. Self Completeness of Component's Parameter) metriklerinden oluşmaktadır. Çalışmada ayrıca, bu metriklerin kombinasyonları da kullanılmıştır. Ancak bu çalışmada temel varlıklara, yeniden kullanılabilir bileşenden daha çok bileşen olarak davranılmıştır [79].

Dersten'in yapmış olduğu çalışma [80], ürün hattının yeniden kullanılabilirliğinin gelişmesi yönünde metot sunmaktadır. Ürün hattında, yeniden kullanılabilir bileşenler geliştirmek ve faydalanmak anahtar faktördür. Yeniden kullanım problemlerinin üstesinden gelmek için bileşen çerçeveleri (İng. component framework) ürün hattında kullanılmaktadır [80]. Çalışmada, ürün hattı için geliştirilen endüstriyel bileşen modeli olan Koala tanıtılmaktadır. Yazılımda ideali, birbirinden bağımsız bileşenlerin birleştirilerek bir sistem oluşturmasıdır. Ancak gerçekte, bileşenler arasında bağımlılıklar söz konusudur. Bağımlılık yeniden kullanılabilirliği negatif olarak etkilemektedir [80].

Telekom firması olan Nokia, ürün hattı yaklaşımını kullanarak 25-30 farklı telefon modelini bir yıl içinde geliştirmiştir. Bosch firması 1997 yılında ürün hattı yaklaşımını üç yıl süresince benimseyen iki farklı firma üzerinde durum çalışması yapmıştır. Firma belirli problemleri tanımlamıştır. Bu problemlerden ilki ürün hattı mimarisini oluşturacak vasıflı mühendis eksiği; ikincisi ise aynı temel varlıkların farklı ürünlerde kullanılacak olması, çatışan kalite gereksinimlerinin olması ve ürün hattı yaklaşımı ile geliştirmenin ilk etapta zaman alıcı olması olarak tanımlanmaktadır. Aynı zamanda Bosch, ürün hattı yaklaşımında kullanılacak araçların yoksunluğundan ve hangi ürünün nihai olarak ürün hattından çıkacağı kararının verilme zorluğundan bahsetmiştir [80].

Ürün hattı sadece yazılım kaynak kodunun yeniden kullanımı konusunda fayda sağlamakla kalmaz aynı zamanda gereksinim fazında gereksinim belirleme zamanını kırtarır [80].

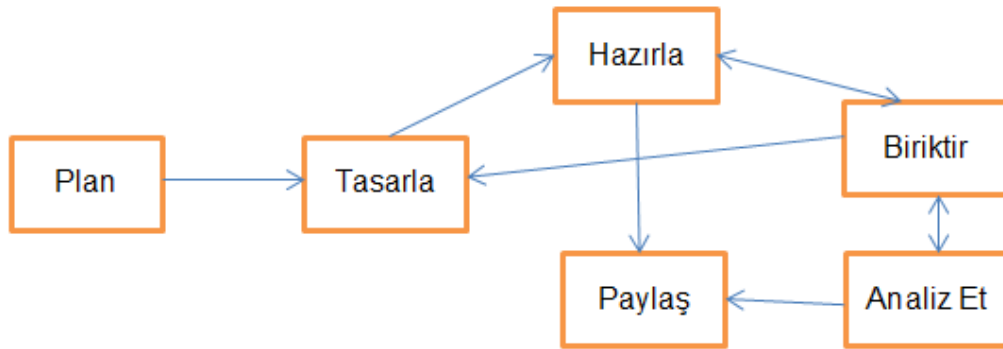
Ommering [81], bileşenin yeniden kullanılabilir olmasını sağlayan iki farklı özelliğinin olduğundan bahsetmektedir. İki bileşenin bağımsız olması, diğeri ise bileşenin başka bir uygulamada kullanıldığı zaman az ya da hiç modifiye edilmiş olmasıdır [81].

#### 4. DURUM ÇALIŞMALARININ TASARIMI

Yin'e göre [82] durum çalışması, güncel bir olgunun gerçek yaşam bağlamında, özellikle bağlam ve olguların sınırlarının kesin olmadığı durumlarda, görgül olarak araştırılması olarak tanımlanmaktadır. Durum çalışmaları "Neden" ve "Nasıl" sorularının yanıtlarını araştırmak için kullanılır. Lineer ve tekrarlayan bir süreçtir. Böylece araştırmacının olaylar üzerindeki etkisi azalır [82].

Durum çalışmaları, gerçekte ortamda neler olduğuna bakma, sistematik bir şekilde verileri toplama, analiz etme ve sonuçları ortaya koyma yoludur. Ortaya çıkan ürün ise olayın niçin o şekilde olduğunun ve gelecek araştırmalar için daha detaylı olarak nelere odaklanma gerektiğinin keskin biçimde anlaşılmasıdır.

Şekil 4.1'de bir durum çalışmasını gerçekleştirme adımları verilmektedir.



Şekil 4.1. Durum çalışması adımları

Plan aşaması, araştırma sorularının tanımlandığı fazdır. Diğer metotlar ile karşılaştırıldığında kullanılacak durum çalışmasına karar verilir. Limitleri ve güçlü yanları ortaya koyulur. Durum seçimi yapılır. Dinleyici kitlesi belirlenir. Araştırma hedefleri belirlenir.

Plan fazından sonra tasarım fazı gelmektedir. Durum çalışması araştırma tasarımı henüz kodlanmış değildir. Tasarım, araştırmacının nasıl analiz edeceğine ve veriyi nasıl biriktirip yorumlayacağına ilişkin kılavuz sağlar. Tasarım fazında, çalışılacak soruların ne olacağı, ilgili verilerin ne olacağı, biriktirilecek verilerin ne olacağı ve sonuçların nasıl analiz edileceği belirlenir. Tasarım fazında, analiz birimi ve üzerinde çalışılacak durumlar tanımlanır. Teori geliştirilir. Durum çalışması tasarım tipi tanımlanır. Durum çalışması tasarım tipi, tekli, çoklu, holistik ve gömülü olarak

çeşitlenmektedir. Son olarak tasarım fazında durum çalışması kalitesini sağlamak için prosedürler tanımlanır. Protokolde, durum çalışmasının konusu belirlenir. Bilgiyi toplamak için prosedür belirlenir. Veri biriktirilirken sorular cevaplanır.

Hazırla fazında, özel bir durum çalışması için antrenman yapılır. Durum çalışması protokolleri geliştirilir. Pilot çalışma bu fazda gerçekleştirilir. Onay alınır. Biriktir fazı, verilerin toplandığı fazdır. Analiz edilmek üzere veriler bir araya getirilir. Analiz fazında biriktirilen verilerin analizi yapılır. Paylaş fazında ise analiz edilen veriler raporlanarak paylaşılır.

Durum çalışmaları işlenen konu bakımından; bireye, gruba, yere, organizasyona ve olaya odaklanmaktadır.

Durum çalışmaları, kanıtlanan teze ve hipoteze bağlı olarak birbirinden ayrılmaktadır. Teze ve hipoteze bağlı olarak araştırmacı, keşifçi, betimleyici, açıklayıcı, iyileştirici, enstrümental olarak alt sınıflara ayrılmaktadır. Tez kapsamında yapılan durum çalışmaları yazılımın yeniden kullanılabilirliğini iyileştirme yönünde yapıldığı için, "iyileştirici" durum çalışma tipine girmektedir.

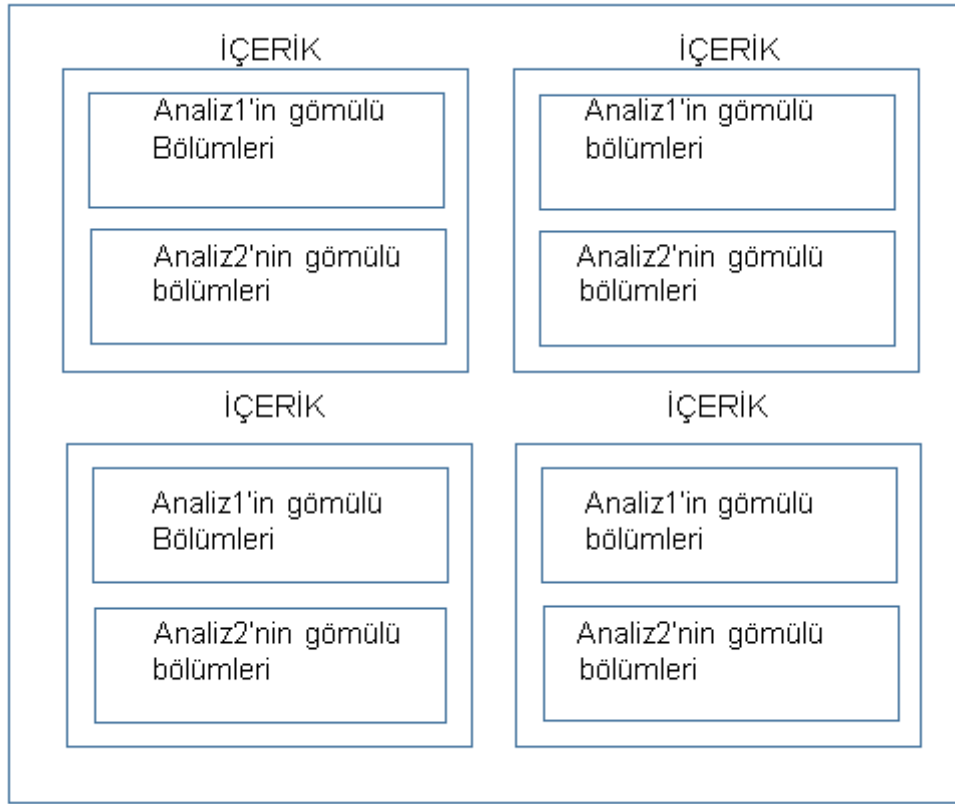
Durum çalışmasının tasarımında aşağıdaki sekiz madde izlenmelidir:

1. Araştırma sorularının geliştirilmesi,
2. Araştırmanın alt problemlerinin geliştirilmesi,
3. Analiz birimlerinin saptanması,
4. Çalışılacak durumun belirlenmesi,
5. Araştırmaya katılacak bireyin seçilmesi,
6. Verinin toplanması ve toplanan verinin alt problemler ile ilişkilendirilmesi,
7. Verinin analiz edilmesi ve yorumlanması,
8. Durum çalışmasının raporlanması.

Yin'e göre durum çalışmalarının tasarım biçimleri aşağıdaki tiplere ayrılmaktadır:

- Holistik tekli durum tasarımı
- Gömülü tekli durum tasarımı
- Holistik çoklu durum tasarımı
- Gömülü çoklu durum tasarımı

Şekil 4.2'de tez kapsamındaki durum çalışmaları için Yin'e göre belirlenen gömülü çoklu durum tasarımının yapısı sunulmaktadır:



Şekil 4.2. Gömülü durum çalışması tasarımı

Çalışmada birden fazla durum söz konusudur. Her durumun içinde alt birimlerin yer alması ve bu alt birimlerin de araştırmaya tabi tutulması gerekmektedir. Tez kapsamında dört farklı radar kullanıcı arayüzü yazılımının, mevcut ve yapılandırma sonrasındaki ölçümleri alındığı için, yapılan durum çalışması, çoklu gömülü durum çalışması olacak şekilde tasarlanmıştır. Projelerin her biri, bir durum çalışmasına karşılık gelmektedir. Projelerin mevcut ölçümlerinin (Analiz 1) ve yapılandırma sonrasındaki ölçümlerinin (Analiz 2) alınması işlemlerinden her biri, analiz birimlerine karşılık gelmektedir. Projelerin her biri ise bir durum çalışmasına karşılık gelmektedir.

Durum çalışmalarında kullanılmak ve faktörleri sayısallaştırmak üzere seçilen metrikler Çizelge 4.1.'de sunulmaktadır.



Çizelge 4.1. Faktör ve metrikler

Faktör	Metrik
Bağlaşım (İng. Coupling)	Giriş/Çıkış Yelpazesi (İng. Fan-in/Fan-out) Ortalama Çevrimsel Bağımlılık (İng. Average Cycling Dependency - ACD)
Uyumluluk (İng. Cohesion)	Metotlardaki Uyum Eksikliği (İng. Lack of Cohesion in Methods – LCOM ) Sınıfın Ağırlıklı Metot Sayısı (İng. Weighted Methods Per Class - WMC)
Karmaşıklık (İng. Complexity)	Çevrimsel Karmaşıklık (İng. Cyclomatic Complexity)

Metotlardaki uyum eksikliği, sınıfların ağırlıklı metot sayısı ve çevrimsel karmaşıklık metrikleri, sınıf seviyesinde hesaplanan metriklerdir. Bu metriklerin OSGi bileşeni seviyesine taşınması için sınıf metrikleri önce normalize edilmiş, sonrasında ise bileşen içinde olup normalize edilmiş tüm sınıf ölçüm değerlerinin ortalaması alınmıştır. Sınıf bazında ölçüm metriğinin normalizasyonu Denklem 3'te verilmektedir. Sınıf seviyesindeki normalize edilmiş metriğin bileşen seviyesine taşınması için yapılan ortalama hesabı ise Denklem 4'te verilmektedir.

$$x_{nor} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (3)$$

$x_{min}$  = bileşendeki min değerli metrik

$x_{max}$  = bileşendeki max değerli metrik

$$x_{ort} = \frac{\sum_1^m x_{nor}}{m} \quad (4) \quad x_{ort} = \text{bileşendeki ortalama metrik değeri}$$

$m$  = bileşendeki sınıf sayısı

#### 4.1. Yeniden Yapılandırma için Özellik Ağacı Kullanılarak Yetenek Seçimi

Tez kapsamında gerçekleştirilen durum çalışmalarında sekiz adet gömülü durum çalışması bulunmaktadır. Gömülü durum çalışmaları, projelerin mevcut bağlaşım, uyumluluk ve karmaşıklık özellikleri ile yeniden yapılandırıldıktan sonra bağlaşım, uyumluluk ve karmaşıklık özelliklerinin belirlenen metriklerle sayısallaştırılması adımlarını kapsamaktadır.

Yazılımların yeniden yapılandırılması için Özellik Ağacı modelinden faydalanılmıştır. Dört farklı yazılım projesinin her birinin Yazılım Gereksinim Özelliği dokümanları kullanılarak Özellik Ağacı oluşturulmuştur. Oluşturulan Özellik

Ağacı EK-3 bölümünde sunulmaktadır. Özellik Ağacı 46 ana yetenek 188 alt yetenekten oluşmaktadır. Yetenek setleri çıkarılırken uzman görüşüne ve akran gözden geçirmeye başvurulmuştur.

Dört projenin yetenek setleri çıkarılarak oluşturulan Özellik Ağacı'nda yetenekler ve projelerin eşlemesi yapılmıştır. Eşleme işlemi tamamlandıktan sonra akran gözden geçirme yapılmıştır.

Yeteneklerin ve projelerin eşleştiği Özellik Ağacı yorumlanmıştır. Bu noktada alan ve uygulama mühendislikleri kavramları devreye girmektedir. Özellik Ağacı'ndaki proje ve yetenek eşlemelerinin her biri yorumlanarak yeniden yapılandırılacak yeteneklerin seçimi yapılmıştır. Yetenek seçimi yapılırken, projeye özel uygulama mühendisliği bileşenine ait olması gereken yeteneğin, alan mühendisliği bileşeni olan yapıtaşında bulunması, o yeteneğin yapıtaşından projeye özel bileşene taşınması gerektiğine işaret etmiş ve karar kriteri olarak uygulanmıştır. Böylece, projeye özgü olması gereken bir yetenek, kullanılmadığı projelerden ayrılarak Ürün Hattı'ndaki alan mühendisliği bileşeninden ayrılmıştır.

Çizelge 4.3'te Özellik Ağacı'ndan küçük bir kesit sunulmaktadır. Radar durum yeteneklerinin Proje 1, Proje 2, Proje 3 ve Proje 4 tarafından kullanılıyor olması, bu yeteneğin alan mühendisliği kapsamında geliştirilen "Yapıtaşı"nda bulunması gerektiği anlamına gelir. Mevcut ürün hattı ailelerinde radar durum yetenekleri, yapıtaşı bileşeninde sunulmaktadır.

Mermi Raporu Yeteneği ve Metro Raporu Yeteneği sadece Proje 1'de kullanılmasına rağmen, ürün hattında alan mühendisliğinin çıktısı olan Yapıtaşı bileşeninde bulunmaktadır. Bu iki yeteneğin, Proje 2, Proje 3 ve Proje 4'te kullanılmıyor olması ve sadece Proje 1'de kullanılıyor olması, bu yeteneğin ürün hattının uygulama mühendisliği çıktısı olan projeye özel bileşende bulunmasını gerektirmektedir. Ne var ki, mermi ve metro raporu yetenekleri mevcut ürün hattında, tüm projelerin kullanımına açılan Yapıtaşı bileşeninde bulunmaktadır. Dolayısı ile bu yeteneklerin Proje 2, Proje 3 ve Proje 4'ün kullanımından çıkarılarak sadece Proje 1'in kullanımına sunulması uygun görülmektedir. Sonuç olarak, metro ve mermi raporu yeteneklerinin dört projede ortak kullanılan Yapıtaşı

bileşeninden çıkarılacak Proje 1'in projeye özel olarak kullanacağı bileşen haline getirilmesine karar verilmiştir.

Çizelge 4.2. Proje yetenek eşleşmesi

Yetenekler	Açıklama	Proje1	Proje2	Proje3	Proje4
<b>RADAR DURUM YETENEKLERİ</b>					
Bağlantı Durum Sunum Yeteneği	Bağlantılar listesi, henüz başlamamış ve devam eden bağlantılar	X	X	X	X
Sıcaklık Sunum Yeteneği	Cihaz birim sıcaklıklarının derece olarak sunumu	X	X	X	X
<b>MERMİ RAPORU YETENEĞİ</b>					
Mermi Raporunun Skoplarda Sunulması		X			
Mermi Raporunun Haritada Sunulması		X			
<b>METRO RAPORU YETENEĞİ</b>					
Metro Raporu Sunum		X			
Metro Raporu'nun KKS'den Alınması		X			

#### 4.2. Araştırma Soruları

Durum çalışmalarının her birinin (1-4), bir projeye (Proje 1-4) karşılık geldiğini belirtmiştik. Bu bölümde her bir durum çalışması için araştırma soruları açıklanmaktadır. Tekrarı önlemek amacıyla, her durum çalışması ve ilişkili proje, X ile kodlanarak anlatılmıştır.

Durum çalışması X içinde iki gömülü durum incelenmiştir. Gömülü çalışmalardan ilki (Analiz-1), Proje X'in mevcut bağlaşım, karmaşıklık ve uyumluluk yazılım özelliklerinin giriş / çıkış yelpazesi, ortalama çevrimsel bağımlılık, metotlardaki uyum eksikliği, sınıfın ağırlıklı metot sayısı ve çevrimsel karmaşıklık metrikleri ile sayısallaştırılması esasına dayanmaktadır. İlk durum çalışmasının ikinci gömülü kısmı (Analiz-2) ise kod yeniden yapılandırıldıktan sonra Proje X'in bağlaşım, karmaşıklık ve uyumluluk yazılım özelliklerinin giriş / çıkış yelpazesi, ortalama

çevrimsel bağımlılık, metotlardaki uyum eksikliği, sınıfın ağırlıklı metot sayısı ve çevrimsel karmaşıklık metrikleri ile sayısallaştırılması içermektedir.

Durum çalışması X'in Analiz-1 kısmı, Proje X'in mevcut durumunun bağlaşım, uyumluluk ve karmaşıklık özelliklerinin ölçülmesini kapsamaktadır. Durum çalışması yapılırken ilk adımında Analiz-1'in gömülü kısımları için araştırma soruları geliştirilmiştir.

- AS1.1: Metot uygulanmadan önce Yazılım-X'in bileşenlerinin LCOM değeri nedir?
- AS1.2: Metot uygulanmadan önce Yazılım-X'in bileşenlerinin WMC değeri nedir?
- AS1.3: Metot uygulanmadan önce Yazılım-X'in bileşenlerinin CC değeri nedir?
- AS1.4: Metot uygulanmadan önce Yazılım-X'in ACD değeri nedir?

Durum çalışması X'in Analiz-2 kısmı, Proje X'in yapılandırma sonrasındaki durumunun bağlaşım, uyumluluk ve karmaşıklık özelliklerinin ölçülmesini kapsamaktadır. Durum çalışması yapılırken ilk adımda Analiz-2'nin gömülü kısımları için araştırma soruları geliştirilmiştir.

- AS2.1: Metot uygulandıktan sonra Yazılım-X'in bileşenlerinin LCOM değeri nedir?
- AS2.2: Metot uygulandıktan sonra Yazılım-X'in bileşenlerinin WMC değeri nedir?
- AS2.3: Metot uygulandıktan sonra Yazılım-X'in bileşenlerinin CC değeri nedir?
- AS2.4: Metot uygulanmadan sonra Yazılım-X'in ACD değeri nedir?

Durum Çalışması X bağlamında Analiz-1 ve Analiz 2 için Understand ve Sonargraph araçları kullanılarak ilgili metrikler toplanmıştır. Ardından durum Çalışması X için Analiz-2'nin sonuçları, Analiz-1'in sonuçları ile karşılaştırılmıştır.

## 5. DURUM ÇALIŞMALARININ BULGULARI

### 5.1. Durum Çalışması-1'in Bulguları

Durum Çalışması-1 için Analiz-1'in sonuçları, Çizelge 5.1'deki yapılandırma öncesi metriklerin sonuçlarına karşılık gelmektedir; Analiz-2'nin sonuçları ise Çizelge 5.1'de yapılandırma sonrası metriklerin sonuçlarına karşılık gelmektedir.

Çizelge 5.1'deki yapılandırma öncesi LCOM metrik değerleri AS1.1'e, yapılandırma öncesi WMC metrik değerleri AS1.2'ye, yapılandırma öncesi CC metrik değerleri ise AS1.3'e karşılık gelmektedir. Çizelge 5.1'deki yapılandırma sonrası LCOM metrik değerleri AS2.1'e, yapılandırma sonrası WMC metrik değerleri AS2.2'ye, yapılandırma sonrası CC metrik değerleri ise AS2.3'e karşılık gelmektedir. ,

Proje 1'de yapılandırma sonucunda LCOM, WMC ve CC metrik değerleri değişen bileşenlerin KontrolModel ve KontrolSunum olduğu görülmektedir. Yapılandırma sonrası bu değerlerin Proje 1 için azaldığı görülmektedir. Yapılandırma sonrasında Proje 1'e, MetroRaporu ve MermiRaporu bileşenleri dahil edilmiştir.

Yapılandırmanın sistemin bütününe olan etkisini görmek için Sonargraph aracının otomatik olarak hesapladığı John Lakos metriği olan Ortalama Çevrimsel Bağımlılık metriğine (ACD) bakılmıştır. Metriğin sonucuna göre Proje 1'in yapılandırma öncesi Ortalama Çevrimsel Bağımlılık metriği değerinin 0,33 olduğu ve yapılandırılan sistemin Ortalama Çevrimsel Bağımlılık metriği değerinin 0,28 olduğu görülmüştür. Ortalama Çevrimsel Bağımlılık metriğinin yapılandırma öncesi ve sonrasındaki değerlerine bakıldığında Proje 1'in genel bağımlılığının azaldığı gözlemlenmiştir.

Çizelge 5.1. Durum çalışması-1 için yapılandırma öncesi ve sonrasında metrikler

YAPILANDIRMA ÖNCESİ METRİKLER			YAPILANDIRMA SONRASI METRİKLER			
	LCOM	WMC	Cyclomatic Modified Complexity	LCOM	WMC	Cyclomatic Modified Complexity
AlarmVerici	0,539	0,31677	0,248062	0,539	0,31677	0,248062
AnaPencere	0,306607	0,160455	0,495192	0,306607	0,160455	0,495192
ArayuzDestek	0,21	0,11625	0,449275	0,21	0,11625	0,449275
CITSunum	0,402	0,22266	0,54142	0,402	0,22266	0,54142
CRAM_KKM	0,542227	0,2861	0,290323	0,542227	0,2861	0,290323
CRAM_KKM_Sim	0,446688	0,329861	0,186813	0,446688	0,329861	0,186813
Destek	0,404581	0,124664	0,639653	0,404581	0,124664	0,639653
Destek2	0,3546	0,33818	0,219048	0,3546	0,33818	0,219048
DurumModel	0,425505	0,271347	0,362135	0,425505	0,271347	0,362135
ErişimKontrolcu	0,32814	0,133291	0,893709	0,32814	0,133291	0,893709
ErkenUyariAlici	0,489444	0,231884	0,282828	0,489444	0,231884	0,282828
ErkenUyariGonderici	0,393	0,242857	0,23913	0,393	0,242857	0,23913
HaritaSunum	0,425867	0,0706552	0,776081	0,425867	0,0706552	0,776081
KKSArayuz	0,445588	0,177602	0,355789	0,445588	0,177602	0,355789
<b>KontrolModel</b>	<b>0,337176</b>	<b>0,108326</b>	<b>0,37</b>	<b>0,33518</b>	<b>0,102625</b>	<b>0,20</b>
<b>KontrolSunum</b>	<b>0,317333</b>	<b>0,092956</b>	<b>0,378134</b>	<b>0,30214</b>	<b>0,051918</b>	<b>0,021521</b>
OlayVeriKutuphanesi	0,794667	0,07508	0,26004	0,794667	0,07508	0,26004
OlayYoneticisi	0,4525	0,347656	0,082109	0,4525	0,347656	0,082109
RadarKabuk	0,41754	0,29	0,23125	0,41754	0,29	0,23125
RaporModel	0,3906	0,2282113	0,460762	0,3906	0,2282113	0,460762
RaporSunum	0,3489	0,16756	0,658879	0,3489	0,16756	0,658879
RSYArayuz	0,4835	0,038615	0,420813	0,4835	0,038615	0,420813
RSYSimulator	0,39024	0,2283	0,45921	0,39024	0,2283	0,45921
Semboloji	0,37095	0,13445	0,612903	0,37095	0,13445	0,612903
SkopSunum	0,316973	0,08786	0,486533	0,316973	0,08786	0,486533
Yardım	0,45054	0,2631	0,303448	0,45054	0,2631	0,303448
MetroRaporu	-	-	-	0,43975	0,28458	0,02512
MermiRaporu	-	-	-	0,484564	0,32656	0,1

## 5.2. Durum Çalışması-2'nin Bulguları

Durum Çalışması-2 için Analiz-1'in sonuçları Çizelge 5.2'deki yapılandırma öncesi metriklerin sonuçlarına, Analiz-2'nin sonuçları ise Çizelge 5.2'deki yapılandırma sonrası metriklerin sonuçlarına karşılık gelmektedir.

Çizelge 5.2'deki yapılandırma öncesi LCOM metrik değerleri AS1.1'e, yapılandırma öncesi WMC metrik değerleri AS1.2'ye, yapılandırma öncesi CC

metrik değerleri AS1.3'e karşılık gelmektedir. Çizelge 5.2'deki yapılandırma sonrası LCOM metrik değerleri AS2.1'e, yapılandırma sonrası WMC metrik değerleri AS2.2'ye, yapılandırma sonrası CC metrik değerleri ise AS2.3'e karşılık gelmektedir.

Çizelge 5.2. Durum çalışması-2 için yapılandırma öncesi ve sonrasında metrikler

YAPILANDIRMA ÖNCESİ METRİKLER			YAPILANDIRMA SONRASI METRİKLER			
	LCOM	WMC	Cyclomatic Modified Complexity	LCOM	WMC	Cyclomatic Modified Complexity
AnaPencere	0,306607	0,160455	0,495192	0,306607	0,160455	0,495192
ArayuzDestek	0,21	0,11625	0,449275	0,21	0,11625	0,449275
CITSunum	0,402	0,22266	0,54142	0,402	0,22266	0,54142
Destek	0,404581	0,124664	0,639653	0,404581	0,124664	0,639653
Destek2	0,3546	0,33818	0,,219048	0,3546	0,33818	0,219048
DurumModel	0,425505	0,271347	0,362135	0,425505	0,271347	0,362135
ErişimKontrolcu	0,32814	0,133291	0,893709	0,32814	0,133291	0,893709
HaritaSunum	0,425867	0,0706552	0,776081	0,425867	0,0706552	0,776081
HariciArayuz	0,385241	0,132723	0,562215	0,385241	0,132723	0,562215
HHF	0,35094	0,228668	0,952285	0,35094	0,228668	0,952285
HHFSim	0,304869	0,098127	0,251215	0,304869	0,098127	0,251215
IFF	0,526124	0,333333	0,6425612	0,526124	0,333333	0,6425612
<b>KontrolModel</b>	<b>0,337176</b>	<b>0,108326</b>	<b>0,37</b>	<b>0,33518</b>	<b>0,102625</b>	<b>0,20</b>
<b>KontrolSunum</b>	<b>0,2508434</b>	<b>0,171783</b>	<b>0,255</b>	<b>0,182515</b>	<b>0,05618</b>	<b>0,2325</b>
OlayVeriKutuphanesi	0,794667	0,07508	0,26004	0,794667	0,07508	0,26004
OlayYoneticisi	0,4525	0,347656	0,082109	0,4525	0,347656	0,082109
RaporModel	0,3906	0,2282113	0,460762	0,3906	0,2282113	0,460762
RaporSunum	0,4491	0,11469	0,851363	0,30212	0,09412	0,851363
RSYArayuz	0,49858	0,024674	0,269852	0,49858	0,024674	0,269852
RSYSimulator	0,29652	0,08964	0,36	0,29652	0,08964	0,36
Semboloji	0,3972634	0,1151515	0,02	0,3972634	0,1151515	0,02
SkopSunum	0,316973	0,08786	0,486533	0,316973	0,08786	0,486533
Yardım	0,45054	0,2631	0,303448	0,45054	0,2631	0,303448
MetroRaporu	-	-	-	-	-	-
MermiRaporu	-	-	-	-	-	-

Yapılandırmanın sistemin bütününe olan etkisini görmek için Sonargraph aracının otomatik olarak hesapladığı John Lakos metriği olan Ortalama Çevrimsel Bağımlılık metriğine bakılmıştır. Metriğin sonucuna göre Proje 2'nin yapılandırma öncesi Ortalama Çevrimsel Bağımlılık metriği değerinin 0,31 olduğu, yapılandırılan sistemin Ortalama Çevrimsel Bağımlılık metriği değerinin ise 0,28 olduğu

görülmüştür. Ortalama Çevrimsel Bağımlılık metriğinin yapılandırma öncesi ve sonrasındaki değerlerine bakıldığında Proje-2'nin genel bağımlılığının azaldığı gözlemlenmiştir.

### 5.3. Durum Çalışması-3'ün Bulguları

Durum Çalışması-3 için Analiz-1'in sonuçları Çizelge 5.3'deki yapılandırma öncesi metriklerin sonuçlarına, Analiz-2'nin sonuçları ise Çizelge 5.3'deki yapılandırma sonrası metriklerin sonuçlarına karşılık gelmektedir.

Çizelge 5.3. Durum Çalışması-3 için yapılandırma öncesi ve sonrasında metrikler

YAPILANDIRMA ÖNCESİ METRİKLER				YAPILANDIRMA SONRASI METRİKLER		
	LCOM	WMC	Cyclomatic Modified Complexity	LCOM	WMC	Cyclomatic Modified Complexity
AnaPencere	0,306607	0,160455	0,495192	0,306607	0,160455	0,495192
ArayuzDestek	0,21	0,11625	0,449275	0,21	0,11625	0,449275
CITSunum	0,402	0,22266	0,54142	0,402	0,22266	0,54142
Destek	0,404581	0,124664	0,639653	0,404581	0,124664	0,639653
Destek2	0,3546	0,33818	0,219048	0,3546	0,33818	0,219048
DurumModel	0,425505	0,271347	0,362135	0,425505	0,271347	0,362135
ErişimKontrolcu	0,32814	0,133291	0,893709	0,32814	0,133291	0,893709
MARArayuz	0,5063	0,13446	0,0642	0,5063	0,13446	0,0642
<b>KontrolModel</b>	<b>0,337176</b>	<b>0,108326</b>	<b>0,37</b>	<b>0,33518</b>	<b>0,102625</b>	<b>0,20</b>
<b>KontrolSunum</b>	<b>0,289354</b>	<b>0,4623656</b>	<b>0,5623</b>	<b>0,2893548</b>	<b>0,4623656</b>	<b>0,4525</b>
OlayVeriKutuphanesi	0,794667	0,07508	0,26004	0,794667	0,07508	0,26004
OlayYoneticisi	0,4525	0,347656	0,082109	0,4525	0,347656	0,082109
RaporModel	0,3906	0,2282113	0,460762	0,3906	0,2282113	0,460762
RaporSunum	0,536969	0,12189	0,15621	0,45615621	0,023161	0,023
RSYArayuz	0,5488216	0,11205	0,89562	0,5488216	0,11205	0,89562
RSYSimulator	0,3913043	0,21267	0,63	0,3913043	0,21267	0,63
Semboloji	0,366213	0,12147	0,4511	0,366213	0,12147	0,4511
SkopSunum	0,316973	0,08786	0,486533	0,316973	0,08786	0,486533
Yardım	0,45054	0,2631	0,303448	0,45054	0,2631	0,303448
MetroRaporu	-	-	-	-	-	-
MermiRaporu	-	-	-	-	-	-

Yapılandırmanın sistemin bütününe olan etkisini görmek için Sonargraph aracının otomatik olarak hesapladığı John Lakos metriği olan Ortalama Çevrimsel Bağımlılık metriğine bakılmıştır. Metriğin sonucuna göre Proje 3'ün yapılandırma öncesi Ortalama Çevrimsel Bağımlılık metriği değerinin 0,32 olduğu,



yapılandırılan sistemin Ortalama Çevrimsel Bağımlılık metriği değerinin ise 0,29 olduğu görülmüştür. Ortalama Çevrimsel Bağımlılık metriğinin yapılandırma öncesi ve sonrasındaki değerine bakıldığında Proje 3'ün genel bağımlılığının azaldığı gözlemlenmiştir.

#### 5.4. Durum Çalışması-4 Bulguları

Durum Çalışması-4 için Analiz-1'in sonuçları Çizelge 5.4'deki yapılandırma öncesi metriklerin sonuçlarına, Analiz-2'nin sonuçları ise Çizelge 5.4'deki yapılandırma sonrası metriklerin sonuçlarına karşılık gelmektedir.

Çizelge 5.4. Durum Çalışması-4 için yapılandırma öncesi ve sonrasında metrikler

YAPILANDIRMA ÖNCESİ METRİKLER				YAPILANDIRMA SONRASI METRİKLER		
	LCOM	WMC	Cyclomatic Modified Complexity	LCOM	WMC	Cyclomatic Modified Complexity
AnaPencere	0,306607	0,160455	0,495192	0,306607	0,160455	0,495192
ArayuzDestek	0,21	0,11625	0,449275	0,21	0,11625	0,449275
CITSunum	0,402	0,22266	0,54142	0,402	0,22266	0,54142
Destek	0,404581	0,124664	0,639653	0,404581	0,124664	0,639653
Destek2	0,3546	0,33818	0,219048	0,3546	0,33818	0,219048
DurumModel	0,425505	0,271347	0,362135	0,425505	0,271347	0,362135
ErişimKontrolcu	0,32814	0,133291	0,893709	0,32814	0,133291	0,893709
<b>KontrolModel</b>	<b>0,337176</b>	<b>0,108326</b>	<b>0,37</b>	<b>0,33518</b>	<b>0,102625</b>	<b>0,20</b>
<b>KontrolSunum</b>	<b>0,36521</b>	<b>0,65112</b>	<b>0,5623</b>	<b>0,2115</b>	<b>0,4</b>	<b>0,2</b>
OlayVeriKutuphanesi	0,794667	0,07508	0,26004	0,794667	0,07508	0,26004
OlayYonetici	0,4525	0,347656	0,082109	0,4525	0,347656	0,082109
RaporModel	0,3906	0,2282113	0,460762	0,3906	0,2282113	0,460762
RaporSunum	0,6212	0,25103	0,15621	0,45615621	0,223161	0,023
RSYArayuz	0,5488216	0,11205	0,89562	0,5488216	0,11205	0,89562
RSYSimulator	0,3913043	0,21267	0,63	0,3913043	0,21267	0,63
Semboloji	0,366213	0,12147	0,4511	0,366213	0,12147	0,4511
SkopSunum	0,316973	0,08786	0,486533	0,316973	0,08786	0,486533
Yardım	0,45054	0,2631	0,303448	0,45054	0,2631	0,303448
MetroRaporu	-	-	-	-	-	-
MermiRaporu	-	-	-	-	-	-

Yapılandırmanın sistemin bütününe olan etkisini görmek için Sonargraph aracının otomatik olarak hesapladığı John Lakos metriği olan Ortalama Çevrimsel Bağımlılık metriğine bakılmıştır. Metriğin sonucuna göre Proje 4'ün yapılandırma öncesi Ortalama Çevrimsel Bağımlılık metriği değerinin 0,34 olduğu görülmüştür.

Yapılandırılan sistemin Ortalama Çevrimsel Bağımlılık metriği değerinin ise 0,31 olduğu görülmüştür. Ortalama Çevrimsel Bağımlılık metriğinin yapılandırma öncesi ve sonrasındaki değerine bakılarak yapılandırmanın Proje 4'ün genel bağımlılığının azaldığı gözlemlenmiştir.

Tüm projelerin genel bağımlılıklarının yapılandırma öncesi ve sonrasındaki değerleri, Çizelge 5.5'de birlikte verilmiştir. AS1.4'ün yanıtı çizelgedeki yapılandırma öncesi ACD metriği değerlerine, AS2.4'ün yanıtı ise çizelgedeki yapılandırma sonrası ACD metriği değerlerine karşılık gelmektedir. Sonuçlar karşılaştırıldığında Proje 1, Proje 2, Proje 3 ve Proje 4 için ortalama bağımlılık değerlerinin düştüğü görülmektedir. İzleyen bölümde bu metrik değerlerindeki düşüşün anlamlılığına bakılmıştır.

Çizelge 5.5. Yapılandırma öncesi ve sonrası Ortalama Çevrimsel Bağımlılık Metriği

	Yapılandırma Öncesi ACD Metrik	Yapılandırma Sonrası ACD Metrik
Proje 1	0,33	0,28
Proje 2	0,31	0,28
Proje 3	0,32	0,29
Proje 4	0,34	0,31

### 5.5. Ortalama Çevrimsel Bağımlılığa Yönelik Bulguların Anlamlılık Analizi

P değeri istatistiksel anlamlılığın varlığının ve varsa mevcut farklılığa dair kanıtının düzeyinin belirlenmesi amacı ile kullanılan bir değerdir. Anlamlılık ise elde edilen farkın istatistiksel olarak anlamlı olduğuna karşılık gelen bir kavramdır.

P değeri bir karşılaştırmada “istatistiksel olarak anlamlı fark vardır” diyeceğimiz zaman yapacağımız olası hata miktarını gösterir. Ünlü bir istatistikçi olan Fisher tarafından bu hatanın maksimum kabul edilebilir düzeyi 0,05 olarak önerilmiş ve kabul görmüştür.

İstatistiksel anlamlılığa bakılarak gruplar arasında anlamlı bir fark olup olmadığı anlaşılabilir. Tez kapsamında, yapılandırma öncesi ve yapılandırma sonrası metrik sonuçları arasındaki farkın önemli ölçüde farklı olup olmadığını anlamak için p değeri kullanılmıştır. İstatistiksel anlamlılık, sayıların güvenilir derecede farklı olduğunu göstererek veri analizine olanak sağlar.

İstatistiksel anlamlılık, %95 güven seviyesi kullanılarak hesaplanır [87]. Grupların arasındaki istatistiksel anlamlılığı hesaplamak literatürde Çizelge 6.1'de verilen formüller kullanılmaktadır [87].

Çizelge 5.6. Anlamlılık hesaplama algoritması [87]

İstatistik	Açıklama	Formül
a1	Bir soruya belirli bir şekilde yanıt veren ilk grubun oranı, bu grubun örnek boyutu ile çarpılır.	$a_1 = p_1 * n_1$
b1	Bir soruya belirli bir şekilde yanıt veren ikinci grubun oranı, bu grubun örnek boyutu ile çarpılır.	$b_1 = p_2 * n_2$
Toplanmış örnek oranı (p)	Her iki grubun iki oranının kombinasyonu.	$p = \frac{a_1 + b_1}{n_1 + n_2}$
Standart Hata (SE)	Oranın gerçek orandan ne kadar uzak olduğuna ilişkin ölçüm. Küçük bir sayı, oranın gerçek orana yakın olduğunu gösterir; büyük bir sayı oranın gerçek orandan uzak olduğunu gösterir	$SE = \sqrt{(p * (1 - p)) * (\frac{1}{n_1} + \frac{1}{n_2})}$
Test İstatistiği (t)	Bir t istatistiği. Standart sapmaların sayısı, ortalamadan bir sayı uzaktadır.	$c = \frac{p_1 - p_2}{SE}$
İstatistiksel Anlamlılık	Test istatistiğinin mutlak değeri, ortalamanın 1,96* standart sapmasından büyükse, istatistiksel anlama sahip bir fark olarak kabul edilir	$ t  > 1.96$

Tez kapsamında, yapılandırma öncesi ve sonrasındaki her proje için hesaplanan ACD metrik değerleri üzerinden anlamlılık değeri hesaplanmıştır.

Proje 1 için ACD metriği için anlamlılık analizi aşağıda verilmiştir:

$$a_1 = 26 * 0.33$$

$$a_1 = 8.59$$

$$b_1 = 28 * 0.28$$

$$b_1 = 7.84$$

$$p = \frac{8.59 + 7.84}{26 + 28}$$

$$p = 0.304$$

$$SE = \sqrt{(0.304 * (1 - 0.304)) * (\frac{1}{26} + \frac{1}{28})}$$

$$SE = 0.08$$

$$t = \frac{8.59 - 7.84}{0.08}$$

$$t = |9.375| > 1.96$$

Yukarıdaki hesaplama göre, Proje 1 için ACD metrik değerindeki değişim anlamlıdır.

Proje 2 için ACD metriği için anlamlılık analizi aşağıdaki gibidir:

$$a_1 = 23 * 0.31$$

$$a_1 = 7.13$$

$$b_1 = 23 * 0.28$$

$$b_1 = 6.44$$

$$p = \frac{7.13 + 6.44}{23 + 23}$$

$$p = 0.242$$

$$SE = \sqrt{(0.242 * (1 - 0.242)) * (\frac{1}{23} + \frac{1}{23})}$$

$$SE = 0.09$$

$$t = \frac{7.13 - 6.44}{0.09}$$

$$t = |7.66| > 1.96$$

Yukarıdaki hesaplama göre, Proje 2 için ACD metrik değerindeki değişim anlamlıdır.

Proje 3 için ACD metriği anlamlılık analizi aşağıda verilmiştir:

$$a_1 = 19 * 0.32$$

$$a_1 = 6.08$$

$$b_1 = 21 * 0.29$$

$$b_1 = 6.09$$

$$p = \frac{7.13 + 6.44}{21 + 19}$$

$$p = 0.30$$

$$SE = \sqrt{(0.30 * (1 - 0.30)) * \left(\frac{1}{19} + \frac{1}{21}\right)}$$

$$SE = 0.102$$

$$t = \frac{6.08 - 6.09}{0.102}$$

$$t = |-0,09| > 1.96$$

Yukarıdaki hesaplama göre, algoritmaya göre t değeri 1.96 değerinden düşük olduğu için Proje-3'teki ACD metrik değerindeki değişim anlamlı değildir. Bu değişimin anlamlı olması için ACD metrik değerindeki düşüşün daha fazla olması gerekmektedir. Yani ürün hattındaki bileşenler arasındaki bağımlılığın düşürülmesi beklenmektedir.

Proje 4 için ACD metriği anlamlılık analizi aşağıdaki gibidir:

$$a_1 = 19 * 0.34$$

$$a_1 = 6.46$$

$$b_1 = 21 * 0.31$$

$$b_1 = 6.51$$

$$p = \frac{6.51 + 6.46}{21 + 19}$$

$$p = 0.32$$

$$SE = \sqrt{(0.32 * (1 - 0.32)) * \left(\frac{1}{19} + \frac{1}{21}\right)}$$

$$SE = 0.012$$

$$t = \frac{6.46 - 6.51}{0.012}$$

$$t = |-4.16| > 1.96$$

Yukarıdaki hesaplama göre, Proje 4 için ACD metrik değerindeki deęişim anlamlıdır.

Anlamlılık analizinde girdi olarak projelerin ACD Anlamlılık analiz sonuçlarına göre, Proje 1, Proje 2 ve Proje 4 için Özelik Ağacı metoduna göre yapılandırılması sonucu ACD metrik değerinde oluşan deęişim anlamlı iken Proje 3 için anlamlı değildir.

## 6. BULGULARA YÖNELİK DEĞERLENDİRME

### 6.1. Nicel Değerlendirme

Durum çalışması 1, 2, 3 ve 4 kapsamında gerçekleştirilen ve AS-1'e karşılık gelen Analiz-1 ile AS-2'ye karşılık gelen Analiz-2 gömülü durumlarından elde edilen sonuçlara göre LCOM, WMC, CC ve ACD metrik değerlerinin tutarlı olarak düştüğü görülmektedir.

WMC metriğinin düşmesi, bileşendeki ortalama karmaşıklık değerini düşürmektedir ve bileşenin toplam karmaşıklık değeri azalmaktadır. WMC değerinin yüksek olması, bileşenin uygulamaya özgü olma ihtimalini arttırmakta ve bu nedenle, yeniden kullanılabilirliği düşürmektedir. Yapılan durum çalışmaları 1, 2, 3 ve 4 kapsamında WMC değerinin düşmesi, geliştirme eforunun düşüp yeniden kullanılabilirliğin arttığı şeklinde yorumlanabilir.

LCOM metriğinin yüksek olması, bileşenin uyumluluğunun düşük olduğunu ve bileşenin alt parçalara bölünmesi gerektiğini göstermektedir. Düşük uyum karmaşıklığı arttıran bir özelliktir. Karmaşıklık ise yeniden kullanılabilirliği düşürmektedir. Bileşendeki yüksek uyumluluk, benzer özellikteki sınıfların bileşende toplandığı anlamına gelmektedir. Durum çalışmaları 1, 2, 3 ve 4 kapsamında bu değer düşmesi, yeniden kullanılabilirliğin arttığını göstermektedir.

CC metrik değeri, bileşenin karmaşıklığı hakkında fikir vermektedir. Durum çalışmaları 1, 2, 3 ve 4 kapsamında CC metrik değerinde düşme olduğu gözlemlenmiştir. Bu düşme, bileşenin yeniden kullanılabilirliğini artırıcı rol oynamaktadır.

Çizelge 6.1, durum çalışmaları kapsamında incelenen projelerin yapılandırma öncesi ve sonrası LCOM, WMC ve CC metrik değerlerini göstermektedir. Çizelgede özetlenen sonuçlara göre yapılandırma sonrası LCOM, WMC ve CC metrik değerlerinde düşüş olduğu gözlemlenmiştir.

Çizelge 6.1 Çoklu durum çalışmasında incelenen projelerin LCOM, WMC ve CC metriklerinin yapılandırma öncesi ve sonrasındaki değerleri

Proje	Bileşen	Yapılandırma Öncesi			Yapılandırma Sonrası		
		LCOM	WMC	CC	LCOM	WMC	CC
Proje-1	KontrolModel	0,337176	0,108326	0,37	0,33518	0,102625	0,20
	KontrolSunum	0,317333	0,092956	0,378134	0,30214	0,051918	0,021521
	MetroRaporu	-	-	-	0,43975	0,28458	0,02512
	MermiRaporu	-	-	-	0,484564	0,32656	0,1
Proje-2	KontrolModel	0,337176	0,108326	0,37	0,33518	0,102625	0,20
	KontrolSunum	0,2508434	0,171783	0,255	0,182515	0,05618	0,2325
	MetroRaporu	-	-	-	-	-	-
	MermiRaporu	-	-	-	-	-	-
Proje-3	KontrolModel	0,337176	0,108326	0,37	0,33518	0,102625	0,20
	KontrolSunum	0,289354	0,4623656	0,5623	0,2893548	0,4623656	0,4525
	MetroRaporu	-	-	-	-	-	-
	MermiRaporu	-	-	-	-	-	-
Proje-4	KontrolModel	0,337176	0,108326	0,37	0,33518	0,102625	0,20
	KontrolSunum	0,36521	0,65112	0,5623	0,2115	0,4	0,2
	MetroRaporu	-	-	-	-	-	-
	MermiRaporu	-	-	-	-	-	-

ACD, yeniden yapılandırmanın sisteme olan etkisi hakkında bilgi vermektedir. Durum çalışmaları 1, 2, 3 ve 4 kapsamında bu değer düşüğü görülmüştür. Bağımlılığın azalması yeniden kullanılabilirliğin arttığı şeklinde yorumlanmaktadır. Çizelge 5.6'da verilen anlamlılık analizinin algoritması kullanılarak Proje-1, Proje-2, Proje-3 ve Proje-4'ün ACD metriğinin yapılandırma öncesi ve sonrasındaki değişimindeki anlamlılık yorumlanmıştır. Yukarı belirtilen ACD metrik değerindeki düşüşün Proje-3 için anlamsız, Proje-1, Proje-2 ve Proje-4 için ise anlamlı olduğu sonucuna varılmıştır.

## 6.2. Nitel Değerlendirme

Bulgulara yönelik nitel bir değerlendirme yapmak için, anket çalışması yapılmıştır. Bu bölümde sunulan anket çalışmasının amacı, yeniden kullanılabilirliği arttırmak için önerilen Özellik Ağacı modeli kullanılarak gerçekleştirilen kod yeniden yapılandırmanın, gelecekte kurum genelinde yaygınlaştırılıp yaygınlaştırılmayacağını araştırmaktır. Çalışmada önerilen ve vaka çalışmaları uygulanarak örneklendirilen metodun kullanımındaki kazanımlar, öngörülen kolaylıklar ve zorluklar araştırılmaktadır.



Anket çalışmasında, hızlı bir şekilde veri alabilmek, bu verileri kolay bir şekilde tasnif ve analiz edebilmek için, çevrimiçi anket yöntemi kullanılmıştır. Önerilen çalışmanın genel geçerliğini inceleyen bu ankette yine Hedef-Soru-Metrik yaklaşımı kullanılmıştır. Bu anket çalışmasında 9 soru hazırlanarak OSGi çerçevesinde nesneye yönelik yazılım geliştiren ekiplerin görüşleri alınmıştır. Hazırlanan ankete EK-4' ten, anket sonuçlarına EK-5' ten ulaşılabilir.

Endüstri açısından anlamlı ve en kullanışlı tasarım ve uygulama adımları bilgileri katılımcıdan almak amacıyla sistematik olarak hazırlanan bu anketin süreçleri aşağıda açıklanmıştır. Anketin süreçleri soru hazırlama, geçeleme, anket yapısını oluşturma, very toplama ve very analizi adımlarından oluşmaktadır.

Soru hazırlama sürecinde, çalışmadaki deneyimler neticesinde gözlemlenen kazanım ve problemleri adresleyebilmek adına, taslak bir soru kümesi oluşturulmuştur. Bu aşama, dikkatli bir gözden geçirme süreci gerektirdiğinden, ekip lideri ile fikir alışverişi yapılmış ve en sonunda, araştırma sorularını adresleyen soru kümesi ortaya çıkmıştır.

Geçeleme amacıyla, anket direktörlükte farklı yazılım geliştirme projelerinde bulunup OSGi alt yapısı kullanarak Java ile nesneye yönelik geliştirme yapan ekiplere uygulanmıştır. Alınan geri bildirimler ve anket doldurmada geçen zaman da dikkate alınarak sorular tekrar düzenlenmiştir.

Ankette çoktan seçmeli, radyo düğmesi ve dereceli cevaplar kullanılmakla birlikte, bazı sorularda “Diğer” seçeneği ile katılımcıya özgür cevap imkânı da sunulmaktadır.

Anketin ilk sorusu katılımcı profilini anlamaya yardımcı olmaktadır. İkinci soru, katılımcıların geliştirdikleri yazılım projelerinde OSGi çerçevesine ne sıklıkta başvurduklarını anlamaya çalışmaktadır. Üçüncü soru, yazılım geliştiricilerin yazılımlarında yeniden kullanılabilirliği arttırmaya yönelik metot uygulayıp uygulamadıklarını, uygulanıyorsa hangi tekniğin uygulandığı bilgisini sorgulamaktadır. Dördüncü soruda yazılım geliştiricilerin yeniden kullanılabilirliği arttırmaya yönelik uyguladıkları tekniğe ne sıklıkta başvurdukları araştırılmaktadır. Beşinci soruda Özellik Ağacı metodu ile yeniden kullanılabilirliği arttırmaya yönelik geliştirilen metodun uygulanabilirliği sorgulanmaktadır. Altıncı soruda ise Özellik

Ađacı metodu ile yeniden kullanılabilirliđi arttırmaya yönelik geliřtirilen metodun uygulama kolaylıđı sorgulanmaktadır.

Anketin hedef kitlesi OSGi çerçevesi kullanarak Java programlama dili ile yazılım geliřtiren ekiplerdir. Anketin hazırlanması için Google Form seğıilmiřtir.

Anket sürecinin en son ařaması olan veri analizi, katılımcıların geliřtirdikleri uygulamalar için verdikleri cevapların incelenmesidir. EK-5' ten anket çalıřmalarının sonucuna ulařılabilir.

Düzenlenen anket çalıřması neticesinde, OSGi çerçevesinde ürün hattı yaklařımına dayalı modüler geliřtirme yapan ekiplerin, tez kapsamında sunulan "Özellik Ađacı ile yeniden kullanılabilirliđi arttırma" metodunu kullanılabilir olarak deđerlendirdiđi sonucuna varılmıřtır. Anket katılımcılarının yazılım geliřtirici pozisyonunda çalıřtıkları görölmektedir. Yeniden kullanım tekniklerinin çođunlukla kaynak kod üzerinde uygulandıđı sonucuna varılmaktadır. EK-4'te anketin kendisine ve EK-5' te anket çalıřmasının sonuçlarına ulařılabilir.

## 7. GEÇERLİLİĞE YÖNELİK TEHDİTLER

Eksik veri setleri ve hatalı veri çıkarımları, geçerliliğe tehdit oluşturan temel konuların başında gelmektedir. Bu bölümde çalışmada karşılaşılan ve geçerliliğe yönelik oluşan tehditlere getirilen çözümler anlatılmaktadır.

### 7.1. İçsel Tehditler (İng. Internal Validity)

Bağımlı değişken üzerindeki değişimin gerçekten bağımsız değişkenden kaynaklı olup olmadığı ile ilgilenen geçerlilik türüdür. İçsel tehditleri engellemek için, durum çalışmaları kapsamında incelenen OSGi çerçevesi kullanılarak geliştirilmiş tüm yazılım projelerimizde önceden tanımlanan metot uygulanmıştır. Durum çalışmaları için protokol hazırlanarak ve veriler araçlarla otomatik toplanarak araştırmacının inceleme üzerindeki etkisi en aza indirilmeye çalışılmıştır. Ayrıca gerek literatür taraması yapılırken gerekse özellik ağacı çıkarılırken akran gözden geçirme uygulanarak iç tehditler azaltılmıştır.

### 7.2. Yapısal Tehditler (İng. Construct Validity)

Çalışmada çalışmanın hedefi ile araştırma soruları arasındaki izlenebilirliği takip etmek önemlidir. Bu hedefi gerçekleştirmek için çalışmamızda Hedef Soru Metrik yaklaşımına başvurulmuştur. Araştırma sonuçları hedeflere yönelik oluşturularak çalışmadaki olası yapısal tehditler sınırlandırılmıştır. Gerçekleştirilen literatür araştırmalarının tekrar edilebilir olması için, Sistemik Haritalama metodolojisi kullanılmıştır. Ayrıca, sistemik haritalama çalışması ile elde edilen metriklerin durum çalışmalarına temel alınması, yapısal tehdidi azaltmıştır. Veri toplanırken verinin bütünlüğünü ve doğruluğunu bozmamak için Understand ve Sonargraph gibi araçlar kullanılarak otomatik toplama yapılmıştır. Sınıf bazındaki metrikleri bileşen seviyesine taşırken istatistiksel normalizasyon uygulanmıştır.

### 7.3. Sonuçla İlgili Tehditler (İng. Conclusion Validity)

Sonuçla ilgili tehditleri azaltmak için ACD metriğinin değişimi üzerinde anlamlılık analizi yapılmıştır. LCOM, WMC, CC ve ACD gibi birden çok ve farklı metrikler kullanılarak ve Proje 1-4 üzerinde çoklu durum çalışması yapılarak sonuca yönelik tehditlerin azaltılması amaçlanmıştır. Ayrıca farklı kaynaklardan veri toplanmıştır. Nicel yöntemlere ek nitel yöntem uygulanarak anket çalışması yapılmış ve yeniden

kullanılabilirlik için önerilen yöntemle ilişkin firma çalışanlarından geribildirim alınmıştır.

#### **7.4. Dışsal Tehditler (İng. External Validity)**

Bu tez çalışmasında önerilen yöntem, sadece bir tek kurum içinde çoklu durum çalışması yapılarak doğrulanmıştır. Elde edilen sonuçların geçerliliği bu bağlamla sınırlıdır ve genellenemez. Dış geçerliliği güvence etmek için, ürün hattındaki bileşenlerin yeniden kullanılabilirliğini attırmak amacıyla önerilen bu yöntem, farklı firmalarda ve bağlamlarda tekrarlanarak sonuçları incelenmelidir.

## 8. SONUÇLAR

Tez kapsamında OSGİ çerçevesinde geliştirilmiş bileşenlerin oluşturduğu ve ürün hattı yaklaşımına dayalı olarak oluşturulmuş radar kullanıcı arayüzü yazılımlarının yeniden kullanılabilirliğin artırılması hedeflenmiştir.

Literatürde önerilen yeniden kullanılabilirliği değerlendirme ve artırma hedefli yaklaşımlara bakıldığında:

- Metrik ve metotların birçoğu sınıf seviyesinde kalmaktadır.
- Yeniden kullanılabilirlik farklı faktörlere bakılarak farklı metriklerle değerlendirilmektedir.
- Yaklaşımların çoğu yeniden kullanılabilirliği beyaz kutu yaklaşımı ile ele almaktadır.
- Yapılan araştırmaların sonucuna göre OSGİ servis platformu ve Ürün Hattı mimarisine dayalı olarak geliştirilmiş nesneye yönelik bir yazılımın yeniden kullanılabilirliği değerlendiren metrik ya da metoda rastlanmamıştır.

Bu kapsamda ilk olarak yeniden kullanılabilirliği niteleyen faktörleri belirlemek için sistematik haritalama metodu ile literatür taraması yapılmıştır. Bağlaşım, uyumluluk ve karmaşıklık özellikleri, bir yazılımın yeniden kullanılabilirliğini en çok etkileyen yazılım faktörleri olarak karşımıza çıkmıştır. Bağlaşım, karmaşıklık ve uyumluluk faktörlerini sayısallaştırmak için LCOM, WMC, CC ve ACD metrikleri kullanılmıştır. ACD dışında metriklerin hepsi sınıf seviyesinde olduğu için bileşen seviyesinde değerlendirme yapabilmek amacıyla metrik değerleri önce normalize edilmiş sonrasında bileşen içindeki tüm sınıflardaki normalize edilmiş metrik değerlerinin ortalaması alınmıştır.

Yeniden kullanılabilirliği arttırmak amaçlı olarak Özellik Ağacı modelinden faydalanılmıştır. Özellik ağacına göre yeniden yapılandırılan Proje-1, Proje -2, Proje-3 ve Proje-4 için yapılandırma öncesi ve sonrası için LCOM, WMC, CC ve ACD metrik değerleri alınmıştır. ACD metriği Ürün Hattı için ortalama bir değer sunmakta olup LCOM, WMC ve CC metrikleri sınıf seviyesinde edinilen metrikler olduğu için ortalama ve normalizasyon teknikleri uygulanarak bileşen seviyesine taşınmıştır.

Yapılandırma öncesi Proje-1, Proje-2, Proje-3 ve Proje-4'te bulunan bileşen grupları ve yapılandırma sonrası Proje-1, Proje-2, Proje-3 ve Proje-4'te bulunan bileşen gruplarının ACD metrik değerindeki değişimin anlamlılığını ölçmek için anlamlılık analizi kullanılmıştır. Proje-1, Proje-2 ve Proje-4 için değişimin anlamlı Proje-3 için ise değişimin anlamsız olduğu görülmüştür.

Gerçekleştirilen Sistemik Haritalama ve anket çalışması ile nitel ve nicel değerlendirme yapılmıştır. Anket, OSGi çerçevesi ile Ürün Hattı mimariye dayalı geliştirme yapan ekiplere uygulanarak metodun uygulanabilirliği ve kullanılabilirliği analiz edilmiştir.

## 9. GELECEK ÇALIŞMALAR

Gelecek çalışmalar kapsamında, Özellik Ağacı çıkarma otomatik olarak ve Doğal Dil İşleme Tekniği kullanılarak yapılacaktır. Doğal dil işleme sonucunda çıkarılan özellik ağacının kalitesi ile hali hazırda tezde sunulan uzman görüşüne dayalı olarak çıkarılan özellik ağacının kalitesi karşılaştırılacaktır.

Yine gelecek çalışmalar kapsamında, tezde kullanılan Sonargraph ve Understand araçları haricinde, daha yüksek seviyeden ölçüm alınmasına olanak veren araçlar araştırılarak farklı metriklerle yazılımların yeniden kullanılabilirlik değerlendirmesi yapılacaktır.

Jira'da yapılandırılan projeler takip edilerek açılan hata kayıtlarının istatistikleri yapılandırma öncesi kayıtların sayısı ile karşılaştırılacaktır. Böylece yapılandırma sonrası projelerdeki hata sayıları tahlil edilerek metodun işlevselliği tartılacaktır.

Çalışma kapsamında ürün metrik değerleri alınmıştır. Gelecek çalışma olarak süreç metrik değerleri de sayısallaştırılacaktır. Hata maliyeti (İng. cost of quality), hata yoğunluğu (İng. defect density), hata giderme etkinliği (İng. defect removal efficiency) gibi ölçümler alınacaktır.

Ayrıca bu çalışmada ele alınan, alan mühendisliğinden uygulama mühendisliğine bileşen aktarılması durumuna ek olarak, projelerdeki ortak özelliklere göre uygulama mühendisliğinden alan mühendisliğine bileşen aktarılması durumu da ayrıca incelenebilecek konular arasındadır.

## EK-1: Sistematik Haritalama ile İncelenen Araştırma Soruları ve Yanıtları

Hedef	Bileşen-tabanlı ve ürün-hattı yazılım geliştirme yaklaşımlarında yeniden kullanılabilirliği ölçmek için yapılan çalışmaları anlamak	
	Soru (AS-1)	Bileşen-tabanlı ve ürün-hattı yazılım geliştirme yaklaşımlarında yeniden kullanılabilirliği ölçmek için hangi metrikler önerilmiştir? Bu metrikler hangi yöntemlerle birlikte kullanılmıştır?
	Metrikler (AS-1'in yanıtları)	<p>Bileşenlerin Tam Yeniden Kullanılabilirliği (İng. COR - Component Overall Reusability)</p> <p>Bulanık temelli yaklaşım, Nöro bulanık temelli yaklaşım, Hibrit yaklaşım kullanarak yeniden kullanılabilirlik seviyesini belirleme (İng. Level of reusability using: Fuzzy-based approach, Neuro-fuzzy based approach, Hybrid fuzzy-GA based approach)</p> <p>Kalite Sistemi (İng. Quality Framework) f(u) – Yeniden kullanılabilirlik modeli, yeni bir metrik sistemi, Segueno tip bulanık arayüz sistemi (İng. Reusability using Segueno-Type Fuzzy Inference System, Reusability model, new metrics framework {EMI, RCO, RCC, SCC, SCCp})</p> <p>Nöro bulanık arayüz sistemi, Bulanık yaklaşım kullanarak yeniden kullanılabilirliğin seviyesi (İng. Level of reusability using: Neuro-fuzzy inference system, Soft Computing Fuzzy approach)</p>
	Soru (AS-2)	Bileşen-tabanlı ve ürün-hattı yazılım geliştirme yaklaşımlarında yeniden kullanılabilirliği ölçmek için yapılan çalışmalarda, yeniden kullanılabilirliği etkileyen faktörler nelerdir?
	Metrikler (AS-2'nin yanıtları)	<p>Adapte olabirlik (İng. Adaptability), Bağlaşım ve Yapışkanlık (İng. Coupling, Cohesion), Karmaşıklık (İng. Complexity), Uyarlanabilirlik (İng. Customizability), Tamamlanabilirlik (İng. Completeness), Gizlilik (İng. Confidence), Ayarlanabilirlik (İng. Configurability), Uygunluk (İng. Compatibility), Ortaklanabilirlik (İng. Commonality), Birleştirilebilirlik (İng. Composability), Anlaşılabilirlik (İng. Comprehensibility), Dokümantasyon (İng. Documentation), Alan İçeriği (İng. Domain context), Extent of templating, Kullanımda kolaylık (İng. Ease-of-use), Fonksiyonel bağımlılık (İng. Functional dependencies), Fonksiyonellik (İng. Functionality), Kalıtım (İng. Inheritance), Arayüz karmaşıklığı (İng. Interface complexity), Arayüzlerin sağlamlığı (İng. Interface soundness), Bakım kolaylığı (İng. Maintainability), Modülerlik (İng. Modularity), Taşınabilirlik (İng. Portability), Güvenilirlik (İng. Reliability), Yeniden kullanım frekansı (İng. Reuse frequency), Düzenlilik (İng. Regularity), Bağlaşımın ayrılması (İng. Separation of Concerns), Büyüklük (İng. Size), Nakledilebilme (İng. Transplanting), Anlaşılabilirlik (İng. Understandability), Kullanılabilirlik (İng. Usability), Yararlılık (İng. Usefulness), Kullanılabilirlik (İng. Utilizability), Değişkenlik (İng. Variability), Hacim (İng. Volume)</p>



## EK-2: Sistematik Haritalama ile İncelenen Çalışmalar ve Metrikler

Ref.	Yazılım Özellikleri	Yazılım Metrikleri	Yeniden Kullanılabilirliği Ölçme Yöntemleri ve Metrikleri
[31]	Portability, Understandability, Adaptability	Existence of Meta Information (EMI), Rate of Component Customizability (RCC), Self-Completeness of Component's Return Value (SCCr)	COR - Component Overall Reusability
[32]	Coupling	DD - Component Dependency Metric	-
[33]	Complexity, Regularity. Volume, Reuse Frequency, Coupling	McCabe's Cyclomatic Complexity, Regularity metric, Halstead Software Science Indicator, Reuse frequency metric, LCOM	Level of reusability using: Fuzzy-based approach, Neuro-fuzzy based approach, Hybrid fuzzy-GA based approach
[34]	-	-	Quality Framework
[35]	Coupling, Volume, Complexity, Regularity, Reuse Frequency	McCabe's Cyclomatic Complexity, Halstead Software Science Indicator	f(u) - Reusability using Segueno-Type Fuzzy Inference System
[36]	Adaptability, Completeness, Maintainability, Understandability, Complexity, Volume, Inheritance, Coupling	McCabe's Complexity, Halstead's Volume, NPSS (Number of Physical Source Statements, Depth of Nested Statements, WMC (Weighted Methods per Class), Depth of Inheritance, Number of immediate children, RFC (Response for a Class), Coupling between objects (CBO).	-
[37]	Understandability, Adaptability, Portability	-	Reusability model, new metrics framework {EMI, RCO, RCC, SCC, SCCp}
[38]	Coupling, Cohesion, Inheritance	CK metrics (WMC, DIT, NOC, CBO, LCOM)	Level of reusability using: Neuro-fuzzy inference system
[39]	Coupling, Cohesion, Inheritance	CK metrics (WMC, DIT, NOC, CBO, LCOM)	Neuro-fuzzy inference system
[40]	Customizability, Interface complexity, Portability, Documentation	-	Soft Computing Fuzzy approach
[41]	Coupling, Cohesion, Inheritance	CK metrics (WMC, DIT, NOC, CBO, LCOM), Regression algorithms	-
[42]	Modularity, Coupling, Cohesion, Interface complexity, Documentation, Complexity, Size	Modularity = f(Lack of Coupling, Cohesion), Interface size = Adjusted Number of public methods, Documentation = A degree of documented lines of code, methods, attributes,	Reusability = f(Modularity, Interface Size, Documentation, Complexity)

		Complexity = f(Size, Complexity of Methods)	
[43]	Complexity, Regularity, Volume, Reuse Frequency, Coupling	-	Neuro-fuzzy inference system
[44]	Cohesion, Coupling	COHOT - Cohesion Metric by Operation Type, COPOT - Coupling Metric by Operation Type	-
[45]	Reusability in accordance to Reuse level	-	$CRL_{LOC}$ – Component Reuse Level
[46]	Domain Context	-	$R = f(Rf)$ , $R = f(Rf, Cu)$ , $R = f(Rf, Cu, Rm)$
[47]	Coupling	CC - Coupling between Classes, CPC - Coupling of Component, CBC - Coupling between Component	-
[48]	Customizability, Configurability, Interface complexity, Portability, Compatibility	-	Fuzzy inference system
[49]	Commonality, Modularity, Adaptability, Comprehensibility, Composability, Interface soundness	CoS - Cohesiveness of Services, MoD - Minimality of Dependencies, CoV - Coverage of Variability, CoA - Completeness of Variant Set, ESA - Effectiveness of Service Adaptation, ICC - Inter-CaaS Composability, CAC - CaaS-to-Application Composability	IR – Integrated reusability
[50]	Complexity, Volume, Regularity, Reuse Frequency, Coupling	McCabe's Cyclomatic Complexity metric, Halstead Software Science Indicator, Regularity Metric, ReuseFrequency Metric, LCOM	Neural Network - Resilient Backpropagation algorithm (RB)
[51]	Understandability	APP - Arguments per Procedure, DAC - Distinct Argument Count, DAR – Distinct Argument Ration, ARS - Argument Repetition Scale, MSC - Mean String Commonality, MIL - Mean Identifier Length, RAD - Reference Argument Density	-
[52]	Coupling, Cohesion	WTcoh - Weighted Transitive Cohesion measure, WTcoup - Weighted Transitive Coupling measure	-
[53]	Coupling, Cohesion,	CK metrics (WMC, DIT, NOC,	Neural Network - Levenberg &

	Inheritance	CBO, LCOM)	Marguardt algorithm
[54]	Coupling, Cohesion, Inheritance	CK metrics (WMC, DIT, NOC, CBO, LCOM)	K-means clustering algorithm & Decision Approach
[55]	Separation of Concerns, Size, Cohesion, Coupling	-	-
[56]	Cohesion	PCoh - Package Cohesion	-
[57]	Coupling	CC1 - CC8 (Coupling Counts)	-
[58]	Functionality, Reliability, Maintainability, Ease-of-use, Transplanting	-	RV - Reusability Value based on Colony Decision
[59]	Functionality, Reliability, Maintainability, Utilizability, Portability		RMV (Reusability Measure Value)
[60]	Adaptability, Composability	Component Adaptability (Ac(kc)), Composability Degree of Component	-
[61]	Reuse Frequency, Understandability, Rate of modification	-	$R = f(Rf)$ , $R = f(Rf, Cu)$ , $R = f(Rf, Cu, Rm)$
[62]	Cohesion	SBFC - Similarity-Based Functional Cohesion	-
[63]	Adaptability, Availability, Complexity, Documentation, Maintainability, Price, Reuse, Quality	-	R(C) - Reusability of component
[64]	Interface complexity	No. public members in component interface	R(C) - Reusability of component

<b>EK-3: Özellik Ağacı</b>					
<b>Yetenekler</b>	<b>Açıklama</b>	<b>PROJE 1</b>	<b>PROJE 2</b>	<b>PROJE 3</b>	<b>PROJE4</b>
<b>Metro raporu</b>					
<b>Atış tanzim</b>					
<b>BAĞLANTI YETENEKLERİ</b>					
Bağlantı Yönetimi Yetenekleri	Kullanıcı arayüzünden bağlantı parametrelerinin belirlenmesi				
<b>RADAR DURUMU YETENEKLERİ</b>					
CİT Sunum Yeteneği	Radardan gelen birimlerin hata kodlarının sunumu	X	X	X	
Bağlantı durumları sunum	Bağlantılar listesi, devam eden ve henüz başlatılmamış bağlantılar	X	X	X	
Sıcaklık Sunum Yeteneği	Cihaz birim sıcaklıklarının derece olarak sunumu	X	X		
Birimlerin güç durumlarının sunumu					
Radar üzerindeki anahtar durumlarının sunumu					
<b>MÜHENDİSLİK YETENEKLERİ</b>					
Algoritma Seçim Yeteneği	CFAR, hüzme planlama vb. algoritmaların çalışma zamanında seçilebilir olması				
B-skop penceresi		X	X	X	
C-skop penceresi		X	X	X	
B-C skop yeteneği					
Ölçüm gösterimi	Tüm ölçümlerin, izi güncelleyen ölçümlerin gösterimi	X	X	X	
Ölçüm raporu sunum	KGÜ ekranında seçilen ölçümlerin ölçüm raporları listelenecektir.		X		
Veri kayıt	Kaydedilen verilerin tekrar oynatılmasıdır.				
Doppler sunum	Seçili hedefin Doppler spektrumu ekranda görülecektir.				
Hedef grafik çizim	Ekranda hedef tespitleri grafik çizimi halinde gösterilecektir.				
Video fading	Video güncellemede kaydırma özelliği olacaktır.				
Ayarlanabilir sabit eşik	Mühendislik amaçlı ekranın ayarlanabilir sabit eşik durumu yeteneği olacaktır. (STT_1287)				
Sensör Grubu'nun düzeyleme bilgisinin sunumu	Sensör Grubu'nun düzeyleme bilgisini KGÜ'de gösterilecektir.				

Arıza lokasyonu sunumu	Sistem'de bakım onarım maksatlı arıza lokalizasyonu bilgisi veren arayüz bulunacaktır.				
<b>EĞİTİM YETENEKLERİ</b>					
Senaryo Tanımlama Yeteneği	Açılış, anten dönüşü, ışıma başlatma, mod seçimi				
Tanımlı Senaryo Gerioynatım Yeteneği	Radar bulgularının sunumu .Radar bulgularının girilen ayarlara (arama ve karartma sektörleri, menzil modu) uygun olması				
Demo modu	Demo modunda simülatör ile tespit oluşturup gösterme	x	x	x	
<b>RADAR PLATFORM KONUMU BELİRLEME YETENEĞİ</b>					
Kullanıcı girişi	Kullanıcının el GPS'i ya da telsiz ile öğrendiği konumu doğrudan girmesi	x	X		
Dış INS / GPS cihazı	GPS bağlantısı üzerinden otomatik konum aktarımı				
<b>KUZEY KALİBRASYONU YETENEĞİ</b>					
Otomatik kuzey kalibrasyonu		x	X		
Manuel kuzey kalibrasyonu		x	X		
Kayıtlı kuzeyi kullan		X			
<b>RADAR GÖREV TANIMLAMA YETENEKLERİ</b>					
	Radarın belirli süre aralığı içinde belirli coğrafi bölgeler üzerine ışıma yapması, menzil modu ve çıkış gücünün bu zaman aralığına özel seçilebilir olması				
<b>YAYIN AYARLARI</b>					
Anten döndürme durdurma			X	x	
Anten dönüş yönü seçimi			X	x	
Işıma başlatma / durdurma		x	X	x	
HAH modu aktifleştirme/durdurma	Havada asılı helikopter modu (checkbox)		X		
LPI modu seçimi			X		
Anteni yanca açısına döndürme	KGÜ Arayüzü'nden AAS'yi hareket ettirmek istenen yanca açısının girilmesi			X	

Nakliye açısına dödürme	KGÜ'den nakliye açısına git butonuna basılır				
Anten Diklik Açısına Gitme Senaryosu	Mühendis, KGÜ Arayüzü'nden AAS'nin diklik açısını (eğimini) girer.				
<b>SEKTÖR TANIMLAMA YETENEKLERİ</b>					
Karartma/kırpma Sektörleri	Karartma ve kırpma sektörlerini yancada tanımlayabilme.	X	X		
Kalıcı karartma/kırpma sektörü tanımlayabilme	Sistem kapatılıp açıldıktan sonra parametreleri değişmeyen				
Arazi silüeti sektörleri	Harita yükleme ve elle sektör(başlangıç-bitiş açıları ve yükseliş) girişi				
HAH Sektörleri	Başlangıç-bitiş yancaları		X		
Hacim Arama Sektörleri	Radarın yan ve yükseliş aralığında hüzme atması				
Ufuk Hattı Arama Sektörleri					
Kapsama modu sektörü	Kapsama modları yanca açıları ile belirlenmiş sektörler seçimli olarak KGÜ'den girilebilecektir.		X		
PRI Jitter/Stagger Sektörü			X		
Öncelikli Bölge Sektörü	Öncelikli bölgeler arayüzden enlem boylam ve yarıçap bilgileri ile KGÜ'den girilebilecektir.	X			
Yükseliş sektörü tanımı	Radarın yükselişte ışıma yapacağı sektörlerin belirlenmesi ve C skopta sunumu.		X		
<b>METRO RAPORU YETENEKLERİ</b>					
Metro raporu sunumu	Operatörün meteoroloji raporu bilgisini girebilmesi için arayüz sunulacaktır.	X			
Metro raporunun KKS'den alınması	KKS sisteminden alınan metro raporu operatöre sunulacaktır.	X			
<b>OTOMATİK TAKİP BÖLGESİ YETENEKLERİ</b>					
	Yan ve yükseliş aralığı içine düşen ölçümlerden otomatik iz başlat yeteneği/başlatma, operatör el ile seçip başlatsın yeteneği				
<b>MANUEL TAKİP BAŞLATMA YETENEĞİ</b>					
	Yan ve yükseliş aralığı içine düşen ölçümlerden otomatik iz başlatma, operatör el ile seçip başlatsın yeteneği			X	
<b>RAPORLAMA YETENEKLERİ</b>					

Rapor Sayısı Sınırlama Yetenekleri	Kullanıcının girdiği süreden daha eski bulguların otomatik olarak sunumdan çıkarılması, kullanıcının girdiği tur sayısından daha eski ve / ya da yanca aralığı dışındaki bulguların sunumdan çıkarılması				
Rapor Filtreleme Yetenekleri	. Operatörün girdiği hız aralığı dışında olan hedeflerin filtrelenmesi . Tespitlerin gösterilmesi / gizlenmesi . Ölçümlerin gösterilmesi / gizlenmesi			X	
Raporlama Bölgesi Yetenekleri	Ölçüm, iz raporlarının hangi bölgelerde çıkartılacağı				
<b>PLATFORM BİLGİLERİ SUNUM YETENEĞİ</b>					
	Hareketli platformun konumunun haritada sunumu Hareketli platformun konumunun bir tabloda sunumu				
<b>İZ DÜŞÜRME YETENEĞİ</b>					
Otomatik iz düşürme	Zaman aşımında izlerin otomatik düşürülmesi		X		
Manuel iz düşürme	Operatör seçimi ile seçili veya tüm izlerin düşürülmesi.		X	X	
Hızlı kara taşıtlarını eleme	Hızlı kara taşıtlarını eleme aktif/inaktif				
Tüm izleri düşürme			X		
<b>İZ ÖNCELİKLENDİRME YETENEĞİ</b>					
İz Önceliklendirme			X		
İz Önceliği düşürme			X		
<b>İZ BİLGİLERİ SUNUM YETENEĞİ</b>					
Temel yetenek	Menzil, hız, yön, yükseliş açısı, irtifa, radara göre yanca, kimlik numarası, teşhis (dost / düşman), sınıflandırma (araç tipi)		X	X	
İz yolu gösterimi	Seçilen izin izlediği yolun gösterimi		X		
Sembol büyüklüğü ayarlama					
Sesli uyarı	Yeni iz bilgisi ve karıştırıcı bilgisi alındığında sesli uyarı verilecektir.			X	
<b>ÖLÇÜM (PLOT) SUNUM YETENEĞİ</b>					
	Hız, yön, teşhis bilgisi olmayan, menzil, genlik, yanca bilgileri olan, tespitlerin gruplanarak elde edilen, aday iz bulgusudur. B, C, PPI skoplarda sunulur, kullanıcı bu grafiklerde seçim yapar, seçilen ölçüm parametreleri bir raporda sunulur.	X	X	X	

<b>KARIŞTIRICI SUNUM YETENEĞİ</b>					
	Yanca ve genlik bilgisi içeren bir radar bulgusudur. Skoplarda düz çizgi olarak, belli bir yancayı gösterecek şekilde sunulur.	X	X		
<b>GEZİNGE SUNUM YETENEĞİ</b>					
Haritada gezinge sunumu	Havan Gezinge mesajı içindeki gezinge bilgisini kullanılarak bir çıkış bir de düşüş konumu hesaplanacak raporda sunulacaktır.	X			
Atış kontrol listesinde gezinge bilgilerinin sunumu		X			
<b>ATIŞ KONTROL YETENEKLERİ</b>					
Atış Kontrol Yeteneği				X	
<b>ATIŞ KALİBRASYON YETENEĞİ -ATIŞ TANZİM YETENEĞİ</b>					
<b>Kara Atış Kalibrasyonu Yeteneği</b>					
Kara Atış Kalibrasyonu Yeteneği	Mermi takibi, merminin çarptığı noktanın tespiti ve imha değerlendirmesi. Kara tehditleri için.	X		X	
<b>Suüstü Atış Kalibrasyonu Yeteneği</b>					
Suüstü Atış Kalibrasyonu Yeteneği	Mermi takibi, merminin çarptığı noktanın tespiti ve imha değerlendirmesi. Su üstü tehditler için.				
<b>Hava Atış Kalibrasyonu Yeteneği</b>					
Hava Atış Kalibrasyonu Yeteneği	Mermi takibi, merminin çarptığı noktanın tespiti ve imha değerlendirmesi. Hava tehditleri için.				
<b>IFF YETENEKLERİ</b>					
IFF Kontrol Yetenekleri	Kod sıfırlama, otomatik/manuel/lethal sorgu başlatma, sorgu parametrelerini belirleme, kod çözme ayarları. Mühendislik modunda ek yetenekler		X		
<b>FREKANS SEÇİMİ YETENEKLERİ</b>					
Frekans seçim	Radarin görev yapacağı frekans kanallarının seçimi. Operatöre sabit sayıda kanal numarasının bulunduğu bir liste sunulur ve radarın bulunduğu birlik ya da üst komutanlıkça belirlenen frekansları operatör seçer. Seçili frekans skoplarda sunulmaz.	X	X		
Frekans atlama modu seçimi	Sabit veya Frekans Atlama				
<b>FREKANS KALİBRASYONU YETENEKLERİ</b>					



	Radarın üretim, bakım ve test amaçlı frekans gönderim ayarlarının yapılmasıdır. Bir frekans kanal - genlik tablosu ve kalibrasyon başlatma butonundan oluşur. Tabloda sunulan veriler skoplarda sunulmaz.				
YAYIN ÇIKIŞ GÜCÜ KONTROL YETENEĞİ					
	Tehdit durumuna göre radarın yüksek ya da düşük güç ile çalışabilmesi operatör tarafından seçilebilmektedir. Bir yanca aralık listesi ve listedeki her eleman için seçilmiş güç değerinden oluşur.				
BİRİM GÜÇLERİ KONTROL YETENEĞİ					
ÇEKİRDEK SEÇİM YETENEĞİ					
	Bakım ve test amaçlıdır. Radarın hangi donanım işlemci çekirdekleri ile çalışacağı bilgisidir. Bir tablo şeklinde sunulmaktadır. Operatör tablodan seçim yapar.				
HIZLI HEDEF ALARMI YETENEĞİ					
	Operatörün girdiği hız üst sınırından daha hızlı olan hedefler için verilen sesli ve görsel uyarıdır. Dost birliklere doğru manevraya geçen tehditlerin daha çabuk algılanması amacıyla çıkarılır.			X	
OLUK ETKİSİ ALTINDA ÇALIŞMA YETENEĞİ					
	Bir yanca sektör aralığı listesidir. Bu liste aralığında radar farklı bir modda çalışır. Bu sektörler skoplarda da sunulur.				
ARAZİ SİLÜETİ YETENEĞİ					
Arazi silüeti hesaplama yeteneği	Radarın konuşlu bulunduğu noktadan yancada 360 derece ufka bakıldığında bulunan coğrafi engellerin en tepe noktalarının radar ile yaptığı yükseliş açılarının tespit edilmesidir. Bulunan yükseliş açıları bir tablo ile operatöre sunulur. PPI skopta da işaretlenir.	X			
UZAYDA RASTGELE HUZME YERLEŞTİRME SEÇİM YETENEĞİ					
ED KARARTMA YETENEĞİ					
YAKIN ALAN TESPİT YETENEĞİ					

<b>MENZİL - GENLİK SKOP YETENEKLERİ</b>					
	Radarin, operatörün seçtiği bir tespitin bulunduğu yönde bir menzil boyunca tespit ettiği genlikleri sunan 2 boyutlu bir grafikdir. Yatay eksen menzil, dikey eksen genlik olarak belirlenir. Grafiğin peek yaptığı yerlerin birbiri ile karşılaştırılır ve gerçek tehditler ayırt edilir.				
<b>YANCA - MENZİL SKOP YETENEKLERİ</b>					
Sembol, Çizgi ve Bölge Tanımlama Yetenekleri	Skop üzerinde sembol (coğrafi nokta) sunumu, yanca ve menzil aralığı giriş ve sunumu, PPI skop için poligon ve elipsoid bölge seçim ve sunumu				
B-Skop Yetenekleri	Yanca ekseni marker sayısı, menzil ekseni marker sayısı, yancada kaydırma, 0 derece etrafında kaydırma, zoomlama yetenekleri, radar bulgularının sunumu	X	X		
<b>PPI-Skop Yetenekleri</b>					
PPI-Skop Yetenekleri	Menzil çember sayısı, yanca marker sayısı (aralığı), yanca birim tipi (derece, milyem), skop tipi (gerçek, izdüşüm), merkez kaydırma ve zoomlama yetenekleri, skop üzerinde görünür olan, olmayan tespit sayılarının sunumu, radar bulgularının sunumu, Antenin baktığı yancanın gösterimi	X			
İmleç konumu	Hava resmi üzerindeki imlecin radara göre mesafesi ve istikamet açısı		X		
<b>YANCA - YÜKSELİŞ SKOP YETENEKLERİ</b>					
	Yanca ekseni marker sayısı, yükseliş ekseni marker sayısı, yanca ve yükselişte 0 derece etrafında kaydırma, zoomlama yetenekleri, yanca ve yükseliş aralığı belirleme ve sunumu, radar bulgularının sunumu		X		
<b>HARİTA YETENEKLERİ</b>					
Harita üzerinde PPI Skop Sunum Yeteneği	Yukarıda anılan PPI skop yeteneklerinin altında katman olarak harita sunulmasıdır.			X	
DTED2 harita sunumu	Yükseklik verisinin renklendirme ile sunumudur.	X			
Geotiff harita sunumu	Resim (yerleşim yerleri, yollar, ormanlık alanlar, yükselti eğrileri) harita sunumudur.	X			
Vektör harita sunumu	Ülke, il, kıyı - kara sınırlarıdır.				

Sembol ve bölge gösterimleri	Coğrafi nokta, yanca - menzil aralığı, polojin bölgeler, çembersel bölgeler, hat (çizgi) bölgeleri, kara - kıyı ayırım çizgileri, sınır çizgileri	X			
Harita yükleme yetenekleri	Geotiff ve DTED2 türde haritaların sisteme dışarıdan kopyalanabilmesidir.	X			
Otomatik harita gösterimi	Haritada merkez kaydırma ve zoomlama yaptıkça önceden yüklü bulunan haritalardan yalnız odaklanılan coğrafi bölgeye ait olanların otomatik olarak görünür duruma gelmesidir.	X			
Sisteme harita yükleme yeteneği	Gömülü yazılıma KGÜ arayüzünden harita gönderim yeteneği				
<b>DOPPLER SPEKTRUM VE SES YETENEĞİ</b>					
	Bir menzil - genlik grafiği ile kara tehditlerinin sınıflandırılabilmesine yarar. Ses alım ve operatöre dinletilmesi yeteneğini içerir. Operatör gelen sesteki tehdit değerlendirme (taşıt, yaya) yapar.			X	
<b>RADAR VİDEOSU YETENEKLERİ</b>					
	PPI skop üzerinde sunulan yanca - menzil aralıkları ve her aralık için tespit edilen genlik değerinin bir renk ve parlaklık ile sunumudur. Yüksek genlikli bölgeler yüksek parlaklık ile sunulur.		X		
<b>3 BOYUTLU ESNEK GRAFİK YETENEKLERİ</b>					
	3 eksenli sayısal grafiklerdir. Mühendislik ve test amaçlı olarak sunulur. Eksenlerdeki verinin tip ve alabileceği değer aralıklarını kullanıcı çalışma zamanında seçer. Grafiğin sunacağı veri seti çalışma zamanında seçilebilir. Grafik çizildikten sonra 3 boyutta döndürülebilir, üzerinde zoomlama yapılabilir.				
<b>ANALİZ DESTEK YETENEKLERİ</b>					
Menzil-Doppler Matrisi Görüntüleme Yeteneği	3 boyutlu genlik - hız - menzil grafiğidir. Analiz ve test amaçlı kullanılır, operatörün girdiği yarıçap radarın elde ettiği bulgulardan hesaplanarak çizilir.				
Menzil Verisi Görüntüleme Yeteneği					
Çıkış Gücü Sunum Yeteneği	Anten birimleri çıkış gücünün anlık olarak sunulduğu bir tablodur.				

Tur Raporu Sunum Yeteneđi	Radarin 360 derece tespit yaptığı modda bulunduđu toplam iz sayısı, tespit sayısı, düşen ve yeni bulunan iz sayısı, ortalama genlik vb. bilgilerin sunulduğu bir tablodur. Bakım, test ve analiz amaçlı kullanılır.		X		
<b>VERİ KAYIT VE GERİ OYNATIM YETENEKLERİ</b>					
Ham Veri Kayıt Yeteneđi	Radarı Kullanıcı Arayüzü Yazılımı için bir başlat - durdur butonu anlamına gelir. Tüm radar bulgularının ya da kaba bir filtreden geçirilmiş bulguların özel veri kayıt ortamlarına kaydedilmesidir.				
Sektörel Veri Kayıt Yeteneđi	Kullanıcının girdiđi bir yanca aralıđı ve bir başlat / durdur butonundan oluşur. Yalnız girilen yanca aralıđındaki radar tespitleri özel veri kayıt donanımına aktarılır.				
Plot Veri Kayıt Yeteneđi	Ham veri kayıt yeteneđine benzer. Radarin yalnız plotlara ait veriyi kaydetmesini sağlar.				
Takip Veri Kayıt Yeteneđi	Ham veri kayıt yeteneđine benzer. Radarin yalnız izlere ait veriyi kaydetmesini sağlar.				
Arayüz Veri Kayıt Yeteneđi	Radarı kullanıcı arayüz yazılımının kurduđu tüm bağlantılar için, giden ve gelen verinin bayt olarak kaydedilmesidir. Test, bakım ve hata giderme amaçlıdır.				
Geri Oynatım Yeteneđi	Askeri sahada alınmış mesaj kayıtlarının sonradan gerçekmiş gibi geri oynatılması ve gerçek sahada görülen tespit ve durumların off-line incelenebilmesi yeteneđidir.				
<b>CİHAZ İÇİ TEST YETENEĐİ</b>					
Operatör kontrollü CİT	Bir başlat butonu ve test sonuçları sunum tablosu / tablolarından oluşur. Başlat butonu ile radar kendi kendini test etmeye başlar. Sonuçlar tabloda gösterilir.	X	X	X	
CİT History yeteneđi (OCİT)	CİT sonuç tablolarının sabit diske, sonucun alındığı tarih ile birlikte kaydedilmesidir. Operatöre bir tarih seçim arayüzü sunulur ve seçilen tarihteki test sonuçları gösterilir. Sistemin arıza durumu zamana bađlı olarak analiz edilebilir.				
Sürekli CİT yeteneđi	periyodik cit istek/cevap senaryosu	X	X	X	
<b>TARİH, SAAT VE ZAMAN SENKRONİZASYON YETENEKLERİ</b>					

Temel yetenek	Tarih, saat, zaman dilimi ve DST ayar seçim arayüzüdür.	X	X		
GPS'ten zaman okuma	Dış arayüzden zaman bilgisinin alınması yeteneği				
Zaman Senkronizasyonu	Dış bağlantı ile alınan tarih saat, zaman dilimi ve DST arayüzünün uygulanmasıdır.	X			
<b>YAZILIM YÜKLEME ve BAKIM YETENEKLERİ</b>					
Sürüm Sorgulama ve Sunum Yeteneği	Sistemdeki tüm yazılımların majör ve minör sürüm numaralarının sunumu. Tablolu gösterimdir.	X	X	X	
Yazılım Yükleme Yetenekleri	Sistemdeki tüm yazılımların ilgili birimlere (bilgisayarlar, gömülü kartlar) kopyalanması ve yazılımın güncellenmesidir. Tablolu gösterim ile birlikte ağ üzerinden kopyalama yapılmaktadır.				
Sistem kullanım süreleri sunumu	Toplam ışırma, anten dönüş, KGÜ çalışma süreleri				
Bakım Yetenekleri	Yapılandırma ayarlarının değiştirilmesi, olay günlüğü kayıtlarının çıkarılabilir belleğe aktarımı, özel veri kayıt donanımı dışında kalan diğer veri kayıtlarının aktarımı.	X			
<b>KULLANICI BİLDİRİMLERİ YETENEĞİ</b>					
	Kullanıcıya sunulacak uyarılar ve hatalar kullanıcı mesajları penceresinde görüntülenecektir.	X	X	X	
<b>ANA PENCERE YETENEKLERİ</b>					
Dil Seçimi Yeteneği	Türkçe, İngilizce dil seçim yeteneği	X	X	X	
Tema Seçimi Yeteneği	Öndeğer tema: Siyah, yeşil, mavi Plastic temalar,	X	X	X	
Yazı Tipi Seçimi Yeteneği	Öndeğer yazı tipi ve büyüklüğü.	X	X	X	
Koordinat sistemi seçimi	Koordinat sistemi tipi (Coğrafi, UTM, MGRS)	X	X	X	
Birim Seçimi Yeteneği	Uzaklık (metre, kilometre), Açrı (derece, milyem), datum tipi (WGS-84, ED-50)	X	X	X	
Birim Seçimi Yeteneği	Milyem/saat	X	X	X	
Yardım Yeteneği	Kullanıcı kitabı sunumu, kullanıcı kitabında arama yapılabilmesi	X	X	X	
Hakkında Yeteneği	Kullanıcı arayüz yazılım sürüm numarası, lisans metni, sistem adı ve numarası sunumu	X	X	X	
Hava resminde mesafe gösterimi enable/disable			X	X	
Hava resminde istikamet açısı gösterimi enable/disable			X	X	

<b>DIŞ ARAYÜZ YETENEKLERİ</b>					
Radar Arayüzü		X	X	X	
INS/GPS Arayüzü					
NFS Arayüzü					
CRAM		X			
ERKEN UYARI ALICI-GONDERICI		X			
KKS		X			
Log Sunucu Arayüzü		X	X	X	
<b>YAZILIM İMHA YETENEĞİ</b>					
Yazılım imha	Operatör kontrolü ile şifre girişi ve yazılımın imha edilmesi				
<b>TOOLBAR YETENEKLERİ</b>					
Kısayol bilgileri sunumu		X	X	X	
Anten dönüş durumu sunumu			X	X	
Işıma durumu sunumu		X	X	X	
HAH modu durumu sunumu			X		
IFF otomatik / manuel sorgu durum sunumu			X	X	
Sistem zamanı sunumu		X	X	X	
<b>YAZILIM MOD ve DURUM YETENEKLERİ</b>					
Mühendislik modu	Kısayol ile mühendislik moduna geçiş				
Savaş modu	Kısayol ile şifre doğrulama sonrası savaş moduna geçiş				
Komuta kontrol ile çalışma modu	HERIKKS ile çalışma modu				
Eğitim Modu					
RHOS Modu					
<b>KOMUTA KONTROL YETENEKLERİ</b>					
Komuta Kontrol Mesajlarının sunumu	Gelen komuta kontrol mesajlarının ayrı bir pencere açılarak derhal operatöre sunulması			X	
Onay	Gelen komuta kontrol mesajlarının uygulanması için operatörden onay alınması			X	
Sesli uyarı	KK mesajı geldiğinde sesli uyarı verilmesi				

Uygulama	Gelen komuta kontrol mesajlarının operatör onayı ardından otomatik uygulanması			X	
<b>JENERİK MESAJ YETENEKLERİ</b>					
	Çeşitli mühendislik ihtiyaçlarının karşılanabilmesi için yazılımla byte mesajları ile haberleşme yapmayı sağlayan arayüz				
<b>AÇILIŞ/Bağlantı YETENEKLERİ</b>					
Kaydedilmiş ayarlarla açılış	KGÜ kapatılıp açıldığında en son girilen ayarları saklayarak çalışma özelliği olacaktır.	X	X	X	
Açılışta konum değişikliği işlemlerinin yaptırılması	KGÜ YKB, RSY YKB ile bağlantı kurulduktan sonra operatöre radarın konum değiştirip değiştirmediğini sorar. Konum değiştiyse sırasıyla konum girişi, kuzey kalibrasyonu, arazi silüeti belirleme yaptırılır.				
<b>ARŞİVLEME</b>					
Tüm kontrol mesajlarının lokal/harici disklere kayıt edilmesi					

#### **EK-4: Anket Çalışması**

##### **Özellik Ağacı Metodu İle Yazılımın Yeniden Kullanılabilirliğinin Arttırılması**

Gerçekleştirilen Sistematik Literatür Taraması sonucunda yazılımların yeniden kullanılabilirliğini değerlendirmek için kullanılan en belirleyici yazılım faktörlerinin bağlaşım (İng. Coupling), uyumluluk (İng. Cohesion) ve karmaşıklık (İng. Complexity) olduğu sonucuna varılmıştır. Literatürde yazılım faktörlerini sayısallaştırmak için sınıf bazında ölçüm yapan Metotlardaki Uyum Eksikliği (İng. Lack of Cohesion in Methods – LCOM), Sınıfın Ağırlıklı Metot Sayısı (İng. Weigted Methods Per Class – WMC) ve Çevrimsel Karmaşıklık (İng. Cyclomatic Complexity) ile modül bazında ölçüm yapan Ortalama Çevrimsel Bağımlılık (İng. Avarage Cycling Dependency – ACD) metrikleri kullanılmaktadır. Belirlenen bu metrik seti ile ortak yazılım ürün hattı ailesinden gelen OSGi çerçevesinde Java programlama dili kullanılarak geliştirilmiş projelerin ölçümleri alınmıştır. Projelerin her birinin Yazılım Gereksinim Özelliği dokümanları kullanılarak Özellik Ağacı oluşturulmuştur. Yetenek setleri çıkarılırken ekran gözden geçirmeye (İng. peer review) başvurulmuştur. Ölçüm sonuçlarına göre yazılım Özellik Ağacı metoduna göre yazılımlar yeniden yapılandırılmıştır. Yeniden yapılandırılmış yazılım projelerinin belirtilen metrik setleri kullanılarak ölçümleri alınmıştır. Yapılandırma öncesi ve sonrasında alınan ölçümlerin sonuçlarına göre yeniden kullanılabilirliğin arttığı gözlemlenmiştir. Bu anket çalışması, OSGi çerçevesi, Java program dili kullanılarak geliştirilen aynı yazılım ürün hattı ailesinden gelen yazılımların yeniden kullanılabilirliğini arttırmak için önerilen Özellik Ağacı metodunun kullanılabilirliğini (İng. usability) ve kullanılabilirlikteki kolaylığını (İng. ease of use) araştırmayı amaçlamaktadır.

#### **Yazılım projelerindeki pozisyonunuz nedir? \***

- Yazılım geliştirici / programcı
- Yazılım Mimari
- Yazılım Testçisi
- Proje Yöneticisi
- Other:

#### **Yazılım geliştirirken OSGi çerçevesini ne sıklıkta kullanıyorsunuz? \***

- Her zaman
- Bazen
- Nadir
- Hiçbir zaman
- Other:



**Yazılımın geliřtirmenin hangi fazında yeniden kullanım yapıyorsunuz?\***

- Gereksinim Belirleme Fazı
- Tasarım Fazı
- Uygulama/ Kodlama Fazı
- Test ve Entegrasyon Fazı
- Bakım Fazı
- Other:

**Yazılım geliřtirirken bařvurduđunuz yeniden kullanım teknikleri nelerdir? \***

- Tasarım Örüntüleri (İng. Desing Patterns)
- Çerçeveseler (İng. Frameworks)
- API ve Kütüphaneler (API & Libraries)
- Üretici Programlama (İng. Generative Programming)
- Model Güdümlü Geliřtirme (İng. Model Driven Development )
- Other:

**Yazılım geliřtirirken yeniden kullanım tekniklerine hangi sıklıkta bařvuruyorsunuz? \***

- Her zaman
- Bazen
- Nadiren
- Hiçbir zaman
- Other:

**Özellik ađacı metodu ile yeniden kullanılabilirliđi artırma yönteminin uygulanabilirliđi nedir? \***

- Uygulanabilirliđi yüksek
- Uygulanabilirliđi düşük
- Uygulanamaz
- Other:

**Özellik ađacı metodu ile yeniden kullanılabilirliđi artırma yöntemini uygulamanın kolaylık derecesi nedir? \***

- Çok kolay
- Kolay
- Zor
- Other:

**Özellik ađacı metodunu uygulamayı düşünür müsünüz? \***

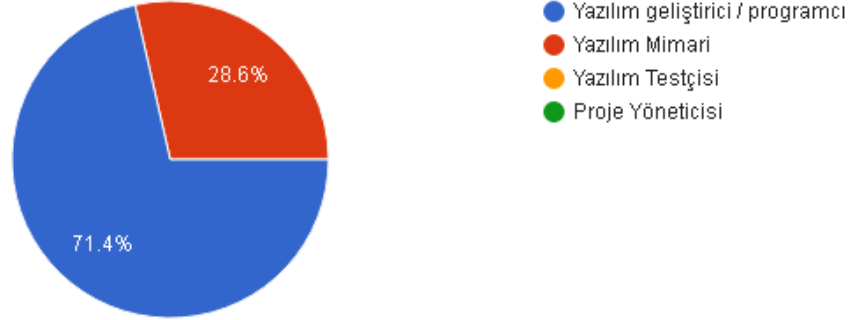
- Evet
- Hayır

**Cevabınız hayırsa öngördüğünüz engeller nelerdir?**

## EK -5 Anket Sonuçları

### Yazılım projelerindeki pozisyonunuz nedir?

7 responses

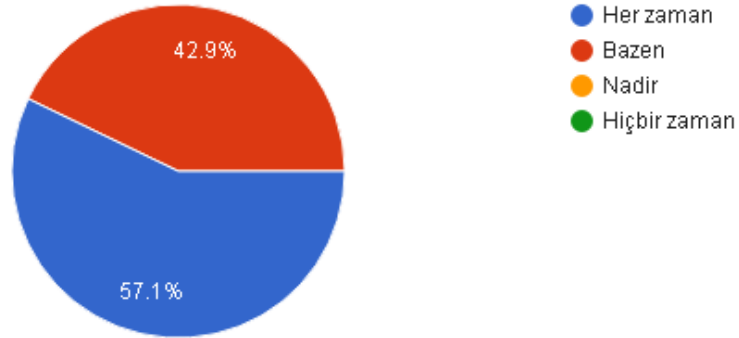


Yazılım Geliştirici:5

Yazılım Mimarı: 2

### Yazılım geliştirirken OSGi çerçevesini ne sıklıkta kullanıyorsunuz?

7 responses

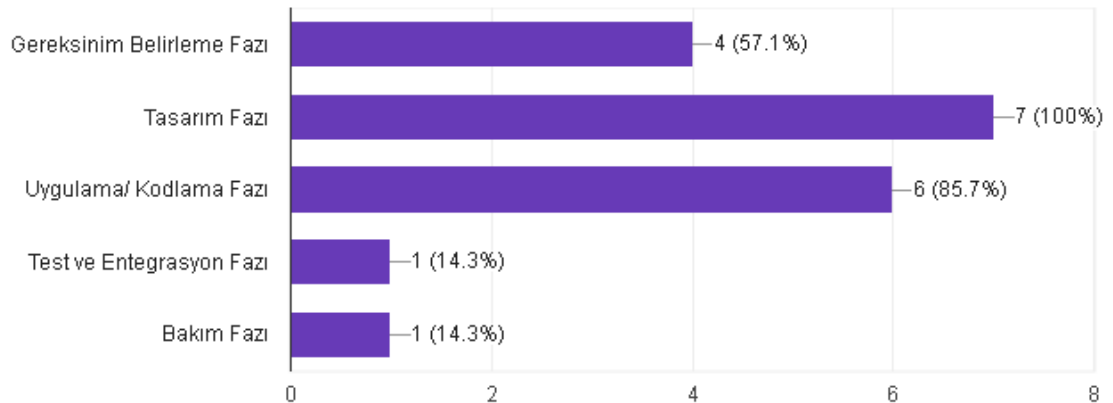


Her Zaman: 4

Bazen: 3

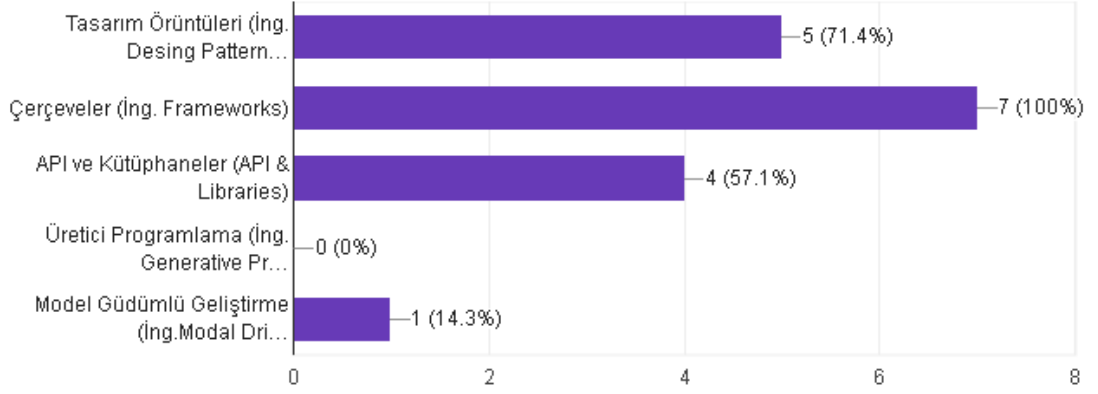
### Yazılımın geliştirmenin hangi fazında yeniden kullanım yapıyorsunuz?

7 responses



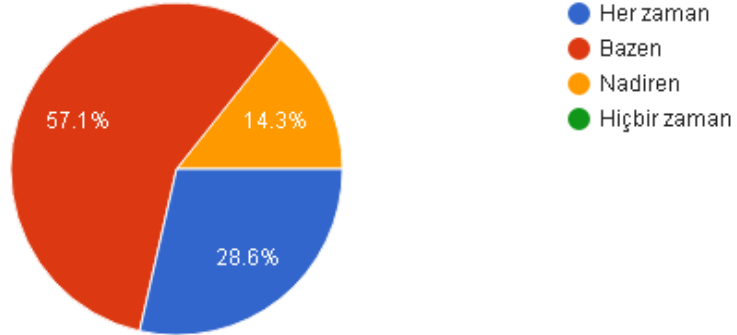
## Yazılım geliştirirken başvurduğunuz yeniden kullanım teknikleri nelerdir?

7 responses



## Yazılım geliştirirken yeniden kullanım tekniklerine hangi sıklıkta başvuruyorsunuz?

7 responses



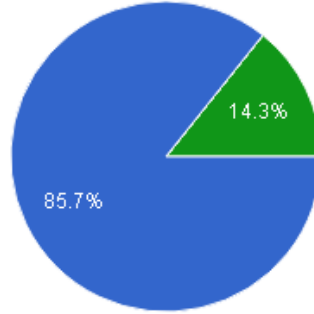
Her Zaman:2

Bazen: 4

Nadiren: 1

## Özellik ağacı metodu ile yeniden kullanılabilirliği artırma yönteminin uygulanabilirliği nedir?

7 responses

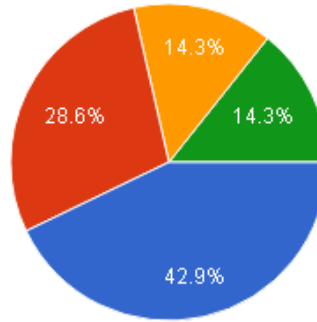


- Uygulanabilirliği yüksek
- Uygulanabilirliği düşük
- Uygulanamaz
- Hangi alana uygulandığından bağımsız olarak uygulanabilir bir yöntemdir. Yöntemin genel uygulama adımları açıktır. Her adımın kendi içinde nasıl uygulanacağı daha detaylandırılabilir. Uygulamada güçlüklerle karşılaşılabirse de gen...

Uygulanabilirliği Yüksek: 6  
Diğer: 1

## Özellik ağacı metodu ile yeniden kullanılabilirliği artırma yöntemini uygulamanın kolaylık derecesi nedir?

7 responses

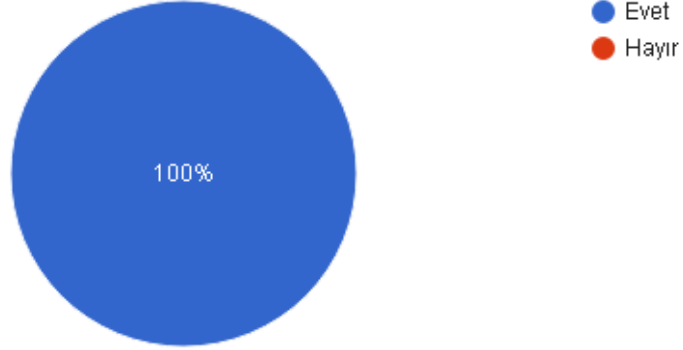


- Çok kolay
- Kolay
- Zor
- Daha önceden özellik ağacı çıkartılmamış projelerde, özellik ağacının oluşturulması zaman alıcı olabiliyor. Bu sebeple geçmişe yönelik projelerde uygulanması zor olabiliyor.

Çok Kolay: 4  
Kolay: 2  
Zor: 1  
Diğer:1

## Özellik ağacı metodunu uygulamayı düşünür müsünüz?

7 responses



## Cevabınız hayırsa öngördüğünüz engeller nelerdir?

0 responses

**No responses yet for this question.**

## KAYNAKLAR

- [1] Seungwon, L., Choi, H., Software Component Reusability Measure In Component Grid, 11th International Conference on Advanced Communication Technology, PyeongChang, South Korea, Vol. 1, 576–578, 2009.
- [2] Cankurtaran, E., Cilden, E., Tarhan, A., Bileşen Tabanlı ve Ürün Hattı Yazılım Geliştirme Yaklaşımlarında Yeniden Kullanılabilirlik Metrikleri, UYMS'16, 2016.
- [3] Cankurtaran, E., Cilden, E., Tarhan, A., OSGi Çerçevesinde ve Ürün Hattı Yaklaşımı ile Geliştirilmiş Bir Radar Kullanıcı Arayüzü Yazılımının Özellik Ağacı ile Yeniden Kullanılabilirliğinin Arttırılması, UYMS'17, 2017.
- [4] Frakes, W., Succi, G., An Industrial Study of Reuse, Quality and Productivity, Journal of Systems and Software, Vol:57, 99-106, 2001.
- [5] Rotaru, O., Dobre, M., Reusability Metrics for Software Components, the 3rd ACS/IEEE International Conference on Computer Systems and Applications, Cairo, Egypt, 2005.
- [6] Parmeze, D., Fifo, M., A Survey of Methods for Measuring and Enhancing Component Reusability, J. Converg. Information Technology, Vol. 4, 2009.
- [7] Torkamani, M., Metric Suite to Evaluate Reusability of Software Product Line, International Journal of Electrical and Computer Engineering (IJECE), Vol:4 No:4, pp: 285-294, 2014.
- [8] Kriens, P., OSGi Introductory Tutorials, Vol:2, 58-165, 2006.
- [9] Schmid, K., Santana, E., Product Line Engineering, IEEE Journal and Magazines, Vol:30, 24-30, 2013.
- [10] Kahraman, E., İpek, T., İyidir, B., Bazlamaççı, C., Bilgen, S., Bileşen Tabanlı Yazılım Ürün Hattı Geliştirmeye Yönelik Alan Mühendisliği Çalışmaları, UYMS'09, 2009.
- [11] Karata, E., İyidir, B., Yazılım Ürün Hattı Yaklaşımında Model GÜdümlü Uygulama Mühendisliği, UYMS'09, 2009.
- [12] Metzger, A., Pohl, K., Software Product Line Engineering and Variability Management: Achievements And Challenges, ICSE Future of Software Engineering Track, FOSE 2014, 2014.
- [13] Sandhu, P., Singh, H., Automatic Reusability Appraisal of Software Components Using Neuro- Fuzzy Approach, International Journal Information Technology, Vol. 1, No. 8, 2407–2413, 2014.
- [14] Badareen, A., Selamat, M., Jabar, Din, M., J, Turaev, S., Reusability Software Component Life Cycle, International Journal of Computers, Vol:5, 191-200, 2011.
- [15] Solingen, R., Basili, V., Caldiera, G., Rombach, D., Goal Question Metric Approach, 1994.
- [16] ISO9126 - *Software Quality Characteristics*. URL: <http://www.sqa.net/iso9126.html>. [Son ziyaret tarihi: **13-Mayıs-2018**].
- [17] Goodman, P., *The Practical Implementation of Software Metrics*, McGraw-Hill, **1993**.
- [18] Schulmeyer, G., *Handbook of Software Quality Assurance*. Prentice Hall, **1987**.
- [19] Chidamber, S.R., Kemerer, C.F., *A Metrics Suite for Object Oriented Design*, Vol:20, Issue:6, **1994**.

- [20] Hanefi Calp M., Arici, N., *Nesne Yönelimli Tasarım Metrikleri ve Kalite Özellikleri İlişkisi*, *Politek. Derg. J. Polytech. Cilt Digit. Object Identifier*, Vol. 14141, no. 10, 9–14, **2011**.
- [21] Murgia, A., Tonelli, R., Concas, G., Marchesi, M., Counsell, S. , *Parameter-Based Refactoring and the Relationship with Fan-in Fan-out Coupling*, *Journal of Object Technology*, Vol:11, No:2, **2012**.
- [22] Sonargraph URL: <https://www.hello2morrow.com/products/sonargraph>. [Son ziyaret tarihi: **21-Mayıs-2018**].
- [23] Understand URL: <https://scitools.com/>. [Son ziyaret tarihi: **21-Mayıs-2018**].
- [24] Boshnakoska, D., *Interpretation of Object Oriented Quality Models*, The 7th International Conference for Informatics and Information Technology, **2010**.
- [25] Kitchenham, B., *Procedures for Performing Systematic Reviews*, Eversleigh, **2004**.
- [26] Basili, V. R., Caldiera, G., Rombach, H.D., *The Goal Question Metric Approach*, *Encyclopedia of Software Engineering*, **1994**.
- [27] Sandhu, P., Singh, H., *A Neuro-Fuzzy Based Software Reusability Evaluation System with Optimized Rule Selection*, International Conference on Emerging Technologies (ICET'06), 664–669, **2006**.
- [28] Torkamani, M. A., *Metric Suite to Evaluate Reusability of Software Product Line*, *International Journal of Electrical and Computer Engineering (IJECE)*, Vol:4 No:4, pp: 285-294, **2014**.
- [29] Wang, J., Wang, J., *Teaching software reuse with JavaBeans*, 30th Annual Frontiers in Education Conference, Kansas City, USA, Vol. 1, 7–8, **2000**.
- [30] Zhang, T., Deng, L. , Wu, J., Zhou, Q., Ma, C. *Some Metrics for Assessing Quality of Product Line Architecture*, International Conference on Computer Science and Software Engineering, **2008**.
- [31] Washizaki, H., Namiki, R., Fukuoka, T., Harada, Y., Watanabe, H., *A Framework For Measuring and Evaluating Program Source Code Quality*, *Product-Focused Software Process Improvement*, Vol. 4589, 284–299, **2007**.
- [32] Yu, L.G., Ramaswamy, S., *Component Dependency in OO Software*, *Journal Computer Science and Technology*, Vol:22, 379-386, **2007**.
- [33] Sandhu, P. S. , Dalwinder, S. S., Singh, H., *A Comparative Analysis of Fuzzy, Neuro-Fuzzy and Fuzzy-GA Based Approaches for Software Reusability Evaluation*, *Proceedings of World Academy of Science: Engineering & Technology*, **2008**.
- [34] Washizaki, H., Namiki, R. , Fukuoka, T., Harada, Y., Watanabe, H., *A Framework For Measuring and Evaluating Program Source Code Quality*, *Product-Focused Software Process Improvement*, Vol. 4589, 284–299, **2007**.
- [35] Sandhu, P. S., Singh, H., *A Fuzzy-Inference System Based Approach for the Prediction of Quality of Reusable Software Components*, International Conference on Advanced Computing and Communications, 349–352, **2006**.
- [36] Dandashi,F., *A Method for Assessing the Reusability of Object-Oriented Code Using a Validated Set of Automated Measurements*, **2002**.
- [37] Venkatesan, V. P., Krishnamoorthy, M., *A Metrics Suite for Measuring Software Components*, *J. Conver. Information Technology*, Vol. 4, **2009**.
- [38] Sandhu, P. Singh, H., *A Neuro-Fuzzy Based Software Reusability Evaluation System with Optimized Rule Selection*, International Conference on Emerging Technologies (ICET'06), 664–669, **2006**.

- [39] Sandhu, P. S., Singh, H., *A Reusability Evaluation Model for OO-Based Software Components*, International Journal Computer Science, Vol. 1, 259–264, **2006**.
- [40] Nerurkar, N. W., Sharma, A., Sagar, S., *A Soft Computing Based Approach To Estimate Reusability of Software Components*, ACM SIGSOFT Software Engineering Notes, Vol. 35, **2010**.
- [41] Zahara, S. I. , Ilias, M., Zia, T., *A Study of Comparative Analysis of Regression Algorithms for Reusability Evaluation of Object Oriented Based Software Components*, International Conference on Open Source Systems and Technologies (ICOSST'13), Pakistan, **2013**.
- [42] Etzkorn, L. H. , Hughes, W. E., Davis, C. G., *Automated reusability quality analysis of OO legacy software*, Information Software Technology, Vol. 43, No. 5, 295–308, (2001).
- [43] Sandhu P., Singh, H., *Automatic Reusability Appraisal of Software Components Using Neuro- Fuzzy Approach*, International Journal Information Technology, Vol. 1, No. 8, 2407–2413, **2007**.
- [44] Ko, B., Park, J., *Component Architecture Redesigning Approach Using Component Metrics in Artificial Intelligence and Simulation*, Heidelberg: Springer Berlin Heidelberg Vol. 3397, 449–459, **2005**.
- [45] Cho, E. S., Kim, M. S., Kim, D., *Component Metrics to Measure Component Quality*, Eighth Asia-Pacific Software Engineering Conference, Macau, China, pp. 419–426, **2001**.
- [46] Bhattacharya, S., Perry, D.A., *Contextual Reusability Metrics for Event-Based Architectures*, International Symposium on Empirical Software Engineering, Australia, (2005).
- [47] Choi, M., Lee, S., *A Coupling Metric Applying the Characteristics of Components*, Computational Science and Its Applications, Vol. 3983, Berlin, Heidelberg: Springer Berlin Heidelberg, 966–975, **2006**.
- [48] Singh, A. P., Tomar, P. , *Estimation of Component Reusability through Reusability Metrics*, World Academy of Science Engineering and Technology International Journal of Computer, Electrical Automation Control and Information Engineering, Vol:8, No: 11, 2003- 2010, **2014**.
- [49] La, H., Her, S., Kim, S., *Framework for evaluating reusability of Component-as-a-Service (CaaS)*, ICSE Workshop on Principles of Engineering Service-Oriented Systems (PESOS), San Francisco, USA, 41–44, **2013**.
- [50] Manhas, S., Sandhu, P.S., Chopra, V., Neeru, N., *Identification of reusable software modules in function oriented software systems using neural network based technique*, World Acad. Sci. Eng. Technol., Vol. 43, 823–827, **2010**.
- [51] Boxall, M.A.S. , Araban, S., *Interface Metrics for Reusability Analysis of Components*, Australian SE Conference Proceedings., Melbourne, Australia, pp. 40–51, **2004**.
- [52] Gui, G., Scott, P. D., *Measuring Software Component Reusability By Coupling And Cohesion Metrics*, J. Comput., Vol: 4, No: 9, **2009**.
- [53] Sandhu, P. S., Kaur, H., *Modeling of Reusability of Object Oriented Software System*, World Acad. Sci. Eng. Technol., Vol. 56, 162–165, **2009**.
- [54] Yingmei, L., Jingbo, S., Weining, X., *On Reusability Metric Model for Software Component” in Software Engineering and Knowledge Engineering: Theory and Practice*, Vol. 114, Springer Berlin Heidelberg, 865–870, **2012**.



- [55] Sant'Anna, C., Garcia, A., Chavez, C., Lucena, C., Staa, A., *On the Reuse and Maintenance of Aspect- Oriented Software: An Assessment Framework*, in Proceedings of Brazilian Symposium on Software Engineering, **2003**.
- [56] Gupta, V., Kumar, J., *Package Level Cohesion Measurement in Object-Oriented Software*, J. Braz. Comput. Soc., Vol. 18, No. 3, 251–266, Sep. **2012**.
- [57] Shri, A., Sandhu, P . S., Gupta, V., Anand, S., *Prediction of Reusability of Object Oriented Software Systems Using Clustering Approach*, World Acad. Sci. Eng. Technol., Vol. 43, 853–856, **2010**.
- [58] Price, M.W., Needham, D.M., Demurjian, S.A., *Producing Reusable Object-Oriented Components: A Domain-And-Organization- Specific Perspective*, Symposium on Software Reusability, Toronto, Canada, 41–50, **2001**.
- [59] Wang, Q., *Research on the Quantifying and Calculating Model of the Software Component Reusability*, Future Control and Automation Heidelberg: Springer Berlin Heidelberg, 103–109, **2012**.
- [60] Rotaru, O.P., Dobre, M., *Reusability Metrics for Software Components*, the 3rd ACS/IEEE International Conference on Computer Systems and Applications, Cairo, Egypt, **2005**.
- [61] Lee, S., Choi, S., *Software Component Reusability Measure In Component Grid*, 11th International Conference on Advanced Communication Technology, PyeongChang, South Korea, Vol. 1, 576–578, **2009**.
- [62] Dallal, J., *Software Similarity-Based Functional Cohesion Metric*, IET Softw., Vol. 3, No. 1, 46–57, **2009**.
- [63] Hristov, D., Hummel, O., Huq, M., Janjic, W., *Structuring Software Reusability Metrics for Component-Based Software Development*, ICSEA, 422-429, **2012**.
- [64] Wang, J., Wang, Y., *Teaching software reuse with JavaBeans*, 30th Annual Frontiers in Education Conference, Kansas City, USA, Vol. 1, pp. 7–8, **2000**.
- [65] Korra, S., Viswanadha, S., Babu, *Strategies for Designing and Building Reusable Software Components*, Sampath Korra et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4, pp: 655-659, **2013**.
- [66] Poulin, J., *Measuring Software Reusability*, **ICSR, (1994)**.
- [67] Singh, C., Pratap, A., Singhal, A., *Estimation of Software Reusability for Component based System using Soft Computing Techniques*, 5th International Conference- Confluence The Next Generation Information Technology Summit, At Amity University Noida, **2014**.
- [68] Sandhu, P., Singh, H., *Automatic Reusability Appraisal of Software Components Using Neuro- Fuzzy Approach*, International Journal Information Technology, Vol. 1, No. 8, 2407–2413, **2007**.
- [69] Schmidt, D. C., *Using Design Patterns to Develop Reusable OO Communication Software*, Communication of the ACM, Vol:56, **1995**.
- [70] Nerurkar, N. W., Sharma, A., Sagar, S., *A soft computing based approach to estimate reusability of software components*, ACM SIGSOFT Softw. Eng. Notes, Vol: 35,No: 4, 1, **2010**.
- [71] Kakkar, P., Sharma, M., Sandhu, P., *Modeling of reusability of software systems*, **2010**.
- [72] Younoussi, S., Roudies, O., *All About Software Reusability: A Systematic Literature Review*, Journal of Theoretical and Applied Information Technology, Vol:76, **2015**.

- [73] Babu, G.N.K. S., Srivatsa, S.K., *Analysis and Measures of Software Resability*, International Journal of Reviews in Computing, **2009**.
- [74] Caldiera, G., Basili, V., *Identifying and Qualifying Reusable Software Components*, Vol:24, Issue:2, 61-70, **1991**.
- [75] Torkamani, M.A., *Metric Suite to Evaluate Resusability of Software Product Line*, Vol:4, Issue:2, **2014**.
- [76] ISO9126 - Software Quality Characteristics. URL: <http://www.sqa.net/iso9126.html>. [Son ziyaret tarihi: **13-Mayıs-2018**].
- [77] SEI, *Security Requirement Reusability and the SQUARE Methodology*, **2010**.
- [78] Fraunhofer Institute for Experimental Software Engineering IESE, *Implementation Issue in Product Line Scoping*, Vol:6, pp:170-189, **2000**.
- [79] Washizaki, H., Yamamoto, H., Fukazawa, Y., *A metric süite for measuring reusability of software components*, Vol: 10, **2003**.
- [80] Atkinson, C., Muthig, D., *Enhancing Component Resability through Product Line Technology*, Vol:3, **2002**.
- [81] Ommering, R. V., *Software Reuse in Product Population*, Vol:31, Issue: 7, **2005**.
- [82] Yin, R.K., *Case Study Resarch Design and Methods*, 4th Edition, Vol:5
- [83] Hamza, M., Walker, R., *Recommending Features and Feature Relationships from Requirements Documents for Software Product Line*, *2015 IEEE/ACM 4th International Workshop on Realizing AI Synergies in Software Engineering*, **2015**.
- [84] Wang, Y., *Semantic Information Extraction for Software Requirements using Semantic Role Labeling*, *2015 IEEE*, **2015**.
- [85] Bakar, N., Kasirun, Z., Salleh, N., Jalab, H., *Extracting features from online software reviews to aid requirements*, *Elsevier*, **2016**.
- [86] Li, Y., Guzman, E., Tsiamura, K., *Automated requirements extraction for scientific software*, *ICCS 2015 International Conference On Computational Science*, **2015**.
- [87] Anlamlılık Analizi, <https://help.surveymonkey.com/articles/tr/kb/Significant-Differences>, [Son ziyaret tarihi: **25-Temmuz-2018**].

## ÖZGEÇMİŞ

### Kimlik Bilgileri

Adı Soyadı : Ezgi Cankurtaran

Doğum Yeri : Ankara

Medeni Hali : Bekar

E-posta : [ecankurtaran@aselsan.com.tr](mailto:ecankurtaran@aselsan.com.tr), [cankurtaranezgi@gmail.com](mailto:cankurtaranezgi@gmail.com)

Adresi : 381. Sok. 2/7 Kırkkonaklar Mah. Çankaya Ankara/ Türkiye

### Eğitim

Lise: Çankaya Milli Piyango Anadolu Lisesi, Ankara, (2004-2008)

Genel Ortalama: 90.88 / 100

Lisans: Gazi Üniversitesi Bilgisayar Mühendisliği, Ankara, (2009-2013)

Genel Not Ortalaması: 3.71 / 4.00 (Bölüm Birinciliği)

Lisans Çift Anadal: Gazi Üniversitesi Endüstri Mühendisliği, Ankara, (2010-2014)

Genel Not Ortalaması: 3.08 / 4.00

Yüksek Lisans: Hacettepe Üniversitesi Bilgisayar Mühendisliği, Ankara, (2014-2018)

Genel Not Ortalaması: 3.82 / 4.00

### Yabancı Dil Düzeyi

Hacettepe Yeterlilik Sınavı: 92

YDS: 73.75

Upper Intermediate Level Certificate, American Exam Center, (2013)

### İş Deneyimi

Bir Sistem Geliştirme ve Entegrasyon Firması, Yazılım Müdürlüğü,  
Kullanıcı Arayüzü Ekibi, Ankara, (2014 - Halen)

GAZI UNIVERSITY TECHNOLOGY AND INNOVATION CENTER  
([www.gutic.org](http://www.gutic.org)),

Biyometri Laboratuvar Asistanlığı, (2011-2013)

ISA INFORMATION SECURITY ASSOCIATION

Bilgi Güvenliği Uzmanlığı, (2011- )

TAI

Üretim Planlama Departmanı, İş Planlama Birimi, Ankara, (2 Ay, 2013)

TURKCELL TEKNOLOJİ

PAF Takımı, Oracle Departmanı, Kocaeli, (3 Ay, 2012)

MICROSOFT

Yaz Okulu, Ankara, (1 Ay, 2012)

ASELSAN

Kriptoloji Müdürlüğü,

ASELSAN 2110 KCT Kriptolu Telefonun yönetimi için arayüz tasarımı, Ankara, (2 Ay, 2012)

TAI

Yazılım Destek Departmanı,

Cisco NAC çözümünün TAI'de entegrasyonu, Ankara, (2 Ay, 2011)

### **Deneyim Alanları**

Yazılım Mühendisliği

### **Tezden Üretilmiş Projeler ve Bütçesi**

-

### **Tezden Üretilmiş Yayınlar**

Cankurtaran E., Çilden E., Tarhan A., *OSGi Çerçevesinde ve Ürün Hattı Yaklaşımı ile Geliştirilmiş Bir Radar Kullanıcı Arayüzü Yazılımının Özellik Ağacı İle Yeniden Kullanılabilirliğinin Artırılması*, UYMS'17, Alanya, 2017.

Cankurtaran E., Çilden E., Tarhan A., *Bileşen Tabanlı ve Ürün Hattı Yazılım Geliştirme Yaklaşımlarında Yeniden Kullanılabilirlik Metrikleri*, UYMS'16, Çanakkale, 2016.

### **Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu ile Katıldığı Toplantılar**

-



HACETTEPE ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
YÜKSEK LİSANS/DOKTORA TEZ ÇALIŞMASI ORJİNALLİK RAPORU

HACETTEPE ÜNİVERSİTESİ  
FEN BİLİMLER ENSTİTÜSÜ  
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI BAŞKANLIĞI'NA

Tarih: 13/09/2018

Tez Başlığı / Konusu: **Yazılım Ürün Hattındaki Modüllerin Özellik Ağacı Analizi ile Yeniden Kullanılabilirliğinin Arttırılması: OSGi Çerçevesi Radar Kullanıcı Arayüzü Örneği**

Yukarıda başlığı/konusu gösterilen tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler d) Sonuç kısımlarından oluşan toplam 93 sayfalık kısmına ilişkin, 12/09/2018 tarihinde ~~gözetim~~/tez danışmanım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 5'tir.

Uygulanan filtrelemeler:

- 1- Kaynakça hariç
- 2- Alıntılar hariç
- 3- 5 kelimedenden daha az örtüşme içeren metin kısımları hariç

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygılarımla arz ederim.

Tarih ve İmza

Adı Soyadı: Ezgi CANKURTARAN  
Öğrenci No: N14124585  
Anabilim Dalı: Bilgisayar Mühendisliği  
Programı: Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı  
Statüsü:  Y.Lisans  Doktora  Bütünleşik Dr.

13.09.2018  
*Ezgi Cankurtaran*

**DANIŞMAN ONAYI**

UYGUNDUR.

*Ayça Tarhan*  
Dr. Öğretim Üyesi AYÇA TARHAN

(Unvan, Ad Soyad, İmza)