

**AÇIK KAYNAK YAZILIMLARDA BAKIM YAPILABİLİRLİĞİ
VE GÜVENİLİRLİĞİ ÖLÇMEK İÇİN İKİ BOYUTLU
DEĞERLENDİRME METODU**

**A TWO-DIMENSIONAL METHOD FOR EVALUATING
MAINTAINABILITY AND RELIABILITY OF OPEN SOURCE
SOFTWARE**

NEBİ YILMAZ

YRD. DOÇ. DR. AYÇA TARHAN
Tez Danışmanı

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

Temmuz 2017

NEBİ YILMAZ'ın hazırladığı “**Açık Kaynak Yazılımlarda Bakım Yapılabilirlik ve Güvenilirliği Ölçmek için İki Boyutlu Değerlendirme Metodu**” adlı bu çalışma aşağıdaki jüri tarafından **BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**' nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Prof. Dr. Ali H. DOĞRU
Başkan



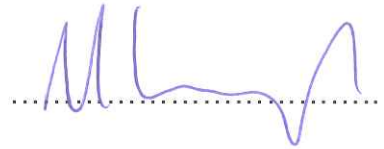
Yrd. Doç. Dr. Ayça TARHAN
Danışman



Doç. Dr. Altan KOÇYİĞİT
Üye



Yrd. Doç. Dr. Mehmet KÖSEOĞLU
Üye



Yrd. Doç. Dr. Oumout CHOUSEINOGLU
Üye



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

Prof. Dr. Menemşe GÜMÜŞDERELİOĞLU
Fen Bilimleri Enstitüsü Müdürü

YAYINLAMA 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 kullanması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

- Tezimin/Raporumun tamamı dünya çapında erişime açılabilir ve bir kısmı veya tamamının fotokopisi alınabilir.

(Bu seçenekle teziniz arama motorlarında indekslenebilecek, daha sonra tezinizin erişim statüsünün değiştirilmesini talep etmeniz ve kütüphane bu talebinizi yerine getirirse bile, tezinin arama motorlarının önbelleklerinde kalmaya devam edebilecektir.)

- Tezimin/Raporumun 01/04/18 tarihine kadar erişime açılmasını ve fotokopi alınmasını (İç Kapak, Özet, İçindekiler ve Kaynakça hariç) istemiyorum.

(Bu sürenin sonunda uzatma için başvuruda bulunmadığım takdirde, tezimin/raporumun tamamı her yerden erişime açılabilir, kaynak gösterilmek şartıyla bir kısmı ve ya tamamının fotokopisi alınabilir)

- Tezimin/Raporumun tarihine kadar erişime açılmasını istemiyorum, ancak kaynak gösterilmek şartıyla bir kısmı veya tamamının fotokopisinin alınmasını onaylıyorum.

- Serbest Seçenek/Yazarın Seçimi

26 / 07 / 2017

Y. Yılmaz
(İmza)

Öğrencinin Adı Soyadı

Nebi YILMAZ

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.

26/07/2017

NEBİ YILMAZ

ÖZET

AÇIK KAYNAK YAZILIMLARDA BAKIM YAPILABİLİRLİĞİ VE GÜVENİLİRLİĞİ ÖLÇMEK İÇİN İKİ BOYUTLU DEĞERLENDİRME METODU

Nebi YILMAZ

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Danışmanı: Yrd. Doç. Dr. Ayça TARHAN

Temmuz 2017, 134 sayfa

Son yıllarda açık kaynak yazılımların (AKY) popülerliğinin artması, birbirine alternatif olarak pazara sunulan bu tür yazılımların sayısının hızla artmasına sebep olmuştur. Ancak açık kaynak yazılımlarının kalitesinin potansiyel kullanıcılar tarafından nasıl değerlendirilebileceğine ışık tutan akademik çalışmalar sınırlı sayıdadır. Dolayısıyla alternatifler arasından ihtiyaç sahiplerinin kalite gereksinimlerini en iyi karşılayanın seçilmesi problemi araştırılması gereken cazip bir problem haline gelmiştir. Bu tez çalışmasında literatürde mevcut çalışmalar sentezlenerek kendi katkılarımızla beraber açık kaynak yazılımlarda bakım yapılabilirliği ve güvenilirliği ölçen bir metot geliştirilmiştir. Bu ölçüm için açık kaynak yazılımlar hem kod-tabanlı hem de toplum-tabanlı olarak iki boyutlu bir metotla değerlendirilmektedir. Kod-tabanlı ölçüm yapabilmek için, en güncel kalite modeli olan ISO/IEC 25010'nun seçilen içsel öznitelikleri ve bu içsel öznitelikleri ölçmek için nesneye yönelik C&K metrikleri kullanılmıştır. Toplum-tabanlı ölçüm yapabilmek için ise sistem ve yazılım mühendisliği için ölçüm süreci standardı olan ISO/IEC 15939 rehber alınarak bilgi ihtiyaçları belirlenmiş ve bu bilgi ihtiyaçlarını değerlendirmek için ürünlerin veri tabanlarında depolanan elektronik posta listeleri, problem (hata) raporları, sıkça sorulan sorular vb. tarihsel verilerden türetilen metrikler kullanılmıştır. Bu tez çalışmasında önerilen iki-boyutlu bu metot, Java dilinde yazılmış üç Java programı

inşa aracının (İng. Java built tool) seçiminde kullanılmış ve elde edilen sonuçlar tartışılmıştır.

Anahtar Kelimeler: Açık kaynak yazılım, AKY, yazılım değerlendirme, yazılım seçme, yazılım kalitesi, bakım yapılabilirlik, güvenilirlik, ISO/IEC 25010, ISO/IEC 15939, yazılım metrikleri.

ABSTRACT

A TWO-DIMENSIONAL METHOD FOR EVALUATING MAINTAINABILITY AND RELIABILITY OF OPEN SOURCE SOFTWARE

Nebi YILMAZ

Master of Science, Computer Engineering Department

Supervisor: Yrd. Doç. Dr. Ayça TARHAN

July 2017, 134 pages

Increased popularity of open source software (OSS) has led to a considerable proliferation of alternative software. However, this being the case, an evident lack of studies that would contribute to evaluation of OSS by organizations has turned the process of selecting the most suitable product into an appealing research problem. In this study, a method to evaluate reliability and maintainability of OSS products by using both code-based and community-based aspects have been obtained from the synthesis of existing studies in the literature and with our contribution. In order to perform code-based evaluation, some internal attributes of the most recently quality model, ISO/IEC 25010, have been selected and object-oriented C&K metrics have been employed in an attempt to measure these attributes. To perform community-based evaluation, metrics derived from historical data such as e-mailing lists, program reports, frequently asked questions, and etc. have been utilized to identify and satisfy information needs as conformant to ISO/IEC 15939 standard for software measurement process. The proposed method has been used to evaluate the maintainability and reliability of three Java program build tools written in Java, and results of evaluation have been presented and discussed.

Keywords: Open source software, OSS, software evaluation, software selection, software quality, maintainability, reliability, ISO/IEC 25010, ISO/IEC 15939, software metrics.

TEŐEKKÜR

Öncelikler bu tez alıřmam sırasında kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösteren ve beni her zaman güler yüz ile karşılayan değerli danışman hocam Yrd. Do. Dr. Aya TARHAN'a sonsuz teőekkür ve saygılarımı sunarım. Ayrıca değerli tez jüri üyeleri Prof. Dr. Ali H. DOĐRU, Do. Dr. Altan KOYİĐİT, Yrd. Do. Dr. Oumout CHOUSEINOGLU ve Yrd. Do. Dr. Mehmet KÖSEOĐLU'na teőekkürü bir bor bilirim.

Yüksek lisans eđitimim boyunca bilgi ve tecrübeleriyle bana destek olan başta değerli hocam Yrd. Do. Dr. Kıvan DİNER olmak üzere bölümümüzdeki tüm hocalarıma teőekkür ederim.

Bu alıřmam boyunca yardımlarını benden esirgemeyen oda arkadaşlarıma teőekkür ederim.

Beni bu günlere getiren ve hayatım boyunca maddi manevi her türlü destekleriyle beni hiçbir zaman yalnız bırakmayan aileme de sonsuz teőekkürlerimi sunarım.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	i
ABSTRACT	iii
TEŞEKKÜR	v
İÇİNDEKİLER.....	vi
ÇİZELGELER	x
ŞEKİLLER	xii
SİMGELER VE KISALTMALAR	xiii
1. GİRİŞ.....	1
1.1. Problemin Tanımı.....	1
1.2. Önerilen Metot.....	3
1.3. Tez Organizasyonu.....	4
2. ÖN BİLGİ.....	5
2.1. Açık Kaynak Yazılımlar (AKY).....	5
2.1.1. Açık Kaynak Yazılımlar Hakkında Genel Bilgi ve Tarihçe.....	5
2.1.2. Açık Kaynak Yazılım Kullanımı.....	6
2.1.3. Açık Kaynak Yazılımların Tercih Edilme Sebepleri.....	9
2.1.3.1 Yazılım Kalitesi.....	9
2.1.3.2. Güvenlik.....	10
2.1.3.3. Üretici Firmaya Bağımlılık.....	10
2.1.3.4. Maliyetler.....	11
2.1.4. Açık Kaynak Yazılım Lisansları.....	15
2.2. Yazılım Kalite Modelleri.....	15
2.2.1. McCall Kalite Modeli.....	16
2.2.2. Boehm Kalite Modeli.....	18
2.2.3. FURPS Kalite Modeli.....	20
2.2.4. Dromey Kalite Modeli.....	20
2.2.5. ISO/IEC 9126 Kalite Modeli.....	21
2.2.6. ISO/IEC 25010.....	22
2.3. Yazılım Ölçme Çatıları (İng. Software Measurement Frameworks).....	23
2.3.1. ISO/IEC 15939 Ölçüm Süreci	23

2.4. Nesneye Yönelik Yazılım Metrikleri.....	26
2.4.1. Kalıtım Ağacının Derinliği (DIT).....	26
2.4.2. Alt Sınıf Sayısı (NOC).....	27
2.4.3. Sınıfın Ağırlıklı Metot Sayısı (WMC).....	27
2.4.4. Sınıfın Tetiklediği Metot Sayısı (RFC).....	28
2.4.5. Metotlardaki Uyum Eksikliği (LCOM).....	29
2.4.6. Nesne Sınıfları Arasındaki Bağımlılık (CBO).....	30
2.4.7. İç İçe Döngü Sayısı (NNL).....	30
2.4.8. Çevrimsel Karmaşıklık (CC).....	30
2.4.9. Açıklama Sayısı (NOS).....	31
3. İLGİLİ ÇALIŞMALAR.....	32
4. GENEL DEĞERLENDİRME SÜRECİ.....	39
4.1. Değerlendirme Yapılacak AKY'ların Seçilmesi (1. Aşama).....	41
4.2. Ölçülmek İstenen Kalite Özniteliğinin Seçilmesi (2.Aşama).....	41
4.3. Değerlendirme Yapılacak Boyutun Seçilmesi (3. Aşama).....	42
4.3.1. Kod-tabanlı Değerlendirme.....	42
4.3.2. Toplum-tabanlı Değerlendirme.....	43
4.4. Genel Sonuçları Elde Etmek İçin Adımların Geliştirilmesi.....	44
5. BAKIM YAPILABİLİRLİK VE GÜVENİLİRLİK İÇİN DEĞERLENDİRME YÖNTEMİ.....	45
5.1. Değerlendirme Yapılacak AKY'ların Seçilmesi.....	47
5.2. Değerlendirme Yapılacak Kalite Özniteliğinin Belirlenmesi.....	48
5.3. Değerlendirme Yapılacak Boyutun Belirlenmesi.....	48
5.3.1. Kod-tabanlı Değerlendirme.....	48
5.3.2. Toplum-tabanlı Değerlendirme.....	52
5.4. Genel Sonuçların Elde Edilmesi.....	55
6. GELİŞTİRİLEN YÖNTEMİN UYGULANMASI ve SONUÇLARI.....	56
6.1. Kod-tabanlı Değerlendirme Sonuçları.....	56
6.1.1. Bakım Yapılabilirliğin Değerlendirme Sonuçları.....	56
6.1.2. Güvenilirliğin Değerlendirme Sonuçları.....	59
6.2. Toplum-tabanlı Değerlendirme Sonuçları.....	59
6.2.1. Yıllara Göre Elektronik Posta Yoğunluğu (ePY).....	60
6.2.2. Hata (İng. Bug) Çözme Başarı İndeksi (HÇBİ).....	62
6.2.3. Hata Ciddiyet (İng. Bug Severity) İndeksi (HCİ).....	63

6.2.4. Ortalama Hata Çözme Süresine Göre Hata Çözme Başarı Oranı (OHÇS).....	64
6.2.5. Ürün Sürümlerine Göre Kümülatif Hata Yoğunluğu (SKHY).....	66
6.2.6. Toplum-tabanlı Değerlendirme Sonuçları.....	67
6.3. Genel Değerlendirme Sonuçları.....	69
7. ÜRÜNLERİN GERİYE DÖNÜK DEĞERLENDİRİLMESİ VE SONUÇLARIN KARŞILAŞTIRILMASI.....	71
7.1. Geriye Dönük Kod-tabanlı Değerlendirme Sonuçları.....	71
7.1.1. Bakım Yapılabilirliğin Geriye Dönük Değerlendirilmesi ve Sonuçları.....	71
7.1.2. Güvenilirliğin Geriye Dönük Değerlendirilmesi Sonuçları.....	72
7.2. Geriye Dönük Toplum-tabanlı Değerlendirme Sonuçları.....	74
7.2.1. Yıllara Göre Geriye Dönük Elektronik Posta Yoğunluğu (GD-ePY).....	75
7.2.2. Geriye Dönük Hata Çözme Başarı İndeksi (GD-HÇBİ).....	76
7.2.3. Geriye Dönük Hata Ciddiyet İndeksi (GD-HCİ).....	77
7.2.4. Ortalama Hata Çözme Süresine Göre Geriye Dönük Hata Çözme Başarı Oranı (GD-OHÇS).....	78
7.2.5. Ürün Sürümlerine Göre Geriye Dönük Kümülatif Hata Yoğunluğu (GD-SKHY).....	80
7.2.6. Geriye Dönük Toplum-tabanlı Değerlendirme Sonuçları.....	81
7.3. Geriye Dönük Genel Değerlendirme Sonuçları.....	83
7.4. Güncel ve Geriye Dönük Değerlendirme Sonuçlarının Karşılaştırılması.....	84
8. ÖNERİLEN METODUN OSMM VE OPENBRR KULLANILARAK DOĞRULANMASI.....	86
8.1. Modellerin Tanımı.....	86
8.1.1. Açık Kaynak Olgunluk Modeli.....	86
8.1.1.1. Ürün İndikatörü.....	87
8.1.1.2. Uygulana İndikatörü.....	91
8.1.2. Açık Kaynak İş Hazırlık Oranı.....	92
8.2. Modellerin Uygulanması.....	94
8.2.1. OSMM'nin Uygulanması.....	94
8.2.2 OpenBRR Modelinin Uygulanması.....	97
8.3. Sonuçların Karşılaştırılması.....	98
9. SONUÇLAR.....	100
9.1. Genel Sonuçlar.....	100

9.2. Çalışmaya ve Sonuçlara Yönelik Kısıtlar.....	101
9.3. Gelecek Çalışmalar.....	103
KAYNAKLAR	105
EK 1.....	109
EK 2.....	114
ÖZGEÇMİŞ.....	116

ÇİZELGELER

	<u>Sayfa</u>
Çizelge 2.1. Microsoft çözümlerinin yazılım maliyeti	12
Çizelge 2.2. AKY çözümün yazılım maliyeti	13
Çizelge 2.3. Microsoft karşısında AKY tasarrufları	13
Çizelge 2.4. AKY lisansları	15
Çizelge 2.5. McCall kalite modeli faktörleri	18
Çizelge 2.6. Boehm kalite modeli ve katmanları	19
Çizelge 2.7. McCall kalite modeli ile Boehm kalite modeli karşılaştırılması	20
Çizelge 2.8. FURPS kalite modeli	20
Çizelge 2.9. ISO/IEC kalite modeli niteliklerinin ve alt niteliklerinin tanımları.....	22
Çizelge 2.10. ISO/IEC 25010 kalite modeli öznitelikleri ve alt-öznitelikleri.....	23
Çizelge 3.1. Literatürde AKY seçimi için önerilen metotlar ve deneysel çalışmaların karşılaştırılması	35
Çizelge 4.1. En bilinir kalite modelleri ve öznitelikleri	42
Çizelge 5.1. Belirlenen aday AKY ürünlerinin özellikleri.....	47
Çizelge 5.2. Bakım yapılabirlik ve güvenilirliği ölçmek için metrikler	49
Çizelge 5.3. Güvenilirliğin değerlendirilmesi için kullanılan metriklerin eşik (threshold) değerleri.....	50
Çizelge 5.4. Güvenilirliğin hesaplanması için kullanılan kurallar ve formüller.....	51
Çizelge 5.5. Toplum-tabanlı değerlendirme için belirlenen bilgi ihtiyaçları ve metrikler.....	53
Çizelge 6.1. Her bir AKY ürünü için metrik ölçüm sonuçları	56
Çizelge 6.2. Bakım yapılabirliğin alt özniteliklerine göre AKY ürünlerinin tercih edilme sıralaması	57
Çizelge 6.3. Bakım yapılabirlik indeksi (BYİ) ve hesaplanması	58
Çizelge 6.4. Güvenilirlik indeksi (Gİ) hesaplanması için verilen ağırlıklar ve uygulanan kurallar	59
Çizelge 6.5. E-posta yoğunluğu indeks değerleri.....	61
Çizelge 6.6. Hata çözme başarı indeksi (HÇBİ).....	62
Çizelge 6.7. Ürünlerin her bir hata seviyesindeki indeksleri ve HCI.....	63
Çizelge 6.8. Son iki yılda açılan hataların çözülmesi ile ilgili bilgiler	65
Çizelge 6.9. Hata yoğunluğu	67

Çizelge 6.10. Toplum-tabanlı değerlendirme indeksi (TDİ) ve hesaplanması	68
Çizelge 6.11. Kod-tabanlı ve toplum-tabanlı değerlendirme sonuçları	69
Çizelge 7.1. Her bir AKY ürünü için metrik ölçüm sonuçları (2013 verilerine göre)	71
Çizelge 7.2. Bakım yapılabilirliğin alt özneliklerine göre AKY ürünlerinin tercih edilme sıralaması (2013 yılı verilerine göre)	72
Çizelge 7.3. Geriye dönük bakım yapılabilirlik indeksi (GD-BYİ) ve hesaplanması	73
Çizelge 7.4. Geriye dönük güvenilirlik indeksi (GD-Gİ) hesaplanması için verilen ağırlıklar ve uygulanan kurallar	74
Çizelge 7.5. Geriye dönük e-posta yoğunluğu ve ağırlık sonuçları	75
Çizelge 7.6. Geriye dönük hata çözme başarı indeksi (GD-HÇBİ)	77
Çizelge 7.7. Ürünlerin her bir hata seviyesindeki indeksleri ve GD-HCİ.....	78
Çizelge 7.8. 2013 yılından önceki iki yıllık dönemde açılan hataların çözülmesi ile ilgili bilgiler.....	79
Çizelge 7.9. 2013 yılına kadar olan veriler temel alınarak hesaplanan hata yoğunluğu	80
Çizelge 7.10. Geriye dönük toplum-tabanlı değerlendirme indeksi (GD-TDİ) ve hesaplanması	82
Çizelge 7.11. Geriye dönük kod-tabanlı ve toplum-tabanlı değerlendirme sonuçları	83
Çizelge 7.12. 2013 ve 2017 yıllarındaki kod-tabanlı ve toplum-tabanlı değerlendirme sonuçları	84
Çizelge 7.13. 2013 ve 2017 yıllarında yapılan genel değerlendirme sonuçlarının karşılaştırılması	85
Çizelge 8.1. OSMM ürün indikatörleri ve her indikatör için skor verme kriterleri.....	90
Çizelge 8.2. OpenBRR değerlendirme kategorileri, değerlendirme metrikleri ve skor verme kriterleri.....	93
Çizelge 8.3. OSMM'ine göre AKY'ların değerlendirme sonuçları.....	97
Çizelge 8.4. OpenBRR modeline göre hesaplanan metrik değerleri ve değerlendirme sonuçları.....	98
Çizelge 8.5. Önerilen metot ile OSMM ve OpenBRR modellerinin değerlendirme sonuçlarının karşılaştırılması.....	99

ŞEKİLLER

	<u>Sayfa</u>
Şekil 1.1. AKY kullanımının yıllara göre değişimi	1
Şekil 1.2. Github'ın kullanıcı sayısındaki değişim	2
Şekil 2.1. AKY kullanımının sektörel dağılımı	6
Şekil 2.2. AKY'ların tercih edilme sebebinin yıllara göre değişimi	14
Şekil 2.3. McCall kalite üçgeni	16
Şekil 2.4. Dromey kalite modeli	21
Şekil 2.5. ISO/IEC 9126 kalite modeli.....	21
Şekil 2.6. ISO/IEC 15939 ölçüm sürecinin aşamaları	24
Şekil 2.7. ISO/IEC 15939 ölçüm modelindeki temel kavramlar ve ilişkileri	25
Şekil 2.8. Kalıtım ağacının derinliği	27
Şekil 2.9. Kod içerisinde çağırılan toplam metot sayısı	28
Şekil 2.10. Metotlar arasındaki uyum	29
Şekil 2.11. Sınıflar arasındaki bağımlılık	30
Şekil 2.12. Basit bir programın kontrol akış grafiği	31
Şekil 4.1. AKY'ların değerlendirmesi için geliştirilen genel metodun aşamaları....	40
Şekil 5.1. AKY'ların bakım yapılabilirliğini ve güvenilirliğini değerlendirmek için geliştirilen metodun aşamaları	46
Şekil 6.1. Yıllara göre kümülatif e-posta yoğunluğu.....	61
Şekil 6.2. Son iki yılda açılan hataların açıldığı günden itibaren 30 gün içerisinde çözümlenme oranları	65
Şekil 6.3. Sürümlere göre kümülatif hata yoğunluğu	67
Şekil 7.1. Yıllara göre kümülatif e-posta yoğunluğu (2013 yılına kadar olan verilere göre).....	76
Şekil 7.2. 2013 yılından önceki iki yıllık dönemde açılan hataların açıldığı günden itibaren 30 gün içerisinde çözümlenme oranları	79
Şekil 7.3. 2013 yılına kadar olan veriler temel alınarak hesaplanan, sürümlere göre kümülatif hata yoğunluğu	81
Şekil 8.1. Her bir üründeki üyelerin ürüne sağladığı katkıların yüzdelik gösterimi.....	94

SİMGE VE KISALTMALAR

Simgeler

\log	Logaritma
Σ	Toplam Sembolü
$\bar{U} - Lt$	Üst Eşik Değer
$A - Lt$	Alt Eşik Değer

Kısaltmalar

AD	Ağırlık Değeri
AKY	Açık Kaynak Yazılım
BugDB	Problem Hata Raporu
BYİ	Bakım Yapılabilirlik İndeksi
CBO	Nesne Sınıfları Arasındaki Bağımlılık
CC	Çevrimsel Karmaşıklık
CVS	Eşzamanlı Sürüm Kontrolü
DIT	Kalıtım Ağacının Derinliği
ePY	Elektronik Posta Yoğunluğu
GD-BYİ	Geriye Dönük Bakım Yapılabilirlik İndeksi
GD-ePY	Geriye Dönük Elektronik Posta Yoğunluğu
GD-Gİ	Geriye Dönük Güvenilirlik İndeksi
GD-HÇBİ	Geriye Dönük Hata Çözme Başarı İndeksi
GD-HCİ	Geriye Dönük Hata Ciddiyet İndeksi
GD-OHÇS	Geriye Dönük Ortalama Hata Çözme Süresi
GD-SKHY	Geriye Dönük Sürümlere göre Kümülatif Hata Yoğunluğu
GD-TDİ	Geriye Dönük Toplum-tabanlı Değerlendirme İndeksi
Gİ	Güvenilirlik İndeksi
HCİ	Hata Ciddiyet İndeksi
HÇBİ	Hata Çözme Başarı İndeksi
IDE	Tümleşik geliştirme ortamı
LCOM	Metotlardaki Uyum Eksikliği

NNL	İç İçe Döngü Sayısı
NOC	Alt Sınıf Sayısı
NOS	Açıklama Sayısı
OHÇS	Ortalama Hata Çözme Süresi
OpenBRR	Açık Kaynak İş Hazırlık Oranı
OSMM	Açık Kaynak Olgunluk Modeli
QSOS	Açık Kaynak Yazılım Niteliği ve Seçimi
RFC	Sınıfın Tetiklediği Metot Sayısı
SKHY	Sürümlere göre Kümülatif Hata Yoğunluğu
SQO-OSS	Açık Kaynak Yazılımlar için Yazılım Kalite Gözlemi
TDİ	Toplum-tabanlı Değerlendirme İndeksi
WMC	Sınıfın Ağırlıklı Metot Sayısı

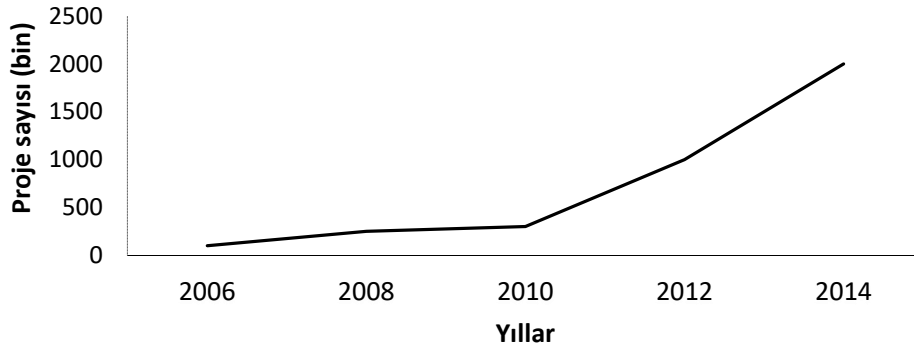
1. GİRİŞ

1.1. Problemin Tanımı

Açık kaynak yazılımlar (AKY), kaynak kodları özel bir telif hakkı lisansı ile herkesin incelemesine, kullanımına ve dağıtımına açılan, böylece kullanıcıya yazılımı değiştirme özgürlüğü sunan ve dünyanın her tarafından bilişim uzmanlarınca imcece yöntemi ile endüstri standartlarında geliştirilen yazılımlardır [1].

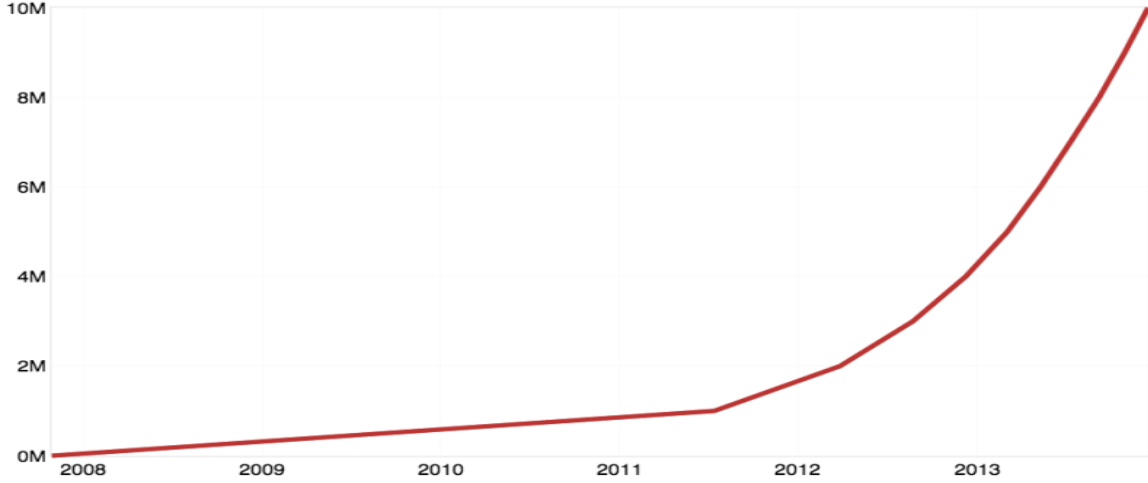
Açık kaynak yazılımların kullanımı son yıllarda büyük oranda artmıştır. Bu yazılımların tercih edilmesinin sebepleri geçmişten günümüze farklılık göstermektedir. Geçmiş yıllarda açık kaynak yazılım kullanımının ana sebebi, mevcut kod tabanının ihtiyaca göre değiştirilerek tekrar kullanımının kaynak ve zaman tasarrufu sağlamasıydı (tekrar kullanılabilirlik). Son yıllarda ise açık kaynak yazılımların tercih edilmesinin temel sebebi bu yazılımları yüksek kaliteli, güvenilir ve güvenli görülmeye başlanması olmuştur. Böyle algılanmasındaki temel sebep ise bu yazılım ürünlerinin birçok geliştiricinin dikkatli incelemesinden geçmiş ve dolayısıyla hatalardan arındırılmış olduğunun kabul edilmesidir.

Şekil 1.1'de verilen Black and Duck Software [2] ve North Bridge Venture [3] organizasyonlarının raporuna göre, özellikle 2010 yılından sonra açık kaynak yazılımların kullanımı dikkat çekecek şekilde artmıştır. 2011 yılında Marc Andreessen makalesinde yazılımın dünyayı ele geçirdiğini savunmuştur [4]. Fakat yine aynı organizasyonların raporuna göre bu algı değişmiş ve "açık kaynak yazılımlar dünyayı ele geçiriyor" haline gelmiştir [5].



Şekil 1.1 AKY kullanımının yıllara göre değişimi [2][3]

Açık kaynak yazılımların popülerliğinin bu denli artması sonucunda, açık kaynak yazılım ürünleri sağlayan Github, SourceForge, Apache, Debian ve Savannah vb. sitelerdeki proje ve kullanıcı sayıları da devasa şekilde büyümüştür.



Şekil 1.2 Github'ın kullanıcı sayısındaki değişim [6]

Şekil 1.2'de sadece Github'ın 2008 yılından bu yana kullanıcı sayılarının artışı gösterilmiştir. 2013 yılından sonra yaklaşık olarak aktif kullanıcı sayısı 10 milyonu geçmiştir. Ayrıca SourceForge'nin şu anki verilerine baktığımızda haftada kullanıcılar tarafından ortalama 32 milyon proje indirildiği gözlenmektedir.

Açık kaynak yazılımların sayısı bu derece artınca yazılım ürünlerinin alternatiflerinin sayısı da otomatik olarak artmıştır. Kullanıcıların açık kaynak yazılımları benimsediği ve yanlış yazılım ürünü seçiminin şirketleri büyük maddi kayıplara maruz bıraktığı düşünüldüğünde, ihtiyacı karşılayan doğru ürünü seçme işi gerçekten önemli ve zor bir süreç haline almıştır. Dolayısıyla ticari yazılımlarda olduğu gibi açık kaynak yazılımlarda da en büyük problem, değerlendirme ve seçme işlemi olmuştur [7]. Ticari yazılımlar kadar olmasa da akademik ve endüstri alanlarında, açık kaynak yazılımların değerlendirilmesi ve seçilmesi için bazı metotlar önerilmiştir [8][9][10][11]. Ne var ki standartlaşmış bir metot yoktur ve bu sebeple yapılan çalışmaların yetersiz kaldığı söylenebilir. Bu eksiklikten dolayı ihtiyaç sahipleri ihtiyaçlarını karşılayan açık kaynak yazılım ürünlerini subjektif değerlendirmeler veya tavsiyeler üzerine seçmektedir [12]. Dolayısıyla alternatifler arasından ihtiyaç sahiplerinin kalite gereksinimlerini en iyi karşılayanın seçilmesi problemi, araştırılması gereken cazip bir problem haline gelmiştir.

Açık kaynak yazılımların en büyük özelliği kaynak kodlarının kullanıcıya açık olmasıdır ve literatürde bu kaynak kodları ölçmek birçok metrik mevcuttur. Ama hangi kalite özneliğinin hangi metrik değerleri kullanılarak nasıl ölçüleceği konusunda sınırlar ve kurallar net olarak belirtilmemiştir.

Açık kaynak yazılımların diğer bir özelliği ise bu ürünlerin piyasaya sürüldüğünden itibaren veri tabanlarında tarihi verilerinin (İng. historical data) tutulmasıdır. Bu veriler karşılaşılan hata (İng. bug) sayıları, eklenen ve silinen satır sayıları, aktif geliştirici (İng. developer) sayıları, hata önem dereceleri (İng. bug severity), geliştirici-geliştirici ve kullanıcı-geliştirici arasındaki eposta (İng. mail) sayıları, hatanın durumu (açık, kapalı, çözüldü) (İng. bug status (opened, closed, fixed), sıkça sorulan sorular vb. verilerdir. Ancak açık kaynak kodlu yazılımların değerlendirilmesinde ihtiyaçların nasıl belirleneceği ve bu ihtiyaçlar doğrultusunda bahsedilen verilerin nasıl kullanılacağı gerçekten karışık bir durumdur.

Bu tez çalışmasında önerilen metot, açık kaynak kodlu yazılımların değerlendirilmesi sürecinde, bakım yapılabilirlik ve güvenilirlik bakımından yukarıda zikredilen karışıklıkları ortadan kaldırmayı hedeflemektedir.

1.2. Önerilen Metot

Bakım yapılabilirlik ve güvenilirlik geçmişten günümüze hemen hemen bütün kalite modelleri tarafından önerilmiş, kaliteyi ölçmekte kullanılan iki önemli kalite özneliğidir. Bu yüzden bu tez çalışmasında, açık kaynak kodlu yazılımları seçme ve değerlendirme işleminde bu iki kalite özneliğini ölçen, iki boyutlu bir metot geliştirilmiştir. Geliştirilen bu metotta açık kaynak kodlu yazılım ürünü hem **kod-tabanlı** hem de **toplum-tabanlı** olarak iki açıdan değerlendirilmiş ve sonuçlar birleştirilmiştir.

Kod-tabanlı değerlendirme yapılırken oluşturulan araştırma soruları ile bu iki özneliği ölçmeye yönelik metrikler belirlenmiştir. Belirlenen metriklerin ölçüm sonuçları kullanılarak ürünlere, göreceli kalite düzeylerini ifade eden ağırlık değerleri verilmiştir.

Toplum-tabanlı değerlendirme yapılırken ISO/IEC 15939 yazılım ölçümü süreç modeli takip edilmiştir. Öncelikle bilgi ihtiyaçları (İng. information needs) belirlenmiş, daha sonra bu bilgi ihtiyaçlarını ölçmek için ürünlerin veri tabanlarındaki bilgiler derinlemesine analiz edilmiş ve ihtiyacı karşılayacak metrikler seçilmiştir. Kod-

tabanlı deęerlendirmede olduęu gibi burada da rnlere, metrik sonuları kullanılarak greceli kalite dzeylerini ifade eden aęırlık deęerleri verilmiřtir.

Hem kod tabanlı hem de toplum tabanlı deęerlendirmede kullanılan aęırlıklar sonucunda en iyi rnler ayrı ayrı belirlendikten sonra, lm yapılan boyutlara da ihtiyaca gre aęırlıklar verilmiřtir (rneęin: kod-tabanlı deęerlendirme: %55, toplum tabanlı deęerlendirme: %45). Tm bu verilen aęırlıklar sonucunda hem bakım yapılabilirlik hem de gvenilirlik iin ayrı ayrı, en tercih edilir rnler belirlenmiřtir.

1.3. Tez Organizasyonu

Bu tez alıřmasının geri kalan kısmı řu řekilde oluřturulmuřtur: İkinci blmn birinci kısmında AKY'ların tarihesi, kullanımı ve tercih edilme sebepleri ile ilgili genel bilgiler verilmiřtir. İkinci blmn ikinci kısmında tez alıřmasında kullanılacak znelikleri sememize yardımcı olan kalite modelleri tanıtılmıřtır. İkinci blmn nc kısmında bu tez alıřmasında AKY'ların deęerlendirilmesinde kullanılan ISO/IEC 15939 sistem ve yazılım mhendislięi iin lm sreci anlatılmıřtır. İkinci blmn drdnc kısmında ise lmlerde kullanılacak nesneye ynelik yazılım metrikleri anlatılmıřtır. nc kısmında bu alanda yapılan ilgili alıřmalar anlatılmıř ve bizim alıřmamızın yapılan alıřmalarla benzer ve farklı ynleri tartıřılmıřtır. Drdnc blmde bu tez alıřmasında nerilen metodu oluřtururken izlenen genel sre anlatılmıřtır. Beřinci blmde, tez alıřması kapsamı iin seilen bakım yapılabilirlik ve gvenilirlik kalite zneliklerini deęerlendirme iin oluřturulan metot anlatılmıřtır. Altıncı blmde, bu metot uygulanarak  Java inřa aracının gncel verilerle bakım yapılabilirlik ve gvenilirlik deęerlendirmesine iliřkin detaylar ve sonular verilmiřtir. Yedinci blmde metodun tutarlılıęını lmek iin, geriye dnk verilerle araların aynı zneliklerini lmeye ynelik uygulama detayları ve sonuları verilmiřtir. Yedinci blmn devamında gncel deęerlendirme sonuları ile geriye dnk deęerlendirme sonuları karřılařtırılmıřtır. Sekizinci blmde nerilen metodun lm sonularının doęrulanması iin, literatrde AKY'ları deęerlendirmeye yarayan ve en popler metotlardan olan OSMM ve OpenBRR kullanılmıřtır. Hem gncel olarak elde edilen hem de geriye dnk elde edilen sonular, bu metotların sonuları ile karřılařtırılmıřtır. Son blm olan dokuzuncu blmde ise genel sonular zetlenmiř, alıřmaya ve sonulara ynelik kısıtlar ile gelecek alıřmalardan bahsedilmemiřtir.

2. ÖN BİLGİ

2.1. Açık Kaynak Yazılımlar (AKY)

2.1.1. Açık Kaynak Yazılımlar Hakkında Genel Bilgi ve Tarihçesi

AKY'lar kullanıcıların çeşitli sebeplerle; yazılımı çalıştırma, kendi ihtiyaçlarına göre değiştirme ve yazılım ürününü kaynak kodu ile dağıtma hakkına sahip olduğu yazılımlardır. Açık Kaynak Girişimine (İng. Open Source Initiative) göre bir yazılımın açık kaynak yazılım sayılabilmesi için şu özelliklere sahip olması gerekir [13]:

1. Yazılımın ücretsiz dağıtılabilmesi
2. Yazılımın kaynak kodunun yazılımla beraber sunulması veya ücretsiz olarak elde edilme imkânı sunulması
3. Yazılımın kaynak kodunda değişiklik yapıldığında yine aynı lisans adı altında dağıtımına izin verilmesi
4. Lisans anlaşmasının belli bir gruba veya kişiye has olmaması
5. Yazılımın kullanımıyla ilgili bir kısıtlamanın bulunmaması
6. Lisansın, yazılımın ulaştığı tüm kişiler için ek süreç gerekmeksizin geçerli olması
7. Lisansın yazılımla üretilen belli bir ürüne has değil, yazılımın kendisine has olması

AKY kavramından 1960'lı yıllarda Amerika'da bulunan Massachusetts Teknoloji Enstitüsü, Stanford Üniversitesi, Kaliforniya Üniversitesi (Berkeley), Carnegie Mellon gibi araştırmaları ile ünlü üniversitelerin yapay zekâ laboratuvarında bahsedilse de ilk resmi adım 1970'li yıllarda atılmıştır [14]. Bu yıllarda Amerikalı bir yazılım geliştirici olan Richard Stallman çok sayıda kullanıcıya sahip işletim sistemi olan UNIX'in kaynak kodu açık bir sürümünü piyasaya sürmüştür. Herkese özgür bir şekilde kaynak kodunu düzenleme imkânı sunan bu sürüm, kullanıcılar tarafında ilgi görmüştür.

AKY'ların kurumsallaştırılması, ilk kez 1983 yılında yine Richard Stallman tarafından, çalıştığı Massachusetts Teknoloji Enstitüsü laboratuvarında bulunan Xerox marka yazıcıya ait sürücü yazılımına ait kaynak kodların kapatılmasına karşı konularak başlamıştır [15]. Daha sonra 1989 yılında açık kaynak kodlu yazılımların

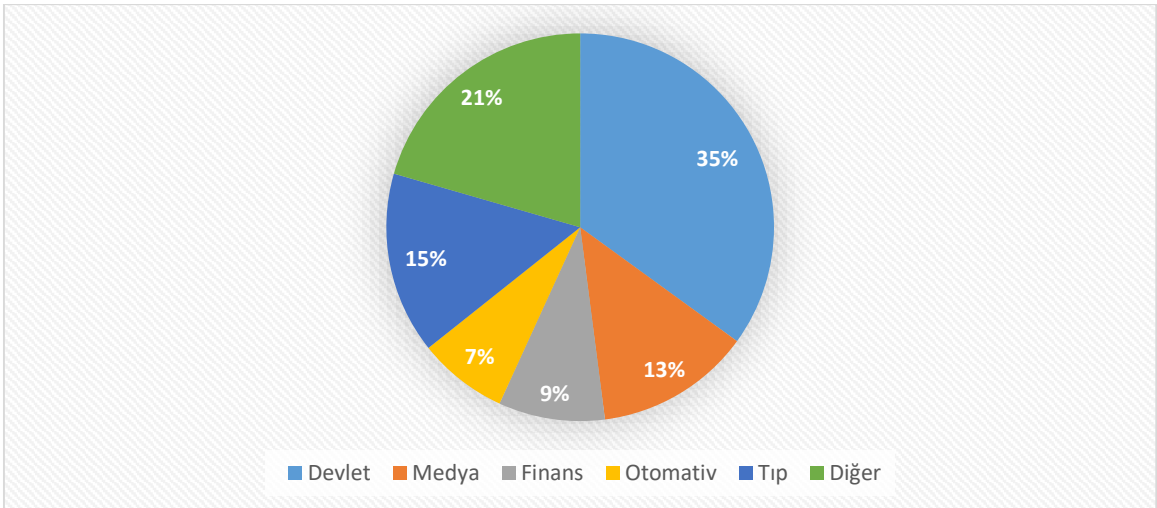
çerçevesini çizmek ve bunların sağlam temellere oturtmak için Genel Kamu Lisansı (GPL- General Public Licence) geliştirilmiştir [15].

Diğer önemli bir gelişme 1998 yılında yaşanmıştır. Microsoft firmasının web tarayıcısı Internet Explorer karşısında büyük ölçüde pazar payı kaybeden Netspace şirketi, kaybettiği konumunu tekrar kazanmak için 1998'in Ocak ayında kritik bir karar alarak web tarayıcısının kaynak kodlarını kullanıcıya açmıştır. Firmanın bu hareketi AKY'ların gelişimi için önemli bir adım olarak görülmüştür [15].

AKY'ların popülerliği bu yıllardan sonra da her geçen gün artan bir grafik izlemiştir. Şekil 1.1'de gösterildiği gibi özellikler 2010 yılından sonra dikkat çekici bir artış göstermiş ve bazı ticari yazılımların yerini alacağına sinyallerini vermeye başlamıştır.

2.1.2. Açık Kaynak Yazılımın Kullanımı

AKY'ların yüksek kaliteli ve güvenilir olması ve kullanıcıların satıcı firmalara bağımlılığını ortadan kaldırması ile bu yazılımlar, kitleler tarafından hızlı bir şekilde tercih edilmeye başlanmıştır. Şekil 2.1'de dünya genelinde AKY'ların kullanım alanları gösterilmiştir [2][3]. Şekilde AKY kullanımının dünya genelinde en çok, özellikle hükümetler tarafında kamu kurumlarında olduğu görülmektedir. Tıp alanında ve medya alanında AKY kullanımı bunu takip etmektedir.



Şekil 2.1 AKY kullanımının sektörel dağılımı [2][3]

Birçok ülke ve kuruluş açık kaynak kodlu yazılım kullanımının artması için çalışmalarda bulunmuştur:

Avrupa Komisyonu: Avrupa Komisyonu Özgür Yazılım Çalışma Grubu'nun [16] kurulması ile birlikte açık kaynak yazılım kullanımına çeşitli girişimlerle destek vermiştir. AKY'ları destekleyen ve yaygınlaşmasını sağlayan birçok projeyi Avrupa Birliği 6. ve 7. Çerçeve Programları kapsamında desteklemiştir. Avrupa Komisyonu'nun desteğiyle AKY'lar için tecrübelerin ve iyi örneklerin paylaşıldığı iş birliği çalışma grubu oluşturulmuştur [14]. Bu iş grubu sayesinde iyi hazırlanmış örnekler ve çalışmalar Avrupa Komisyonu'nun İnternet sitesinden yayınlanmıştır [14]. Avrupa Birliği tarafından hazırlanan (Avrupa'nın Dijital Rekabet Edebilirlik Raporu-2010 (İng. Digital Agenda for Europe) raporunda, AKY'nın pazar payının her yıl ortalama %30-40 arasında artacağı ve bu yazılımların yenilikçi fikirlere açık olduğu belirtilmiştir.

İngiltere: İngiltere ilk olarak 2004 yılından sonra AKY'ların kullanımını politik seviyede desteklemiştir. AKY kullanımını kendi kamu kuruluşlarında zorunlu hale getirmemiştir fakat kullanılan ticari yazılımların AKY alternatiflerinin araştırılmasını teşvik edici yollar izlemiştir. Daha sonra kamu kuruluşlarında AKY kullanımı ciddi bir boyutta artmaya başlamış ve 300 binden fazla kullanıcı Linux altyapısı ile çalışmaya başlamıştır [14].

Yayınlanan bir eylem planı (Açık kaynak kodu, Açık Standart ve Yeniden Kullanım-2009" [17]) ile İngiltere de AKY kullanımını kendi kamu kuruluşlarında zorunlu hale getirilmiştir. Ticari yazılım tercih edenlerden detaylı gerekçeler istenilmiş ve ancak geçerli sebepler sunanlara izin verilmiştir.

İngiltere'de tüm kamu kurumlarınca kullanılan, AKY kullanılarak ve açık standartlar gözetilerek oluşturulmuş, ortak bir iletişim altyapısı bulunmaktadır. Kamu kurumlarınca ihtiyaç duyulan tüm yazılımlar bu altyapı üzerinde çalıştırılmakta, bu sayede uygulamaların birlikte çalışabilirliği temin edilebilmektedir [14].

Hollanda: Hollanda'da AKY kullanımına ilişkin çalışmalar Maliye Bakanlığı tarafında takip edilmektedir. Yayınlanan bir eylem planı (Açık Standartlar ve Açık Kaynak Kodlu Yazılım-2003 (İng. Open Standarts and Open Source Software)) ile Hollanda kendi kamu kurumlarında AKY kullanımını teşvik etmiştir. 2007 yılında bu eylem planı çerçevesinde bir anket yapılmış ve kamu kurumlarının % 60'ının AKY benimsediği görülmüştür. Ancak Hollanda Hükümeti bunu yetersiz bulmuş ve 2007

yılından sonra AKY kullanımını arttırmak için eylem planının kapsamını genişletmiştir [14].

Malezya: AKY kullanımını teşvik amaçlı eylem planı yayınlayan ülkelerden biri de Malezya'dır ve 2004 yılında hazırladığı eylem planında (Malezya Kamu Sektörü Açık Kaynak Kodlu Yazılım Programı [14]) üç aşamalı bir çalışma planlamıştır. İlk aşama 2006, ikinci aşama 2010 yılında tamamlanmış ve üçüncü aşama hala devam etmektedir. Bu eylem planından sonra Malezya'da kamu kuruluşlarında AKY kullanımı büyük oranda artmış ve Gartner şirketinin [18] (bilgi teknolojisi araştırma ve danışmanlık şirketi) raporlarına göre şu kazanımlar elde edilmiştir [14]:

- Lisans masraflarında %80 tasarruf
- Danışmanlık ve geliştirme çalışmalarında %58 azalma
- Yazılım destek hizmetlerinde %7 tasarruf
- Toplamda yaklaşık %30 tasarruf sağlanmıştır.

Türkiye: Türkiye'de AKY kullanımı e-Dönüşüm projesinde (2003) ele alınan önemli konulardan biridir [14]. Bu bağlamda ilk olarak, 2005 Eylem Planı 7 no'lu (Kamu Kurum ve Kuruluşlarında Açık Kaynak Kodlu Yazılımların Uygulanabilirliği) eylemi ile Kalkınma Bakanlığı Bilgi Toplumu Dairesi tarafından ilgili paydaşların katılımıyla konuya ilişkin bir rapor yayınlanmıştır [14]. Bu raporda, AKY'ların temel özellikleri, tarihçesi, kullanım alanları ve sağladığı avantajlar yer almakta, ayrıca AKY hukuki ve mali yönden incelenmektedir [14].

Türkiye'de AKY kullanımı kamu kuruluşlarında istenilen seviyeye henüz ulaşmamıştır. Ama bu bağlamda TÜBİTAK bünyesinde Linux temel alınarak geliştirilen açık kaynak kodlu bir işletim sistemi olan PARDUS [19], Türkiye'de birçok kamu kuruluşunda kullanılmaya başlanmıştır. Bu işletim sisteminin geliştirilmesi için 2003 yılında TÜBİTAK-BİLGEM bünyesinde çalışmalara başlanmış ve 2005 yılında ilk sürümü ortaya çıkmıştır. PARDUS bireysel ve kurumsal olmak üzere iki farklı sürüme sahiptir ve günümüzde 32-bit ve 64-bit sürümleri mevcuttur.

Adalet Bakanlığı, Ulusal Yargı Ağı Projesi'nin (UYAP) başlatıldığı tarih olan 1999 yılından bu yana AKY'lara yönelik çalışmalar yürütülmekte olup, ilk AKY'ye kapsamlı geçiş çalışmalarına 2006 – 2007 yıllarında başlamış ve yaklaşık 50.000 bilgisayar kullanıcısının bulunduğu kurumda işletim sisteminin PARDUS'a taşınması planlanmıştır [14].

450 kişilik tek merkezli bir kurum olan Enerji Piyasası Düzenleme Kurumu'nda (EPDK), AKY'lara yönelik çalışmalar 2009 yılında başlamıştır. Sunucular Temmuz 2010'dan beri PARDUS kurumsal işletim sistemi üzerinde çalışmakta olup kullanıcı bilgisayarlarının da yakın zamanda PARDUS'a geçirilmesi planlanmıştır [14].

Milli Savunma Bakanlığı PARDUS işletim sistemini, piyasaya sürülüşünden beri takip etmekte ve piyasaya sürüldüğünden beri bünyesinde kullanmaktadır. MSB'nin Muhabere Elektronik Bilgi Sistemleri Dairesi, misyon metninde şu ifadelerle yer vererek AKY kullanımını desteklediğini vurgulamıştır [15]:

*"MSB'nin gelecek beş yıl içinde hedefleri, modern teknolojiye sahip, modüler, **açık kaynak kodlu** ve millî olarak üretilmiş yazılımları kullanan, "e-Devlet" kapsamında "vatandaş odaklı" hizmetler sunan, siber güvenliğe ve bilişim hukukuna uygun, "Sistemlerin Sistemi" anlayışıyla tasarlanmış, "Sistemlerin Federasyonu" mantığıyla yönetilen MSB Bilgi Sistemine sahip olmaktır."*

Çevre ve Şehircilik Bakanlığı tüm kurumlarında açık kaynak kodlu ofis programı ve açık kaynak e-posta uygulaması kullanmaktadır.

Yukarıda bahsedilen ülkelerin dışında Fransa, Almanya, İspanya, Meksika, Brezilya, Kore ve Hindistan, kendi kamu kuruluşlarında AKY kullanımını benimsemiş ve bilgi toplumu stratejilerinin bir parçası haline getirmişlerdir [20]. Dünya Bankası ve UNESCO gibi kuruluşlar da AKKY kullanıma destek vermişlerdir.

2.1.3. Açık Kaynak Yazılımların Tercih Edilme Sebepleri

AKY'lar çeşitli sebeplerden dolayı kullanıcılar tarafından tercih edilmektedir. Başlıca sebepleri şu şekilde açıklayabiliriz:

2.1.3.1. Yazılım Kalitesi

Son yıllarda AKY'lar ticari yazılımlara nazaran daha kaliteli ürünler üreterek karşımıza çıkmaktadır. Bunun en önemli sebeplerinden bir tanesi AKY'ların dünyanın dört bir tarafından yazılım geliştiricilerin dikkatli incelemesinden geçmesi ve hatalarından arındırılmış olmasıdır. Ürün bazında AKY'lar ticari yazılımlara göre daha kaliteli olmasına rağmen dokümantasyon kalitesi bakımından ticari yazılımlar seviyesine henüz ulaşmamıştır. Bunun en büyük sebebi ise AKY ekiplerinin kod geliştirici kimliklerinin iyi olması fakat yardım dokümantasyonlarının hazırlanması konusunda çok titiz olmamalarıdır.

2.1.3.2. Güvenlik

Piyasada var olan bütün yazılım ürünleri arka kapılara (İng. backdoor) sahip olabilir. Arka kapı, bilgisayar sistemlerinin normal güvenlik işleyişini göz ardı eden bir yöntemdir ve bu sayede bilgisayar sistemi yetkisiz erişim ve işlemlere açık hale gelir [14][15]. Bu durum özellikle bilgi güvenliğinin üst seviyede olduğu kurumlarda çok kritik bir durumdur. Ticari yazılımlarda kullanılan ürünün kaynak kodlarına sadece üretici firma tarafından erişildiği düşünüldüğünde, yazılım ürününün arka planda neler yaptığının kullanıcılar arasında bilinmesi mümkün değildir. Yazılımın içinde kullanıcıya zarar verebilecek, bilgilerini çalıp başka birimlere gönderebilecek kodların eklenip eklenmediğinin kimse garantisini veremez. Aynı zamanda ticari yazılımlarda yazılım üreticisi kullanıcının elinde olmayan güncellemelerle yazılım ürününe müdahale edebilir. Örneğin, geçtiğimiz yıllarda Microsoft korsan Windows kullanıcılarının önüne geçmek için Çin’de bütün korsan Windows kullanıcılarına bir güncelleme gönderdi. Bu güncellemeden sonra korsan Windows kullananlar bilgisayarlarını açtığında bir hata ile karşılaştı. Örnekte olduğu gibi bu güncellemeler bu örneğin aksine kötü amaçlarla da kullanılabilir. AKY’ların kodlarına erişilebildiği için bu ve bunun gibi sorunlarla karşılaşmayacağına garantisi geliştiricilerin ve kullanıcıların elindedir. Bu yüzden kullanıcılar tarafından AKY’lar daha güvenli olduğu için tercih edilmeye başlanmıştır.

2.1.3.3. Üretici Firmaya Bağımlılık

Ticari yazılımlarda üretici firmalar çeşitli anlaşmalarla kullanıcılarını kendilerine bağlamakta ve belli bir süre zarfında o firmanın yazılım ürünlerinden başka yazılımlar kullanılamamaktadır. Bu durum kullanıcıları kısıtlamaktadır. Diğer bir sorun ise üretici firmaların kullanıcıya sağladığı ürünlerin sadece kendi yazılım ürünleriyle uyum içinde çalışması ve rakip firmaların yazılım ürünleriyle pek uyuşmamasıdır. Üretici firmalar böyle yaparak rekabeti ortadan kaldırmayı hedeflemekte ve kullanıcıları kendilerine bağımlı hale getirmeye çalışmaktadır. AKY’larda bu ve bunun gibi sorunlar tamamen ortadan kaldırılmıştır. AKY kullananların elinde ürünlere ait kaynak kodlar da olduğundan üretici firmalara olan bağımlılık ortadan kalkar ve kullanıcılar yazılım ürünlerini kendi ihtiyaçlarına göre kolayca değiştirebilirler.

2.1.3.4. Maliyetler

AKY lisansları bize yazılımı özgür kullanma şansı sunar ve bu lisanslar için bir ücret ödenmez. Ancak yazılımın Toplam Sahip Olma Maliyeti (İng. Total Cost of Ownership-TCO) hesaplanırken sadece yazılım ürünlerinin lisans bedelleri göz önüne alınmaz. Toplam Sahip Olma Maliyeti bakımından AKY ile ticari yazılımlar arasında karşılaştırma yapılırken değerlendirilmesi gereken maliyetler: donanım maliyetleri (satın alma ve bakım dâhil), doğrudan yazılım maliyeti (satın alma, destek ve bakım dâhil), dolaylı yazılım maliyeti (özellikle lisans maliyeti), personel maliyeti, destek maliyeti ve sistemdeki aksamaların maliyetleridir [15]. Yazılımların ilk elde edinimi sırasında AKY lisans bedeli ödenmeden elde edilirken, ticari yazılımlara büyük miktarlarda lisans bedeli ödenerek elde edilir. Bu durum AKY'ların tamamen ücretsiz olduğu anlamına gelmez. Çünkü AKY'lar elde ediniminden sonra destek, eğitim, belgeleme, bakım vb. maliyetlere sahip olacaktır.

Avustralya merkezli Cybersource firması [21] "Linux Windows'a karşı: son söz" çalışmasında Microsoft tarafından pazarlanan yazılımlar yerine benzer işlevleri yapan AKY'ların kullanılması ile sağlanabilecek tasarrufu analiz etmiştir [15]. Bu karşılaştırmayı ilk elde edinim maliyeti üzerinden 50 çalışanı olan bir şirket için uygulamıştır. Çizelge 2.1'de Microsoft'un pazarladığı yazılımların maliyeti, Çizelge 2.2'de ise bu yazılımların alternatifleri olan AKY'ın maliyeti verilmiştir [15]. Bu karşılaştırmada standart ofis yazılımları, e-posta, İtranet ve İnternet hizmetleri ile veri tabanı erişimi öngörülmüştür.

Çizelge 2.1. Microsoft çözümlerinin yazılım maliyeti [15]

Yazılım	Lisans Adedi	Fiyatı
Norton Anti virüs 2002	50	\$2,498.00
MS Internet Information Server 5	2	\$0.00
MS Windows 2000 Advanced Server	5	\$19,995.00
MS Commerce Server	1	\$12,333.00
MS ISA Standard Server 2000	1	\$1,499.00
MS SQL Server	1	\$4,999.00
MS Exchange Standard Server 2000	1	\$1,299.00
Windows XP Professional	50	\$14,950.00
MS Visual Studio 6.0	3	\$3,237.00
MS Office Standard	50	\$23,950.00
Adobe Photoshop 6	2	\$1,218.00
Ek Kullanıcı Erişim Lisansları	30	\$2,010.00
Toplam		\$87,988.00

Çizelge 2.2. AKY çözümün yazılım maliyeti [15]

Yazılım	Lisans Adedi	Fiyatı
GNU/Linux Dağıtımı (SuSE 7.3)	1 adet yeterli	\$79.95
Apache Web Sunucusu	Kısıtlama yok	\$0.00
Squid Proxy Sunucu	Kısıtlama yok	\$0.00
PostgreSQL Veritabanı	Kısıtlama yok	\$0.00
iptables Güvenlik Duvarı	Kısıtlama yok	\$0.00
Sendmail/Postfix E-posta Sunucuları	Kısıtlama yok	\$0.00
Kdevelop Tümlleşik Geliştirme Ortamı	Kısıtlama yok	\$0.00
GIMP Grafik İşleme Yazılımı	Kısıtlama yok	\$0.00
Openoffice.org Ofis Paketi	Kısıtlama yok	\$0.00
OSCommerce e-ticaret paketi	Kısıtlama yok	\$0.00
Toplam		\$79.95

Çizelge 2.3. Microsoft karşısında AKY tasarrufları [15]

	Microsoft'un Çözümü	AKY Yazılım Çözümü	Tasarruf Miktarı
A Şirketi: 50 çalışanlı	\$87,988.00	\$79.95	\$87,908.05
B Şirketi: 100 çalışanlı	\$136,734.00	\$79.95	\$136,654.05
C Şirketi: 250 çalışanlı	\$282,974.00	\$79.95	\$282,894.05

Çizelge 2.3'te, Çizelge 2.1 ve Çizelge 2.2'de elde edilen sonuçların 100 ve 250 çalışanı olan şirketlere uygulanış maliyetleri verilmiştir.

Çizelge 2.1 ve Çizelge 2.2'ye baktığımızda AKY ile ticari yazılımların ilk elde edinim fiyatları arasında büyük ölçüde fark olduğu görülmektedir. Ayrıca Çizelge 2.3'e baktığımızda AKY kullanan şirketlerde şirket ne kadar büyüye de ilk elde edinme maliyeti değişmemektedir.

Yazılımların ilk elde edinim maliyeti bakımından AKY'ların açık ara üstünlüğü vardır. Bu üstünlük destek maliyetlerinde de göze çarpmaktadır. AKY destek hizmetleri veren şirketler piyasada vardır ve bu destek birimlerinin fiyatları rekabete açıktır. Aynı zamanda destek hizmeti beğenilmediği durumlarda kullanıcı özgürce destek hizmeti veren şirketi değiştirme hakkına sahiptir. Bu durum ticari yazılımlarda farklıdır ve destek hizmeti beğenilmese dahi satıcı firmadan destek almak zorunludur.

Yukarıda bahsedildiği gibi AKY'lar bu ve bunlar gibi birçok sebepten dolayı kullanıcılar tarafından daha tercih edilir pozisyondadır. Black Duck software [2] ve North Bridge Venture [3] organizasyonlarının ortak sunduğu bir raporda, Şekil 2.2'de görüldüğü gibi, 2011 yılından önce kullanıcıların satıcı firmaya bağlı kalmamak için AKY'ları tercih etmeleri birinci sırada yer alıyordu. Çünkü yazılımlar satın alınırken kullanıcı ile satıcı firma arasında kullanıcıyı o yazılımı belli bir süre kullanması için zorlayan bazı anlaşmalar imzalanır ve kullanıcı yazılımın bazı yönlerinden memnun olmasa bile o süre zarfında anlaşmaya uymak zorunda kalır. Yine 2011 yılında AKY'ların kullanıcılar tarafından daha ucuz ve bakım maliyetinin daha düşük olması sebebiyle seçilmiş olması ikinci sırada ve AKY'ların daha kaliteli yazılımlar olduğu için kullanıcılar tarafından seçilmesi son sırada yer almıştır. 2012 yılına gelindiğinde AKY'ların kullanıcılar tarafından kaliteli olmasından dolayı tercih edilmesi üçüncü sıraya gelmiş ve satıcı firmaya olan bağımlılıklarının olmaması yine ilk sırada yer almıştır. 2013 yılı ve sonrasında açık kaynak yazılımların kullanıcılar tarafından tercih edilmesinin birinci sebebi ise bu yazılımların daha iyi yazılım kalitesine sahip olmuş olmasıdır.



Şekil 2.2 AKY'ların tercih edilme sebebinin yıllara göre değişimi [2][3]

2.1.4. Açık Kaynak Yazılım Lisansları

AKY lisansları hiç kimseye bağlı kalmaksızın izin istemeden kullanıcıya yazılım ürününü değiştirme, kopyalama veya dağıtılma gibi imkânlar sunan yetkililerdir. Lisansın türüne göre bu yetkiler farklılık gösterir. Bünyesinde AKY ürünleri barındıran kullanıcıların, ürünlerin AKY lisansına sahip olup olmadığı konusunda dikkatli olmaları gerekir. Aksi halde çok büyük yaptırımlarla karşılaşabilirler. Standartlara uygun AKY lisans sayıları 2017 yılı itibariyle 80'ne ulaşmıştır [13]. Bunların en önemlileri ve en çok kullanılanları Çizelge 2.4'te verilmiştir.

Çizelge 2.4. AKY lisansları [22]

Lisanslar	Yapılabilecekler	Kısıtlar	Zorunlu olanlar
Genel Kamu lisansı (General Public Licence- GPL)	*Ticari olarak kullanılabilir. *Değiştirilebilir. *Değiştirebilirsiniz. *Dağıtılabilirsiniz. *Yazar sorumlu tutulamaz.	*Lisanslanamaz.	*Orijinali bulundurulmalı.
MIT (Massachusetts Institute of Technology lisansı	*Ticari olarak kullanılabilir. *Değiştirilebilir. *Dağıtılabilir. *Lisanslanabilir. *Hususi Kullanılabilir.	*Yazar sorumlu tutulamaz.	*Telif bulundurulmalı. *Orijinali bulundurulmalı.
Apache Lisansı	*Ticari olarak kullanılabilir. *Değiştirilebilir. *Dağıtılabilir. *Lisanslanabilir. *Hususi Kullanılabilir.	*Yazar sorumlu tutulamaz. *Yazar isimleri kullanılamaz.	*Telif bulundurulmalı. *Lisansı bulundurulmalı. *Değişiklikleri belirlenmeli. *Değişiklikler bildirilmeli.
Berkeley Yazılım Dağıtma (Berkeley Software Distribution-BSD) Lisansı	*Ticari olarak kullanılabilir. *Değiştirilebilir. *Dağıtılabilir. *Yazar sorumlu tutulamaz.	*Yazar sorumlu tutulamaz. *Yazar sorumlu tutulamaz.	
CC (Creative Common) Lisansı	*Ticari olarak kullanılabilir. *Değiştirilebilir. *Dağıtılabilir.		

2.2. Yazılım Kalite Modelleri

Kalite kavramı kullanıcıların beklentisine göre değişkenlik gösteren ve tanımı sürekli tartışılan bir kavramdır. En genel tanımı ile bir ürün ihtiyaç ve beklentileri ne derece karşılıyorsa o kadar kalitelidir diyebiliriz. Yazılım kalitesi ise örtük (İng. implicit) ve açık (İng. explicit) gereksinimlerin karşılanma derecesi olarak tanımlanabilir [23].

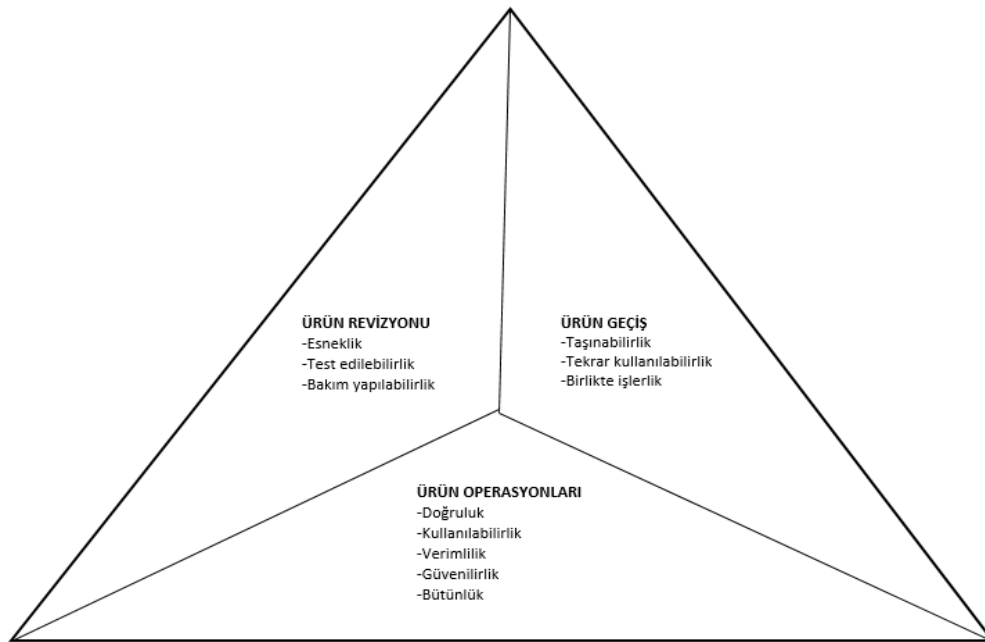
Yazılım kalite modeli, kalite gereksinimlerinin belirlenmesinde ve bir bileşenin veya sistemin kalitesinin değerlendirilmesinde yol gösterici öznelilikler kümesidir. Bu öznelilikler yazılım kalitesi ile ilişkilidir ve ölçülebilmesi için uygun metrikler kullanılması gerekir. Geçmişten günümüze birçok kalite modeli önerilmiştir. Bunlardan en bilinen olanları şunlardır: McCall Kalite Modeli (1977), Boehm Kalite Modeli (1978), FURPS Kalite Modeli (1992), Dromey Kalite Modeli (1995), ISO/IEC 9126 Kalite Modeli (2001).

2.2.1. McCall Kalite Modeli (1977)

Yazılım kalite modellerinin en ünlü örneklerinden biri, Jim McCall ve arkadaşları tarafından sunulan kalite modelidir [24]. Bu model öncelikle sistem geliştiricilerine ve sistem geliştirme sürecine yöneliktir. McCall modelinde, hem kullanıcıların görüşlerini hem de geliştiricilerin önceliklerini yansıtacak bir takım yazılım kalitesi faktörlerine odaklanarak, kullanıcılar ve geliştiriciler arasındaki boşluğu kapatmaya çalışmıştır.

McCall kalite modeli, Şekil 2.3'te gösterildiği gibi, bir yazılım ürününün kalitesini tanımlamak için üç temel perspektife sahiptir:

- Ürün revizyonu (İng. product revision): değişimden geçme kabiliyeti
- Ürün geçiş (İng. product transition): yeni çevreye uyarlanma
- Ürün operasyonları (İng. product operations): işletme karakteristikleri.



Şekil 2.3 McCall kalite üçgeni [25]

-Ürün revizyonu: Yazılım ürününü değiştirme kabiliyetini etkileyen kalite faktörlerini tanımlar. Bu faktörler şunlardır:

- Bakım yapılabilirlik (İng. maintainability): Programdaki hataların (İng. defect) yerlerini tespit etmek ve çözmek için harcanan efordur.
- Esneklik (İng. flexibility): Çalışma ortamının değişikliğine göre gerekli değişiklikleri yapma kolaylığıdır.
- Test edilebilirlik (İng. testability): Programın test edilmesinin kolaylığı; hatasız olduğundan ve şartnameye uygun olduğundan emin olunmasıdır.

-Ürün geçişi: Yazılımı yeni ortamlara uyarlama becerisini etkileyen kalite faktörlerini tanımlar.

- Taşınabilirlik (İng. portability): Yazılım ürünü bir ortamdan başka bir platforma taşınırken gereken efordur.
- Tekrar kullanılabilirlik (İng. reusability): Yazılım bileşenlerinin farklı içeriklerde kullanım kolaylığıdır.
- Birlikte işlerlik (İng. interoperability): Bir sistemi başka bir sistem ile birleştirmek için gösterilen çabadır.

-Ürün operasyonları: Yazılımın spesifikasyonunu ne ölçüde yerine getirdiğini etkileyen kalite faktörlerini tanımlar.

- Doğruluk (İng. validity): Bir programın spesifikasyonunu ne ölçüde yerine getirdiğidir.
- Kullanılabilirlik (İng. usability): Kullanım kolaylığıdır.
- Verimlilik (İng. efficiency): Sistem kaynağı (disk, bellek, ağ, vb.) kullanımınıdır.
- Güvenilirlik (İng. reliability): Sistemlerin başarısızlığa uğramamasıdır.
- Bütünlük (İng. integrity): Programın yetkisiz erişime karşı korunmasıdır.

Yukarıda 3 ana perspektif içinde verilen 11 faktör yazılımın dış görünümünü (İng. external view) tanımlar. Çizelge 2.5'te verilen bu 11 faktörle ilişkili her bir kalite kriteri, yazılımın iç görünümünü (İng. internal view) tanımlar.

Çizelge 2.5. McCall kalite modeli faktörleri [24]

Faktörler	Kalite kriterleri
Doğruluk	İzlenile-bilirlik, bütünlük, tutarlılık
Güvenilirlik	Tutarlılık, kesinlik, hata toleransı
Verimlilik	Çalışma verimliliği, depolama verimliliği
Bütünlük	Erişim kontrolü, erişim denetimi
Kullanılabilirlik	İşletile-bilirlik, alıştırma (eğitim), konuşkanlık
Bakım yapılabilirlik	Basitlik, özlülük, kendini tanımlama, modülerlik
Test edilebilirlik	Basitlik, enstrümantasyon, kendini tanımlama, modülerlik
Esneklik	Kendini tanımlama, genellik, genişletile-bilirlik
Taşınabilirlik	Yazılım-sistem bağımsızlığı, makine bağımsızlığı, kendini-tanımlama
Tekrar kullanılabilirlik	Kendini tanımlama, genellik, modülerlik, makine bağımsızlığı
Birlikte-işlerlik	Modülerlik, iletişim ortaklığı, veri ortaklığı

2.2.2. Boehm Kalite Modeli

Yazılım kalite modellerinin temeli ve kurucusu olan ikinci model Barry W. Boehm tarafından 1978 yılında önerilmiştir. Boehm yazılım kalitesini otomatik ve nicel olarak ölçen modellerin eksiklerini ele almıştır. Boehm modeli yazılım kalitesini verilen metrik ve özniteliklerle nitel olarak tanımlamaya çalışır. Boehm kalite modeli, her biri genel kalite seviyesine katkıda bulunan üç hiyerarşik katmandan oluşur (Çizelge 2.6). Bu özelliği ile McCall kalite modeli ile benzerlik gösterir. Bu katmanlar:

-Üst düzey özellikler (İng. high-level characteristics): Üst düzey özellikler, bir yazılım alıcısının sahip olduğu üç temel soruyu ele alır:

- Kullanım (İng. as-is utility): Ne kadar iyi (kolay, güvenilir ve verimli) kullanabilirim?
- Bakım-yapılabilirlik: Anlamak, değiştirmek ve yeniden test etmek ne kadar kolay?
- Taşınabilirlik: Ortamını değiştirsem de kullanabilir miyim?

-Orta seviye özellikler (İng. intermediate level characteristics): Bir yazılım sisteminden beklenen nitelikleri birlikte temsil eden 7 kalite faktöründen oluşur.

- Taşınabilirlik: Kod mevcut olan bilgisayarın dışındaki bilgisayarlarda da kolayca ve sorunsuz çalışabilmesi için taşınabilirliğe ne derece sahiptir?
- Güvenilirlik: Kod beklenen işlevleri tatminkâr bir şekilde yerine getirmede ne derece güvenilirliğe sahiptir?
- Verimlilik: Kod mevcut kaynakları israf etmeden amacını ne derece yerine getiriyor?

- Kullanılabilirlik: Kod ne derece güvenilir, verimli ve kolay bir şekilde kullanıma sahiptir?
- Test-edilebilirlik: Kod ne derece test edilebilir ve doğrulanabilir özelliğe sahiptir?
- Anlaşılabilirlik (İng. understandability): Kod ne derece anlaşılabilirliğe sahiptir?
- Esneklik: İstenilen değişim niteliği belirlendikten sonra, kaynak koddaki değişimler ne derece kolay yapılıyor?

-İlkel özellikler (İng. primitive characteristics): Orta seviye özelliklerinde bahsedilen 7 kalite faktörleri bu en alt tabaka ile ölçülebilen (Çizelge 2.6) ilkel yapılara ayrılmıştır.

Çizelge 2.6. Boehm kalite modeli ve katmanları [24]

Genel kullanım (General Utility)							
Üst düzey özellikler	Kullanım (As-is utility)			Bakım yapılabilirlik			Taşınabilirlik
	Güvenilirlik	Verimlilik	Kullanılabilirlik	Test edilebilirlik	Anlaşılabilirlik	Esneklik	
İlkel özellikler	Kendine bağlılık, kesinlik, eksiksizlik, bütünlük, tutarlılık	Hesap verilebilirlik, cihaz verimliliği, erişilebilirlik	Bütünlük, erişilebilirlik, iletişim	Hesap verilebilirlik, iletişim, öztamamlayıcılık, yapısalılık	Tutarlılık, yapısalılık, özlülük, okunaklılık	Yapısalılık, genişletilebilirlik	Aygıt bağımsızlığı, kendine bağlılık,

McCall ve Boehm kalite modelleri birbirleriyle benzerlik gösterse de Boehm kalite modeli McCall tarafından önerilen kalite modelinden bazı öznelikleri çıkarıp Çizelge 2.7’de gösterildiği gibi yenilerini eklemiştir.

Çizelge 2.7. McCall kalite modeli ile Boehm kalite modeli karşılaştırılması

Kriter	McCall	Boehm
Doğruluk (Correctness)	x	x
Güvenilirlik (Reliability)	x	x
Bütünlük (integrity)	x	x
Kullanılabilirlik (Usability)	x	x
Verimlilik (Efficiency)	x	x
Bakım yapılabilirlik (Maintainability)	x	x
Test edilebilirlik (Testability)	x	
Birlikte işlerlik (Interoperability)	x	
Esneklik (Flexibility)	x	
Tekrar kullanılabilirlik (Reusability)	x	x
Taşınabilirlik (Portability)	x	x
Açıklık (Clarity)		x
Değiştirilebilirlik (Modifiability)		x
Belgelilik (Documentation)		x
Esneklik (Resilience)		x
Anlaşılabilirlik (Understandability)		x
Geçerlilik (Validity)		x
İşlevsellik (Functionality)		
Genellik (Generality)		x
Ekonomi (Economy)		x

2.2.3. FURPS Kalite Modeli

FURPS kalite modeli 1992 yılında Robert Grady tarafından [26] önerilmiştir. Bu model McCall kalite modelinin birçok kalite faktörünü beş kategoride toplamıştır ve bu kalite faktörlerini Çizelge 2.8’de gösterildiği gibi alt faktörlerle ilişkilendirmiştir.

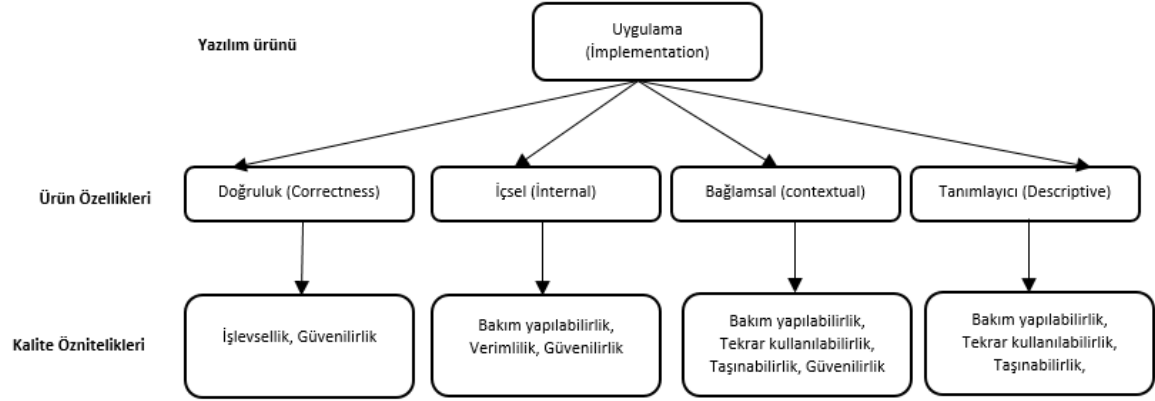
Çizelge 2.8. FURPS kalite modeli [26]

Kalite Faktörleri	İşlevsellik (Functionality)	Kullanılabilirlik (Usability)	Güvenilirlik (Reliability)	Performans (Performance)	Desteklenebilirlik (Supportability)
Alt Kalite Faktörleri	Kapasite, uyumluluk, birlikte işlerlik, taşınabilirlik, güvenlik	İnsan faktörü, Dokümantasyon, Cevaplana-bilirlik	Geçerlilik, Geri kazanılabilirlik, tahmin edilebilirlik, kesinlik	Hız, verimlilik, kapasite, ölçeklenebilirlik	Bakım yapılabilirlik, test edilebilirlik, adapte-olabilirlik, esneklik

2.2.4. Dromey Kalite Modeli

McCall, Boehm ve FURPS kalite modelleriyle benzerlik gösteren daha yeni bir model olan Dromey kalite modeli 1995 yılında R. Geoff Dromey tarafından

önerilmiştir [24][27]. Dromey, kalite değerlendirmesinin her ürün için farklı olduğunu ve süreci modelleme için daha dinamik bir fikrin farklı sistemlere uygulanacak kadar geniş olması gerektiğini tanıyan bir ürün tabanlı kalite modeli önermektedir. Dromey, kalite öznitelikleri ile alt nitelikler arasındaki ilişkiye odaklanmanın yanı sıra, yazılım ürün özelliklerini yazılım kalitesi nitelikleriyle birleştirmeye çalışmıştır. Dromey kalite modeli Şekil 2.4'de gösterilmiştir.



Şekil 2.4 Dromey kalite modeli [24]

2.2.5. ISO/IEC 9126 Kalite Modeli

ISO/IEC 9126 kalite modeli McCall ve Boehm kalite modellerini esas almıştır. Temel olarak benzer olmakla birlikte bu kalite modelinden farklı olarak, işlevsellik (İng. functionality) faktörünü eklenmiştir. Ayrıca bu kalite modeli yazılım ürünlerinin özelliklerini hem içsel (İng. internal) hem de dışsal (İng. external) olarak tanımlamıştır. ISO/IEC 9126 kalite modeli 6 temel nitelik tanımlayarak ürünlerin kalitesini ifade etmiştir. Bu nitelikleri toplamda 27 alt nitelikle ilişkilendirmiştir. Şekil 2.5'de gösterilen niteliklerin ve alt niteliklerin tanımları Çizelge 2.9'da verilmiştir.



Şekil 2.5 ISO/IEC 9126 kalite modeli [28]

Çizelge 2.9. ISO/IEC 9126 kalite modeli niteliklerinin ve alt niteliklerinin tanımları
[29]

Nitelikler ve alt-nitelikler	Tanım
İşlevsellik	Yazılım ürününün belirli şartlar altında istenilen fonksiyonel özellikleri sağlama
Uygunluk	Yazılımların istenilen fonksiyon özelliklerine sahip olması ve bu fonksiyonlara uygunluğu
Kesinlik	Yazılımların anlaşma doğrultusunda doğru etkiyi veya sonucu vermesi
Birlikte çalışabilirlik	Yazılımların istenilen sistemlerle birlikte çalışabilme becerisi
Fonksiyonel uygunluk	Yazılımların uygun standartlara, anlaşmalara, yasal düzenlemelere ve talimatlara uygun olması
Güvenlik	Yazılımlardaki verilere yetkilendirilmemiş erişimi önleyebilme özelliği
Güvenilirlik	Yazılımın ilk kez karşılaştığı durumlarda performans seviyesini koruyabilme becerisi
Olgunluk	Yazılımlardaki beklenmeyen sebeplerle yazılımın çökme sıklığının düşük olması özelliği
Hata toleransı	Yazılımın başarısızlığa dayanma (ve kurtarma) yeteneği.
Kurtarılabirlik	Sistemin işlevlerini yerine getirememesi durumunda etkilenen verinin geri yüklenebilmesi ve performans seviyesinin korunabilmesi
Kullanılabilirlik	Belirlenmiş şartlar da kolayca anlaşılabilme, öğrenilebilme ve kullanıcıyı etkileyebilme yeteneği
Anlaşılabilirlik	Sistem işlevlerinin ne kadar kolay anlaşılacağını belirler
Öğrenilebilirlik	Yazılım uygulamalarının kullanıcı tarafından öğrenilebilmesi için harcanacak kullanıcı çabası
İşletilebilirlik	Yazılım operasyonları ve operasyonların kontrolü için gereken kullanıcı çabası
Dikkat çekici olma	Sistem arayüzünün ne kadar dikkat çekici olduğunu gösterir
Kullanım Uygunluğu	Sistemin ne kadar kullanıma yakın olduğunu gösterir.
Verimlilik	Belirlenen şartlara kaynağa bağlı olmak şartıyla yazılımın belirlenen performansı göstermesi
Süreye karşı davranış	Yazılımda belirlenen komutlara verilen cevap sürelerini temsil eder.
Kaynak kullanımı	Yazılımın kullanım süresi boyunca fonksiyonları yerine getirirken kullanılan kaynak miktarı ile ilgilenen yazılım özellikleri
Bakım yapılabilirlik	Gereksinimler doğrultusunda yazılımda geliştirilebilme ve değiştirilebilme becerisi
Çözümenebilirlik	Yazılımda meydana gelen bir hatanın ana sebebini belirleme yeteneğini temsil eder
Değiştirilebilirlik	Bir sistemde yapılan değişiklik için gerekli çaba miktarını temsil eder
Kararlılık	Yapılan değişikliklerle beraber yazılımın beklenmeyen etki doğurması riskini temsil eder
Test edilebilirlik	Sistem değişikliğini doğrulamak (test etmek) için gerekli çabayı temsil eder
Bakım Uygunluğu	Sistemin değişikliklere uygunluğunu temsil eder.
Taşınabilirlik	Yazılımın çalıştığı ortamdan başka bir ortama adapte olabilmesi
Uyarılabilirlik	Belirlenen amacın sağlanabilmesinde değişiklik gerçekleştirilmeden yazılımın farklı çevrede adapte edilebilme özellikleri
Kurulabilirlik	Yazılımın kurulumu için gerekli eforu temsil eder
Değiştirilebilirlik	Taşınabilirlik ile ilgili anlaşmalar veya standartlara bağlı yazılım geliştirme becerisi
Taşınabilirlik	Yazılımın başka bir ortamda işlevlerini yapması için harcanması gereken çaba
Birlikte çalışabilirlik	Yazılımların istenilen sistemlerle birlikte çalışabilme becerisi

2.2.6. ISO/IEC 25010 Kalite Modeli

ISO/IEC 25010 kalite modeli [30] yazılım ürününün kalitesini paydaşların gereksinimlerini karşılama derecesi olarak tanımlar. Buradaki gereksinimler hem daha önce belirtilmiş olanlar (İng. stated) hem de doğal olarak olması beklenen (İng. implied) gereksinimlerdir. Model bu gereksinimleri belirlediği öznitelikler ve alt özniteliklerden oluşan hiyerarşik bir yapıyla tanımlar. Bu hiyerarşik modeli önerirken daha eski bir kalite modeli olan ISO/IEC 9126'yı temel almıştır. 6 tane ana özniteliği olan ISO/IEC 9126 kalite modelini güncelleyerek ve yeni özellikler ekleyerek Çizelge

2.10'da gösterildiği gibi 8 ana öznitelikten oluşan yeni bir kalite modeli önermiştir. Önerilen bu kalite modeli kullanımdaki kalite modeli (İng. quality in use) ve ürün kalite modeli (İng. product quality) olmak üzere ikiye ayrılmaktadır.

Çizelge 2.10. ISO/IEC 25010 kalite modeli öznitelikleri ve alt-öznitelikleri [30]

Öznitelikler	Alt öznitelikler
Fonksiyonel uygunluk	Uygunluk, doğruluk, birlikte işlerlik, güvenlik, uyumluluk
Güvenilirlik	Olgunluk, hata dayanıklılığı, geri kazanılabilirlik, uyumluluk
Performans verimliliği	Zamana göre davranış durumu, kaynak kullanımı, uyumluluk
İşletilebilirlik	Uygunluk, tanınabilirlik, kullanım kolaylığı, öğrenilebilirlik, çekicilik, teknik ulaşılabilirlik, uyumluluk
Güvenlik	Gizlilik, bütünlük, inkâr edilememe, hesap verebilirlik, aslına uygunluk, uyumluluk
Uyumluluk	Değiştirilebilirlik, birlikte var olabilme, birlikte işlerlik, uyumluluk
Bakım yapılabilirlik	Modülerlik, tekrar kullanılabilirlik, çözümlenebilirlik, değişebilirlik, değiştirilebilme kararlılığı, sınanabilirlik, uyumluluk
Aktarılabilirlik	Taşınabilirlik, adapte olabilirlik, yüklenebilirlik, uyumluluk

Bu tez kapsamımda en güncel kalite modeli olan ISO/IEC 25010'nun kalite modeli öznitelikleri ve alt öznitelikleri dikkate alınmış ve güvenilirlik ve bakım yapılabilirlik öznitelikleri AKY seçimi için temel alınmıştır.

2.3. Yazılım Ölçüm Çatıları (İng. Software Measurement Frameworks)

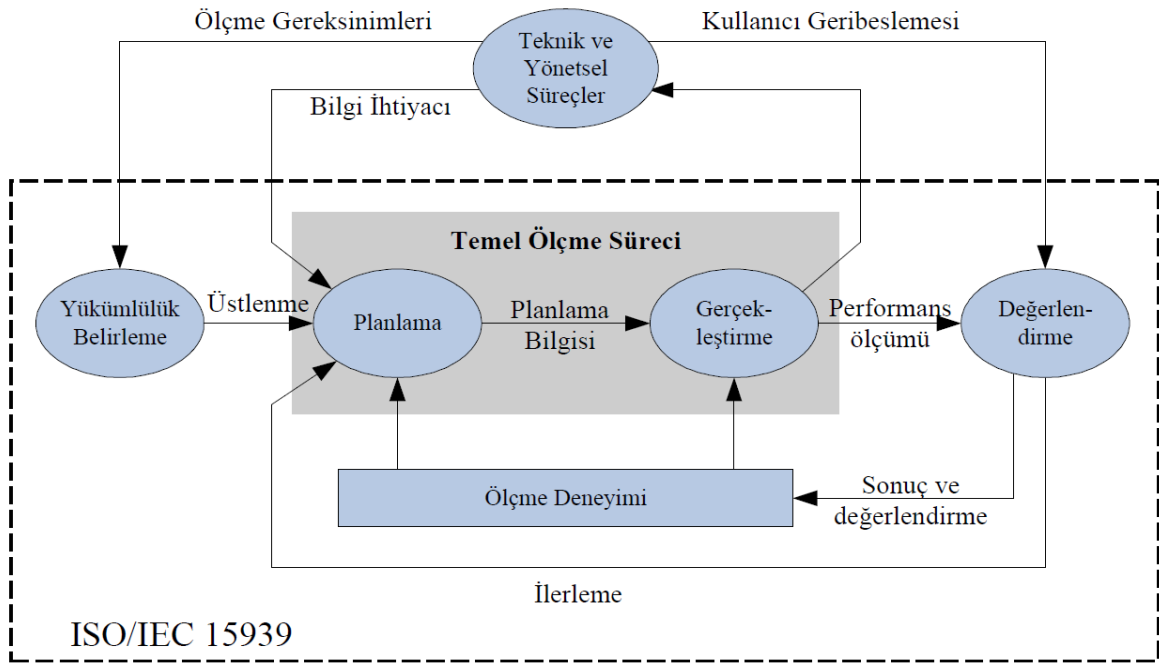
Bu tez çalışmasında iki boyutlu değerlendirme önerilmiş ve her boyutun değerlendirmesi için farklı yollar izlenmiştir. Kod-tabanlı değerlendirmede kalite özniteliklerinin ve ölçüm yapılacak metriklerinin belirlenmesi için araştırma soruları hazırlamıştır. Toplum-tabanlı değerlendirme yapmak için ise kullanıcının bilgi ihtiyaçlarının (İng. information needs) belirlenmesinden ölçüm sonuçlarının elde edilmesine kadarki süreçte, ISO/IEC 15939: Sistem ve Yazılım Mühendisliği-Ölçüm Süreci [31] uluslararası ölçüm standardı takip edilmiştir. İzleyen alt bölümde bu ölçüm süreci açıklanmıştır.

2.3.1. ISO/IEC 15939 Yazılım Ölçüm Süreci

ISO/IEC 15939 sistem mühendisliği, yazılım mühendisliği ve yönetim disiplinleri için geçerli olan bir ölçüm sürecini tanımlar [29]. Süreç, hangi ölçüm bilgisinin gerekli olduğunu, ölçümlerin ve analiz sonuçlarının nasıl uygulanacağını ve analiz

sonuçlarının geçerli olup olmadığını tanımlayan bir ölçüm süreci modelidir. Kalite bakımından ürünün değerlendirilmesinde, ilgili verilerin toplanmasında ve sonuçların ihtiyaca uygun değerlendirilmesinde bize yol gösterici bir standarttır.

Planla-Uygula-Denetle-Harekete geç (İng. Plan-Do-Check-Act) standart temel kalite geliştirme sürecine uygun olarak, Şekil 2.6'de gösterildiği gibi dört ana aşamadan oluşur.

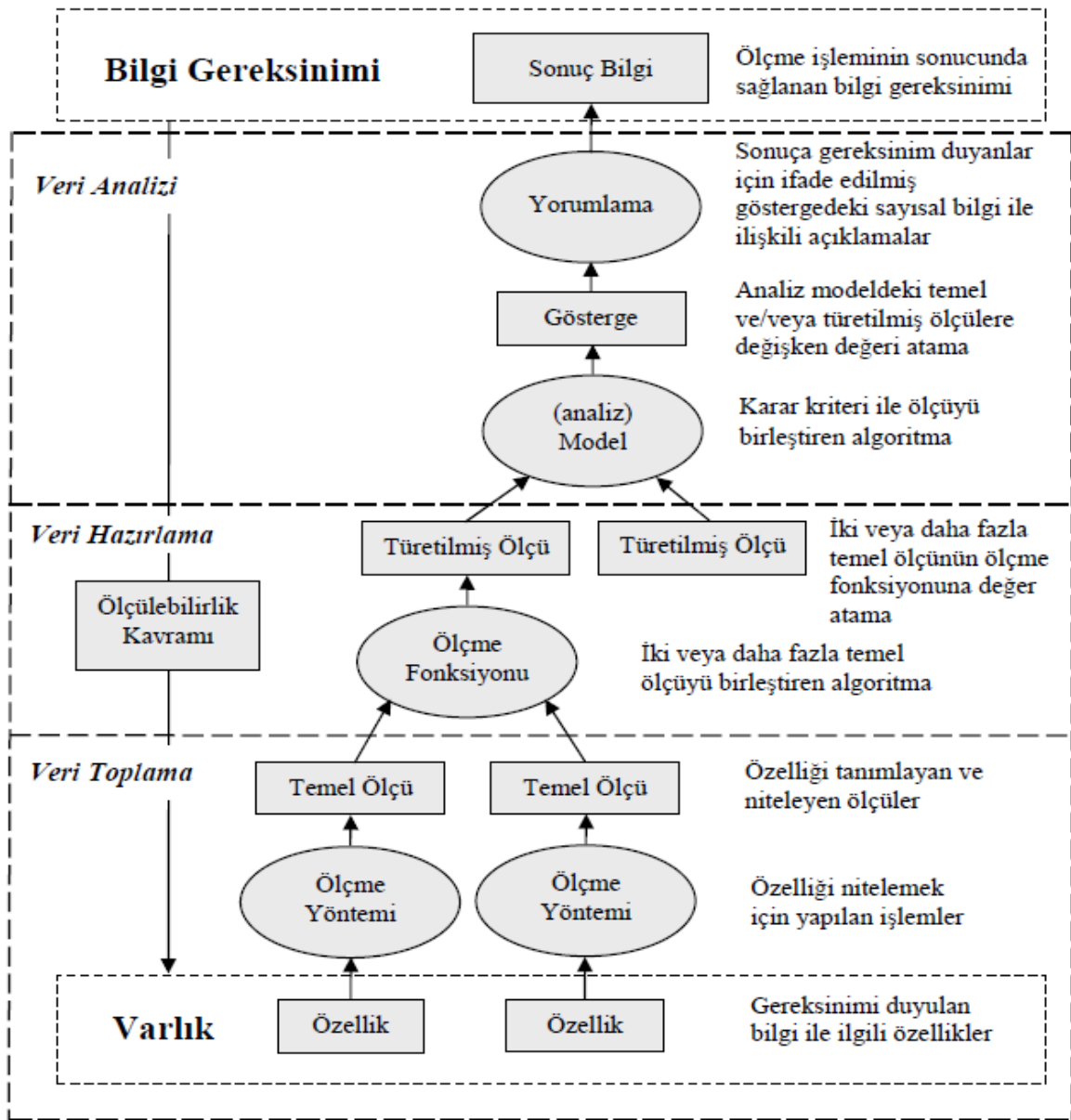


Şekil 2.6 ISO/IEC 15939 ölçüm sürecinin aşamaları [29]

1. **Yükümlülük belirleme:** Bu aşamada ölçmenin amacının tanımlanması ve yönetim isteklerine göre sorumlulukların tespit edilmesi, daha sonra da ölçmeye ilişkin görevlerin ve kaynakların atanması yer alır.
2. **Ölçme sürecinin planlanması:** Bu aşamada bilgi ihtiyacının belirlenmesi, bu ihtiyaçların öneme göre sıralanması, hangi metriklere göre değerlendirileceği ve ölçme için kullanılacak araçların elde edilmesi bulunmaktadır.
3. **Ölçme sürecinin gerçekleştirilmesi:** Bu aşamada ihtiyaçlar doğrultusunda verilerin toplanması, toplanan verilerin doğrulanması, verilerin analizi, analiz sonuçlarının belgelendirilmesi ve süreçten yararlanan kişiler arasındaki iletişim içerilir.

4. **Sonuçların değerlendirilmesi:** Bu aşamada ölçme süreci genel olarak değerlendirilir.

ISO/IEC 15939 ölçme sürecindeki kavramların temel ilişkileri Şekil 2.7’de gösterilmiştir. Şekilde ölçülmek istenen bilgiyi sağlayan varlığa ait özelliklerden başlayarak, bir dizi etkinliğin nasıl modellendiği anlatılmaktadır. Varlığın özelliğini nitelenmek için temel ölçülere (metriklere) değer atanmasını takiben, temel ölçülerden ölçme fonksiyonu sayesinde türetilen ölçüler elde edilmekte ve ölçüm gerçekleştirilmektedir. Sonrasında ise elde edilen sayısal ölçüm bilgilerine ilişkin yorumlama yapılarak, bilgi gereksinimi karşılanmaktadır [29].



Şekil 2.7 ISO/IEC 15939 ölçüm modelindeki temel kavramlar ve ilişkileri [29]

2.4. Nesneye Dayalı Yazılım Metrikleri

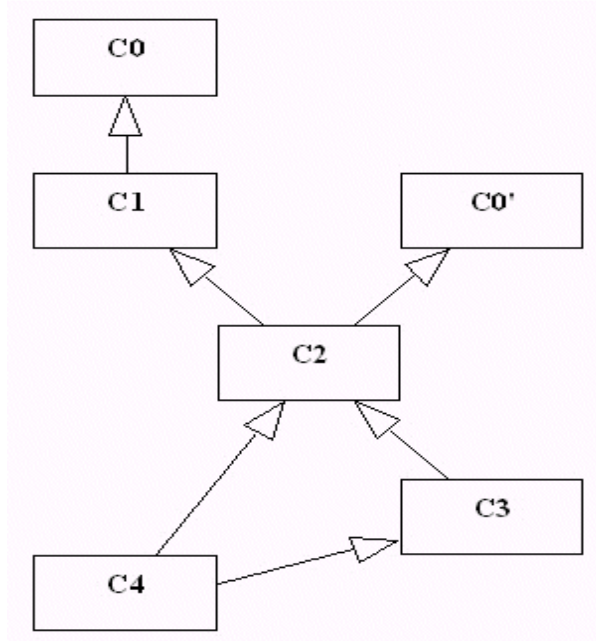
Yazılımın kalitesini kontrol altında tutmak ve bu doğrultuda geliştirmek için ölçüm yapmak önemlidir. Herkesin yazılımdan beklentisinin farklı olacağı düşünüldüğünde bu beklentilerin belirlenip nasıl ölçüleceği bu alandaki en büyük problemlerdendir. Yazılım metrikleri ölçüm sonucunda sayısal değerler vererek yazılım kalitesinin ölçümünde öznel değerlendirmelerden kurtulmamızı sağlar. Goodman [32] yazılım metriklerini *“ölçüm temelli tekniklerin yazılım geliştirme sürecine sürekli uygulanması, süreç ve (onun) ürünlerinin gelişmesi için bu tekniklerin kullanılmasıyla birlikte, anlamlı ve zamanında bilgi yönetimini sağlaması”* ve Schulmeyer [33] ise *“bir sistemin, bileşenin veya sürecin verilen özelliğe sahip olma derecesinin nicel ölçümü”* olarak tanımlamıştır. Yazılım dünyasında nesneye yönelik geliştirilen programların ölçümünü yapmak ve bireysel 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 [34] ve bu tez çalışmasında kod-tabanlı boyutta değerlendirmeye temel alınmıştır.

Tez çalışmasında bakım yapılabilirlik ve güvenilirlik ölçümünde kullanılan metriklerin tamamı izleyen alt bölümlerde açıklanmıştır.

2.4.1. Kalıtım Ağacının Derinliği (İng. Depth Of Inheritance Tree-DIT)

Sınıfın kalıtım ağacının köküne uzaklığını ölçer [35]. Bir sınıf her hangi bir sınıftan türetilmemiş ise değeri 0'dır. Şekil 2.8'da gösterildiği gibi kalıtım ağacının derinliği en uzak köke olan uzaklıktır. Şekilde her bir sınıf için kalıtım ağacının derinliği şöyledir;

$DIT(C0) = 0$, $DIT(C0') = 0$, $DIT(C1) = 1$, $DIT(C2) = 2$, $DIT(C3) = 3$, $DIT(C4) = 4$



Şekil 2.8 Kalıtım ağacının derinliği [36]

Şekilde görüldüğü gibi kalıtım ağacının derinliğinin 0 il 4 arasında olması idealdir [36]. Kalıtım ağacının derininde olan sınıflarda türetilen metot sayısı çok olacağından yazılımın davranışını kontrol etmek zorlaşır ve tasarım karmaşıklığı oluşur. Bu metrik değerinin fazla olması değişebilirlik ve kararlılığı olumsuz etkiler. Aynı zamanda güvenilirlik ile bu metrik değeri arasında ters orantılı bir ilişki vardır [37][38][39][40].

2.4.2. Alt Sınıf Sayısı (İng. Number of Children-NOC)

Bir sınıftan direkt türetilmiş alt sınıf sayısını ölçer [34]. Alt sınıf sayısının çok fazla olması kalıtımın yanlış kullanıldığının göstergesidir. Bu metrik değerinin düşük olması istenir. Bu metrik değerinin yüksek olması yeniden kullanımın yüksek olduğu, daha çok hatanın oluşabileceği [34] ve test yapılırken daha çok efor harcanacağından, test edilebilirliğin düşük olacağı anlamına gelmektedir [35]. Dolayısıyla bir sınıfı test etmek için harcanacak bütçe hakkında bilgi verir. Aynı zamanda bu metrik değeri ile güvenilirlik arasında ters bir orantı vardır [37][38][39][40].

2.4.3. Sınıfın Ağırlıklı Metot Sayısı (İng. Weighted Methods Per Class -WMC)

Bu metrik değeri bir sınıftaki metotların karmaşıklık derecesini ve sayısını ölçer [41]. Bir sınıfın metotlarının karmaşıklık derecesi ve sayısı, sınıfın geliştirilmesine ve bakımına harcanacak zaman ve çaba hakkında fikir verir [41].

Eğer bir C sınıfının metotları sırayla $M_1, M_2, M_3, \dots, M_n$ ise ve 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:

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

Bu metrik değerinin düşük olması istenir. Çünkü metotların karmaşıklık derecesi ve sayısı artarsa kodun çözümlenebilirlik süresi de otomatik olarak artacaktır. Aynı zamanda bu metrik değeri ile güvenilirlik arasında ters orantılı bir ilişki vardır [37][38][39][40].

2.4.4. Sınıfın Tetiklediği Metot Sayısı (İng. Response for a Class-RFC)

Bu metriğin değeri, bir sınıfın bir nesnenin metotlarının çağırması durumunda, bu nesnenin tetikleyebileceği tüm metotların sayısıdır. Yani, bir sınıfta yazılan ve çağrılan toplam metot sayısıdır [35]. Bu metrik değerinin büyük olması bir sınıftan çok sayıda metot çağırıldığını gösterir. Bu da kodun anlaşılabilirlik ve test edilebilirliğini düşürür. Hata ayıklaması daha zor olacağından test maliyeti de artar. Şekil 2.9'a bakıldığında verilen kod parçasındaki sınıfta (*PageSecurityService*) kırmızı olarak işaretlenmiş 7 tane metot çağırılmıştır. Dolayısıyla RFC metriğinin değeri 7 olacaktır. Aynı zamanda bu metrik değeri ile güvenilirlik arasında ters orantı vardır.

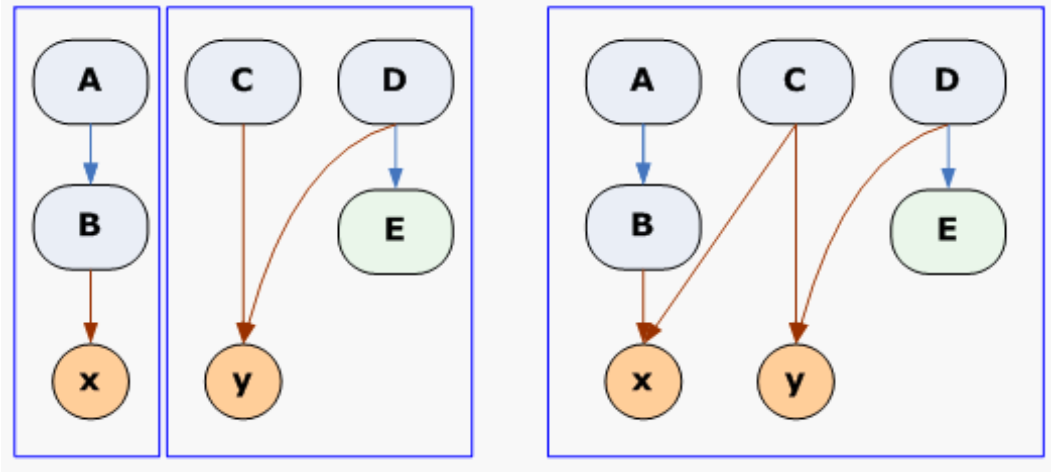
```
class PageSecurityService {
    PageSecurityService(SecurityContext securityContext) { ... }

    boolean hasAccessTo(User user, Page page) {
        return !securityContext.getGlobalLock().isEnabled() &&
            securityContext.getApplicationContext().
                getSecurityDao().userHasPermission(user, page);
    }
}
```

Şekil 2.9 Kod içerisinde çağırılan toplam metot sayısı [42]

2.4.5. Metotlardaki Uyum Eksikliği (İng. Lack of Cohesion in Methods-LCOM)

Bu metrik n adet metot kümesinin kesişiminden oluşan kümelerdeki uyumsuzlukların sayısıdır ve metotlardaki benzerlik derecesini ölçer [41]. Bu metrik değeri fazla ise Şekil 10'da görüldüğü gibi metotlar arasında uyum eksikliği mevcuttur.



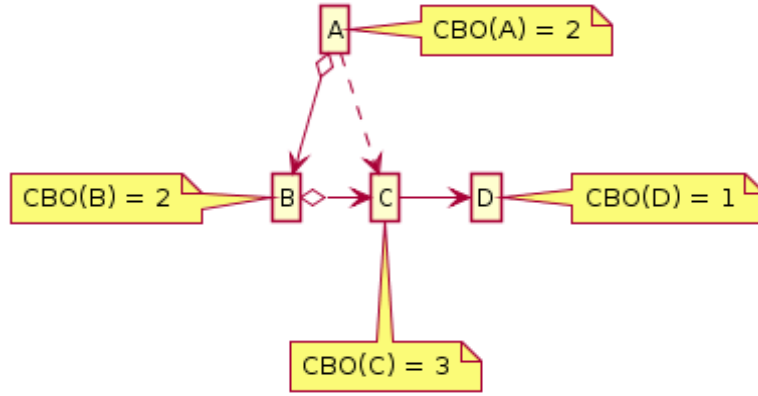
Şekil 2.10 Metotlar arasındaki uyum [43]

Şekilde A'dan E ye metotları olan ve x ve y olmak üzere iki tane değişkeni olan bir sınıf gösterilmiştir. Soldaki şekilde A metodu B'yi çağırıyor ve sadece x değişkenine erişebiliyor. C ve D metotları y değişkenine erişebiliyor ve D metodu E metodunu çağırabiliyor ve E hiçbir değişkene erişemiyor. Bu durumda bu metrik değeri 2 olarak ölçülmüştür. Çünkü sınıf içerisindeki metotlar 2'ye bölünmüştür ve bu iki grup arasında bir uyum yoktur. Sağdaki şekle baktığımızda bu uyumsuzluğu azaltmak için C metodunun x değişkenine eriştiği varsayılmıştır. Bu durumda metotların daha iyi uyum içinde olduğu görülmektedir ve dolayısıyla bu metriğin değeri 1 olarak ölçülmüştür.

Sonuç olarak metotlar arasında uyum yüksek ise bu metrik değerinin düşük olması beklenir. Metotlar arasında uyum düşük olursa karmaşıklık artar ve geliştirme sırasında hata yapma oranı da otomatik olarak artar. Bu durum kodun değişebilirliğini azaltır. Ayrıca bu metrik değeri ile güvenilirlik arasında ters orantılı bir ilişki vardır [37][38][39][40].

2.4.6. Nesne Sınıfları Arasındaki Bağımlılık (İng. Coupling Between Object Classes-CBO)

Bu metrik sınıfın bağımlı olduğu sınıf sayısını ölçer. Bu bağımlılık sınıf içerisindeki bazı özelliklerin veya metotların başka sınıflarda, sınıflar arasında kalıtım olmaksızın kullanılması durumundaki bağımlılıktır [34]. Sınıflar arasındaki bağımlılığın fazla olması, modüler tasarıma zarar verir [35] ve değişebilirliği azaltır. Şekilde A, B, C ve D sınıfları arasındaki bağımlılık gösterilmiştir.



Şekil 2.11 Sınıflar arasındaki bağımlılık [44]

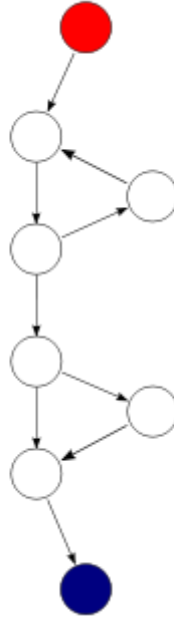
Şekil 2.11’de her bir sınıfın diğer sınıflar arasındaki bağımlılığına bakıldığında; A sınıfı için bu metrik değerinin 2, B sınıfı için 2, C sınıfı için 3 ve D sınıfı için 1 olduğu görülmektedir. Aynı zamanda bu metrik değeri ile güvenilirlik arasında ters orantılı bir ilişki vardır [37][38][39][40].

2.4.7. İç İçe Döngü Sayısı (İng. Number of Nested Loop-NNL)

Bu metrik değeri bir sınıf içerisindeki döngülerin iç içe geçme derinliğini ölçer ve ne kadar büyük bir değer alırsa kodun test edilebilirliği ve kararlılığı azalır.

2.4.8. Çevrimsel Karmaşıklık (İng. Cyclomatic Complexity-CC)

Bu metrik programın kaynak kodunun birbirinden bağımsız olarak takip ettiği yolların sayısını ölçer ve direkt kodun karmaşıklığı ile ilgilidir. Bu metrik değerinin büyük olması istenmeyen bir durumdur ve çözümlenebilirliği azaltır. Şekil2.12’de basit bir programın kontrol akış grafiği gösterilmiştir.



Şekil 2.12 Basit bir programın kontrol akış grafiği [45]

Şekildeki program en üstteki (kırmızı) düğümle başlamış en sondaki (mavi) düğümle sona ermiştir. Bu programın çevrimsel karmaşıklık değeri şu şekilde hesaplanır:

K: Grafikteki köşe sayısı (İng. edge)

D: Grafikteki düğüm sayısı (İng. node)

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

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

Dolayısıyla çevrimsel karmaşıklık değeri $9-8+2*1=3$ olarak hesaplanır.

2.4.9. Açıklama Sayısı (İng. Number of Statement-NOS)

Bu metrik program içerisindeki karmaşıklığı azaltmak adına bize yol gösterecek yorumların ve açıklamaların sıklığını ölçer. Programın takip edilmesini ve çözümlenebilmesini kolaylaştırır. Dolayısıyla bu metrik değerinin büyük olması istenir ve ne kadar büyükse çözümlenebilirlik o kadar fazladır [9].

3. İLGİLİ ÇALIŞMALAR

AKY ürünleri yazılımın kaynak koduyla birlikte kullanıcılara birçok bilgi sunduğu için değerlendirme aşamasında bu bilgilerin kullanılması ve bu bilgiler ışığında en iyi ürünün seçilmesi zor bir süreçtir. Bu yüzden mevcut çalışmalar bu bilgileri kullanmada farklı yollar izlemişlerdir. Bu durum AKY'ların değerlendirilmesi aşamasında bilgi karmaşasına yol açmıştır.

Literatürde AKY'ların değerlendirilmesi ile ilgili çeşitli metotlar ve deneysel (İng. empirical) çalışmalar yer almaktadır. Bu metotların en bilinen olanları; Açık Kaynak Olgunluk Modeli (İng. Open Source Maturity Model-OSMM) [8], Açık Kaynak İş Hazırlık Oranı (İng. Open Business Readiness Rating-OpenBRR) [10], Açık Kaynak Yazılımın Niteliği ve Seçimi (İng. The Qualification and Selection of Open Source software-QSOS) [11], Açık Kaynaklı Yazılımlar için Yazılım Kalite Gözlemi (İng. Software Quality Observatory for Open Source Software-SQO-OSS) [9] modelleridir.

OSMM [8], herhangi bir AKY ürününün organizasyon için uygun olup olmadığını kontrol eden modeldir. Bu model ürünlerin olgunluğunu (İng. maturity) temel alarak ve iki ayrı indikatör kullanarak değerlendirmeyi yapar. Bunlar ürün (İng. product) indikatörü ve uygulama (İng. application) indikatörüdür. Ürün indikatörü değerlendirilen AKY ürününün gelecekteki davranışı ile ilgili yapılan öngörülen sonuçların göstergesidir. Uygulama indikatörü ise AKY ürününün kullanımı ile elde edilen ölçüm sonuçlarının göstergesidir. OSMM, ürün indikatöründe 12 ve uygulama indikatöründe 15 olmak üzere toplamda 27 tane alt indikatörden oluşur. Kullanıcı bu alt indikatörlere 1, 3 ve 5 olmak üzere skorlar vererek her iki açıdan (ürün indikatörü ve uygulama indikatörü için) ayrı ayrı en iyi ürünü belirler. OSMM kendi değerlendirme modelinde bu indikatörlerin hiyerarşik listesini ve skorlar verilirken uyulması gereken kriterleri vermiştir. Ancak bu indikatörler ürünün kaynak kodunu değerlendirme imkânı sunmadığı için model bu yönüyle eleştiri almış ve yetersiz kalmıştır [9].

OpenBRR [10], AKY'ların değerlendirilmesi için önerilmiş diğer bir modeldir. Değerlendirme için genellikle yüksek-düzye (İng. high-level) metrikler temel alınmıştır. Bu metrikler arasında işlevsellik (İng. functionality), operasyonel yazılım

özellikleri (İng. operational software characteristics), destek (İng. support), servis (İng. service), benimseme (İng. adoption), geliştirme süreci (İng. development process) sayılabilir. Her bir metrik direkt olarak ölçülemeyeceği için OpenBRR modeli bu metrikleri hiyerarşik olarak alt metriklere ayırmıştır. Bu alt metriklere ağırlıklar verilerek değerlendirme yapılır. Fakat bu metrikler (örn: işlevsellik) öznel değerlendirmeler yapmaya yatkındır. Bu yüzden de bu model değerlendirmelerin özneliği yönünün (kullanıcı-tabanlı) fazla olmasından dolayı eleştirilmiştir [9].

Bu alanda önerilen diğer bir model QSOS [11] modelidir. Model iterasyonlarla ilerleyen 4 aşamadan oluşur. Birinci aşamada değerlendirme yapılacak kriterler tanımlanır. İkinci aşamada değerlendirilecek AKY için belirlenen kriterler dâhilinde bilgiler toplanır ve kriterlere 0 ile 2 arasında (hangi değer aralığı için hangi skor atanacağı metotta belirtilmiştir) skorlar atanır. Üçüncü aşamada, belirlenen kriterlere göre filtreler tanımlanır ve bu filtreler sayesinde ilişkisiz ürünler elenir. Son aşamada ise kullanıcı ihtiyaçlarını karşılayan uygun AKY'ler belirlenir. QSOS hiyerarşisindeki kriterler objektif değerlendirmeye uygun olduğu ve kullanıcıya bağımlı olmadığı halde, bu metodun yeterince esnek olmaması ve uygulanmasının zor olması bir dezavantajdır [9]. Ayrıca bu metodun skor değerlerinin 0 ile 2 arasında olması eleştirilmiş ve ürün seçimlerinde aralığın küçük olmasının kullanıcının işini zorlaştırdığı düşünülmüştür [9].

Diğer güncel modellerden bir tanesi ise QOS-OSS modelidir [9]. Bu model bizim önerimizde olduğu gibi değerlendirmesini yaparken hem kaynak koddan hem de ürünün bize sağladığı tarihi verilerden faydalanmıştır. Ölçüm için hiyerarşik yapıdan oluşan kalite modeli tanımlamıştır. Model değerlendirme yaparken otomatik değerlendirme sürecini ön plana çıkarmıştır ve kullanıcının görüşleri ikinci planda kalmıştır. Ürün seçimi için kullanıcı ihtiyaçlarının önemli olduğu düşünüldüğünde metodun kullanıcı ihtiyacını tam karşılamaması olası bir durumdur. Ayrıca toplum-tabanlı metriklerin otomatik değerlendirmeye yatkın metrikler olarak seçilmeye çalışılması, değerlendirmede kullanılacak metriklere bir sınırlama getirmektedir.

Yukarıda anlattığımız, bizim çalışmamıza yön vermeyi sağlayan metodların yanında, literatürde 20'den fazla model mevcuttur. K.J. Stoll ve M.A. Babar [46] literatürde bulunan 20 tane değerlendirme metodunu ele almış ve çalışmalarında, AKY seçiminde hangi modelin kullanıcı için uygun olduğunu anlamak amacıyla FOCOSEM (Framework for Comparing Open Source Software Evaluation Methods)

[46] isminde bir çerçeve (İng. framework) geliştirmişlerdir. Bu çerçeve değerlendirme modeli seçimine yönelik birçok sorudan oluşmaktadır. Kullanıcı bu soruların cevaplarına göre kendine en uygun değerlendirme modelini seçmektedir.

Bu tez kapsamında yukarıda anlatılan AKY'ların değerlendirilmesine yönelik metotların iyi yönleri alınarak kapsamı kullanıcı ihtiyacına göre şekillenebilen bir metot geliştirilmek istenmiştir. Bizim çalışmamızın önceki çalışmalara benzerlikleri Çizelge 3.1'de özetlenmiştir. Çizelge 3.1'de gösterildiği gibi, bazı metotlarda kullanıcı değerlendirmesi (kullanıcı-tabanlı) ön plana çıkarılmış bazıları ise değerlendirme tamamen otomatik (kullanıcı görüşleri önemsenmeden) yapılmıştır. Bazı çalışmalarda sadece toplum-tabanlı değerlendirmeye ağırlık verilmiş (örn. OSMM) bazıları ise kod-tabanlı değerlendirme önemsenmiştir (örn. QOS-OSS).

Çizelge 3.1 Literatürde AKY seçimi için önerilen metotlar ve deneysel çalışmaların karşılaştırılması

	Çalışmalar	Değerlendirme boyutu		Değerlendirme şekli		Değerlendirme modeli	
		Kod-tabanlı	Toplum-tabanlı	Kullanıcı-odaklı (sübjektif)	Kural-odaklı (objektif)	Yeni değerlendirme modeli	Mevcut modeli uyarlama
Metot öneren çalışmalar	OSMM [8]		X	X		X	
	OpenBRR [10]		X	X		X	
	QSOS [11]		X		X	X	
	QOS-OSS [9]	X	X		X	X	
	Önerilen metot	X	X	X	X		X
Deneysel çalışmalar	I. Stamelos ve arkadaşları [47]	X			X		
	D. Wheeler [48]		X				
	J. W. Sung ve arkadaşları [49]					X	
	C. M. Hanefi ve arkadaşları [41]	X			X		
	V. Garousi [50]		X				
	A. Mackus ve arkadaşları [51]		X				
	J. Antony ve arkadaşları [38]	X			X		
	J. Antony [37]	X			X		

Bu çalışmada bilgi ihtiyacının belirlenmesinde değerlendirici ihtiyaçlarının ön plana çıkarıldığı (kullanıcı-odaklı) ama ölçüm sonuçlarına skor verirken değerlendirciden bağımsız (kural-odaklı) uygulanan bir metot sunulmuştur. Bu metot sübjektif olarak belirlenen bilgi ihtiyaçlarının objektif olarak ölçülmesine olanak sağlar. Ayrıca diğer metotlar AKY'ların karakteristiğini yansıttığını düşündükleri ve önerdikleri yeni kalite modelleri çerçevesinde ölçümlerini yapmışlardır. Bizim önerdiğimiz metotta uluslararası standart olan ISO/IEC 25010 kalite modelinin öznitelikleri ve alt özniteliklerine bağlı kalınarak ölçümler yapılmıştır.

Diğer bir farklı yön ise bazı çalışmaların [9] bizim çalışmamız gibi, iki boyutta değerlendirme imkânı sunması fakat bunu yaparken iki boyutu tek boyut gibi ele alıp sonuçları elde etmeleridir. Örneğin [9] çalışması, güvenilirliği ölçerken ölçümü iki ayrı boyutta ele almamış ve güvenilirliği ölçmek için elde edilen kod-tabanlı ve toplum-tabanlı metrikleri, bu özneteliği ölçmek için kullanılan metrikler olarak, tek çatı altında toplamıştır. Bizim çalışmamızda bir özneteliğin ölçümünde, her bir boyut için değerlendirmeler ayrı ayrı yapılmış ve bu ayrı değerlendirmeler sonunda, tek bir sonuç (indeks) elde edilmiştir.

Çizelge 3.1'de herhangi bir modele bağlı kalmadan yapılan deneysel (İng. empirical) çalışmalar da verilmiştir. I. Stamelos ve arkadaşları [47] çalışmalarında AKY'ların kod kalitesini, test edilebilirlik, okunabilirlik (İng. readability), basitlik (İng. simplicity) ve kendini tanımlama kriterleri bakımından değerlendirmiştir. Bu değerlendirmeyi yaparken Legiscope kod analiz aracının sağladığı metrikleri kullanmışlar ve öznetelikleri bu metriklerle ilişkilendirmişlerdir. Her metriğin bir özneteliği ölçmedeki ağırlığının farklı olduğunu öngörmüşler ve her özneteliği ölçmek için ilişkili metrikleri kullanarak formüller üretmişlerdir. Bu çalışma yukarıda verilen dört öznetelik için sınırlı kalmış ve sadece kaynak kodu ölçerek yapılmıştır.

Deneysel alanda en önemli çalışmalardan bir tanesi D. Wheeler tarafından yapılmıştır [48]. Yazar çalışmasında AKY değerlendirme sürecini ticari yazılımları değerlendirme süreci ile kıyaslamış ve AKY'ları toplum-tabanlı olarak dört aşamada değerlendiren bir süreç önermiştir. Bu süreç herhangi bir vaka çalışmasında uygulanmamış ve çalışmada bazı öznetelikleri ölçmek için ne gibi veriler kullanılabileceği konusunda ipucu verilmiştir. Mesela bakım yapılabilirliğin ölçülebilmesi için ürünlerin elektronik posta listelerinin incelenmesi tavsiye edilmiştir. Buradan geliştiriciler arasındaki konuşmalardan ürünün geliştirilmesi için harcanan

çaba hakkında (mesaj içeriğini kullanarak) bilgi edinilebileceği savunulmuştur. Diğer yandan güvenilirlik için ise ürünlerin olgunluk derecesine ve ürünün son kullanıcılar için ne derece hazır olup olmadığına bakılması önerilmiştir. Bunun için ürünün web sitesinden ürünün olgunluğu ile ilgili bilgilerin edinilebileceğine değinilmiştir.

Literatürde göze çarpan diğer bir çalışmada ise J. W. Sung ve arkadaşları [49] AKY seçimi için yeni bir kalite modeli sunmuşlardır. Literatürde var olan kalite modellerinin AKY karakteristik özelliklerini tam olarak yansıtmadığını düşünmüşler ve ISO 9126 kalite modelini yeniden düzenlemişlerdir. 23 tane büyük çaplı şirket belirlenmiş ve kalite öznitelikleri ve alt öznitelikleri için AKY'lara uygunluk bakımından anket yapılmıştır. Bu anket sonucunda hangi özniteliklerin ve alt özniteliklerin AKY'lar için öncelikli olduğu ve hangi AKY karakteristik özelliklerini yansıttığı bulunmaya çalışılmıştır. Çalışma sonucunda 4 öznitelikten ve 10 alt öznitelikten oluşan yeni bir model önerilmiştir. Daha sonra bu alt özniteliklerin hepsini ölçebilmek için formüller türetilmiştir.

Bu alanda C. M. Hanefi ve arkadaşları [41] tarafından yapılan çalışma, AKY'ların bazı özniteliklerini değerlendirirken metriklerin elde edilmesi konusunda yol göstericidir. Çalışmada C&K metrik serisi kullanılmış ve bu metrikler verimlilik, karmaşıklık, anlaşılabilirlik, yeniden kullanılabilirlik, test edilebilirlik ve dayanıklılık ile ilişkilendirilerek bu özniteliklerin ölçümünde kullanılmıştır.

Toplum-tabanlı değerlendirme yaparken kullanılacak metrikler sunan diğer bir çalışma V. Garousi [50] tarafından yapılmıştır. Bu çalışmada AKY'ların başarısını ölçmek için dört tane AKY ürünü 3 tane toplum-tabanlı metrik (developer sayısı, indirilme sayısı, hata çözme performansı) kullanılarak karşılaştırılmıştır. Çalışma bu metriklerin kullanımı açısından bizlere bilgi sağlasa da hem ürünlerin kaynak kodlarını değerlendirmedeğinden hem de kullanılan toplum-tabanlı metriklerin kapsamının çok geniş olmayışından dolayı geniş çaplı ürün seçimi için pek önerilmemektedir.

Toplum-tabanlı değerlendirme bakımından rehber olan diğer bir çalışma A. Mackus ve arkadaşları [51] tarafından yapılan çalışmadır. Bu çalışmada Apache ve Mozilla'da bulunan tarihi verilerin neler olduğu, neler içerdiği ve nasıl kullanılması gerektiği hakkında kapsamlı bir inceleme yapılmıştır.

A. S. Jadhav ve arkadaşları [52] literatürde AKY'ların değerlendirilmesi ve seçilmesi ile ilgili yapılan çalışmaları inceleyerek belirledikleri 5 araştırma sorusu (İng. research question) çerçevesinde bir sistematik haritalama (İng. systematic mapping) çalışması yapmışlardır. Bu çalışma, literatürde yapılan çalışmaları listeleme ve anlatma adına çok yararlı bir çalışmadır.

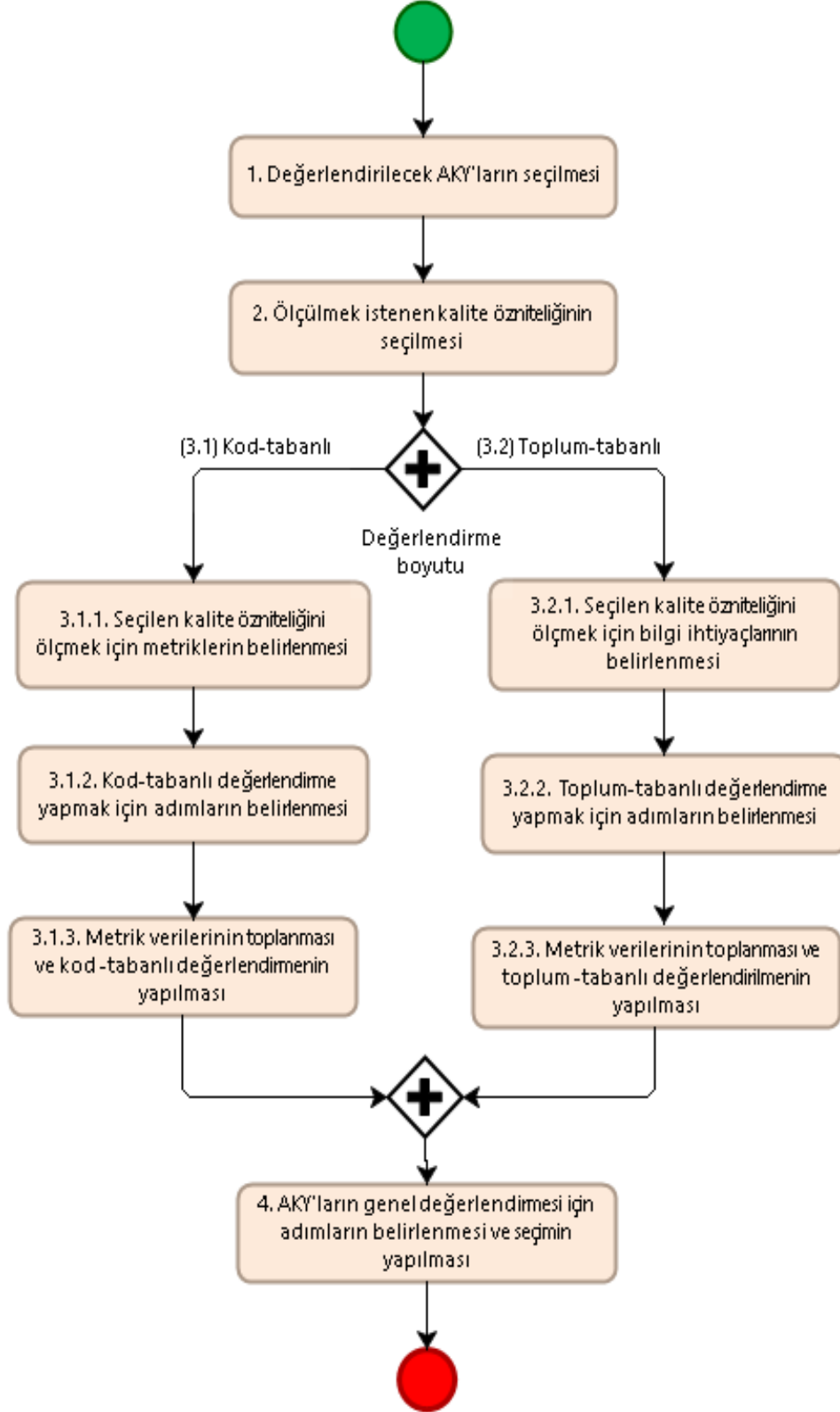
Bu alanda yapılan çalışmalardan bazıları AKY'ları, önemli bir öznelik olan ve hemen hemen bütün kalite modelleri tarafından önerilen güvenilirlik açısından incelemişlerdir. Bazı çalışmalar [37][38][39][40] sadece kaynak kodunu kullanarak güvenilirliği ölçmüşlerdir. Bu ölçümü yaparken güvenilirlikle C&K metrik serisini ilişkilendirmişler ve geliştirilen formülle ürünün hataya yatkınlığı ve gelecekteki davranışı hakkında bilgi edinmişlerdir. Bazı çalışmalar [9] ise güvenilirliği ölçmek için ürünlerin veri tabanlarında bulunan tarihi verileri kullanarak metrikler üretmişlerdir.

Özetle bu tez çalışmasında, sadece kod-tabanlı değerlendirme yapan çalışmalardan [9][37][38][41][47] ve sadece toplum-tabanlı değerlendirme [8][10][11][48][50][51] yapan çalışmalardan farklı olarak, [9] çalışmasında yapıldığı gibi iki boyutta değerlendirme yapılmıştır. Bu değerlendirme yapılırken bazı aşamalarda (ihtiyaçların belirlenmesi), [8][10] çalışmalarında yapıldığı gibi kullanıcı görüşleri önemsenerek (kullanıcı-tabanlı boyut), bazı aşamalarında (ağırlıklandırma) ise, [9][11][38][37][41][47] çalışmalarında yapıldığı gibi kullanıcı görüşü önemsenmeden (kural-odaklı) değerlendirme yapılmıştır. Değerlendirme aşamasında diğer modellerdeki [8][9][10][11][49] gibi yeni bir değerlendirme modeli önerilmemiş ve ISO/IEC 25010 kalite modeline bağlı kalınmıştır.

4. GENEL DEĞERLENDİRME SÜRECİ

Bu tez çalışmasında bakım yapılabilirlik ve güvenilirlik için geliştirilen değerlendirme yöntemi için, bu bölümde anlatılacak olan genel değerlendirme süreci takip edilmiştir. Dolayısıyla genel değerlendirme süreci takip edilerek bu iki öznelik dışında değerlendirilmek istenen başka öznelikler için de değerlendirme yöntemi oluşturulabilir.

AKY'ların sürekli gelişerek popülerliğini korumasındaki en önemli sebep, bilgilerinin kullanıcılara açık olmasıdır. Kullanıcılar hem ürünlerin kaynak kodlarına hem de ürünlerin geliştirme aşamasında veri tabanlarında kaydedilen bilgilere erişebilmektedirler. Bu sebeple bu tez çalışmasında, AKY'ları hem kod-tabanlı hem de toplum-tabanlı değerlendirme imkânı sunan iki boyutlu bir metot geliştirilmiştir. AKY'ların değerlendirilmesi ve kullanıcı ihtiyacını karşılayan en uygun ürünün seçilmesi için geliştirilen metodun aşamaları Şekil 4.1'de verilmiştir.



Şekil 4.1 AKY'ların değerlendirmesi için geliştirilen genel metodun aşamaları

Şekildeki metot incelendiğinde sürecin, değerlendirmesi yapılacak aday yazılım ürünlerinin belirlenmesinden başladığı ve bu aday yazılım ürünlerinden kullanıcı ihtiyacını karşılayan en uygun ürünün seçimiyle sonlandığı görülmektedir. Geliştirilen genel sürecin aşamaları aşağıdaki alt bölümlerde açıklanmıştır.

4.1. Değerlendirmesi Yapılacak AKY'ların Seçilmesi (1. Aşama)

Bu ilk aşamada ihtiyaç sahipleri öncelikle hangi çeşit yazılım ürünü için değerlendirme yapacaklarını belirlemelidirler. Bu doğrultuda piyasada var olan, gereksinimlerini karşılayacak aday yazılım ürünlerini araştırmalıdır.

İhtiyaç duyulan aday AKY ürünlerini ararken akla ilk gelen yöntem, bir İnternet arama motoru (Google, Yandex, Yahoo, vb.) kullanarak arama yapıp değerlendirmeye dâhil edeceğimiz aday ürünleri belirlemektir. Eğer arama yapacağımız ürünün ismini biliyorsak ona rakip olabilecek isimleri bulmaya yönelik araştırma yapmak da uygun olur. Arama yapılacak ürünün ismi tam olarak bilinmiyorsa, aday olacak ürünlerden herhangi birini gözden kaçırmamak için ürün ismi ile ilgili bütün kombinasyonlar denenmelidir. Örneğin, X formatını Y formatına dönüştüren bir yazılım ürünü için arama yaptığımızda, X2Y, XtoY, vb. kombinasyonların denenmesinde fayda vardır.

Arama yaparken kullanılacak ikinci yöntem ise ürünlerin kodlarını ve dokümanlarını sağlayan sitelerden (Apache, SourceForge, Github, Debian ve Savannah, vb.) aday AKY ürünlerini belirlemektir.

Bütün bu araştırmaları yaparken ürünlerin Bölüm 2.1.4'de bahsedilen açık kaynak yazılım lisanslarına (General Public Licence (GPL), Library or Lessor General Public Licence (LGPL), BDS-style vb.) sahip olduğuna dikkat etmek gereklidir [7]. Çünkü lisansların bizlere sağladığı imkânlar dışında ürün üzerinde işlem yapıldığında büyük yaptırımlarla karşılaşılabilir.

4.2. Ölçülmek İstenen Kalite Özniteliğinin Seçilmesi (2. Aşama)

Bu aşamada, belirlenen AKY ürün alternatiflerinin hangi kalite özniteliklerine göre değerlendirileceği ve karşılaştırılacağı belirlenir. Bölüm 2.2'de bahsedildiği gibi, geçmişten günümüze önerilen kalite modellerinden en bilinen olanları McCall (1977), Boehm (1978), FURPS (1992), Dromey (1995) ve ISO 9126 (2001) modelleridir. Farklı kalite modellerinin içerdiği öznitelikler (İng. attributes) farklılık göstermektedir. Bazıları önceki modellere yeni öznitelikler eklemiş, bazıları önceki modellerin

özniteliklerini modellerinden çıkarmış ve bazıları da başka model için önerilen bir özniteliği kendi modelinde bir özniteliğin alt özniteliği olarak kullanmıştır. Çizelge 4.1’de en bilinir kalite modellerinin önerdiği kalite öznitelikleri listelenmiştir.

Çizelge 4.1 En bilinir kalite modelleri ve öznitelikleri

Kalite öznitelikleri	Boehm	McCall	FURPS	ISO 9126	Dromey
Verimlilik	X	X	X	X	X
Güvenilirlik	X	X	X	X	X
Fonksiyonellik			X	X	X
Bakım yapılabilirlik	X	X	X	X	X
Taşınabilirlik	X	X		X	X
Kullanılabilirlik		X	X	X	X
Test edilebilirlik	X	X		X	
Doğruluk		X			
Anlaşılabilirlik	X			X	
Esneklik		X	X		
İnsan yönetimi	X				
Bütünlük		X		X	
Birlikte işlerlik		X		X	
Olgunluk					X
Değişebilirlik	X				
Tekrar kullanılabilirlik		X			X

Çizelgede geçmişten günümüze en bilinen modellerin önerdiği ortak özniteliklerin; verimlilik, güvenilirlik, taşınabilirlik, kullanılabilirlik ve bakım yapılabilirlik olduğu görülmektedir [12].

4.3. Değerlendirme Yapılacak Boyutun Seçilmesi (3. Aşama)

Bu aşamada her iki değerlendirme boyutu (kod-tabanlı veya toplum-tabanlı) için ayrı süreç akışları izlenecektir.

4.3.1. Kod-tabanlı Değerlendirme

Bu aşamada AKY’lar sağlayan sitelerden ürünlerin kaynak kodlarına erişilir. Bir önceki aşamada (bkz. Şekil 4.1, **2.aşama**) belirlenen öznitelikleri ölçmek için hangi metrikleri kullanabileceğimizin araştırılması yapılmalıdır (bkz. Şekil 4.1, **3.1.1. aşama**). Bu araştırmayı yaparken belirlenen öznitelikleri ölçmeye yönelik araştırma soruları (İng. research question) oluşturulabilir. Bu araştırma soruları kullanılarak belirlenen öznitelikleri ölçmeye yönelik metrikler literatürden çıkarılabilir.

Araştırma soruları kullanılarak uygun metrikler bulunduktan sonra, ürünlerin kaynak kodlarından ilişkili metrik değerlerini ölçmek için, bir kod analiz aracı kullanılır. Her bir AKY ürünü için belirlenen metriklerin değerleri ölçülür.

Daha sonra Şekil 4.1'de **3.1.2. aşamada** gösterildiği gibi, ölçüm sonucunda elde edilen sayısal metrik değerlerinin değerlendirmede nasıl kullanılacağına ilişkin adımlar belirlenir.

Ardından şekilde **3.1.3. aşamada** gösterildiği gibi, elde edilen metrik değerleri ve belirlenen adımlar uygulanarak bu boyut (kod-tabanlı) açısından en tercih edilebilir AKY ürünü seçilir.

4.3.2. Toplum-tabanlı Değerlendirme

AKY'lar kaynak kodlarının yanı sıra piyasaya sürüldüğünden itibaren düzenli olarak kaydedilen tarihi verilere (İng. historical data) ulaşma imkanı da sunar. Bu veriler Concurrent Version Control (CVS), Problem Reporting Database (BugDB) ya da Bugzilla gibi veri tabanlarında kaydedilmektedir. Bu veri tabanlarında; karşılaşılan hata (İng. bug) sayıları, eklenen ve silinen satır sayıları, aktif geliştirici (İng. developer) sayıları, hata önem dereceleri (İng. bug severity), geliştirici-geliştirici ve kullanıcı-geliştirici arasındaki e-posta sayıları, hatanın durumu (açık, kapalı, çözüldü) (İng. bug status-opened, fixed, closed), sıkça sorulan sorular ve benzeri veriler düzenli olarak kaydedilir.

Toplum-tabanlı değerlendirme yapılırken ihtiyaçların nasıl belirleneceği ve ihtiyaçlar doğrultusunda bu verilerin nasıl kullanılacağı gerçekten karmaşık bir durumdur. Bu karmaşık durumdan kurtulmak için bilgi ihtiyaçlarının belirlenmesinde, ilgili verilerin toplanmasında ve bu verilerin değerlendirilmesinde yol gösterici olarak ISO/IEC 15939 ölçme süreci takip edilebilir. Bu sayede ölçümler sağlam bir temele dayandırılmış olur.

İlk olarak, daha önce (Şekil 4.1, **2. aşama**) seçilen kalite özneteliğini ölçmeye yönelik bilgi ihtiyaçları (İng. information needs) belirlenir (Şekil 4.1, **3.2.1. aşama**). Bu bilgi ihtiyaçlarını karşılamak için, AKY ürünlerinin veri tabanlarındaki bilgiler derinlemesine analiz edilir. Bilgi ihtiyaçlarını karşılamak amacıyla, nicel değerler elde edebilmek için metrikler belirlenir.

Daha sonra Şekil 4.1'in **3.2.2. aşamasında** gösterildiği gibi ölçüm sonucunda elde edilen sayısal metrik değerlerinin değerlendirmede nasıl kullanılacağına ilişkin adımlar belirlenir.

Ardından şekilde **3.2.3. aşamada** gösterildiği gibi, elde edilen metrik değerleri ve belirlenen adımlar uygulanarak bu boyut (toplum-tabanlı) açısından en tercih edilebilir AKY ürünü seçilir.

4.4. AKY'ların Genel Değerlendirmesi için Adımların Belirlenmesi ve Seçimin Yapılması

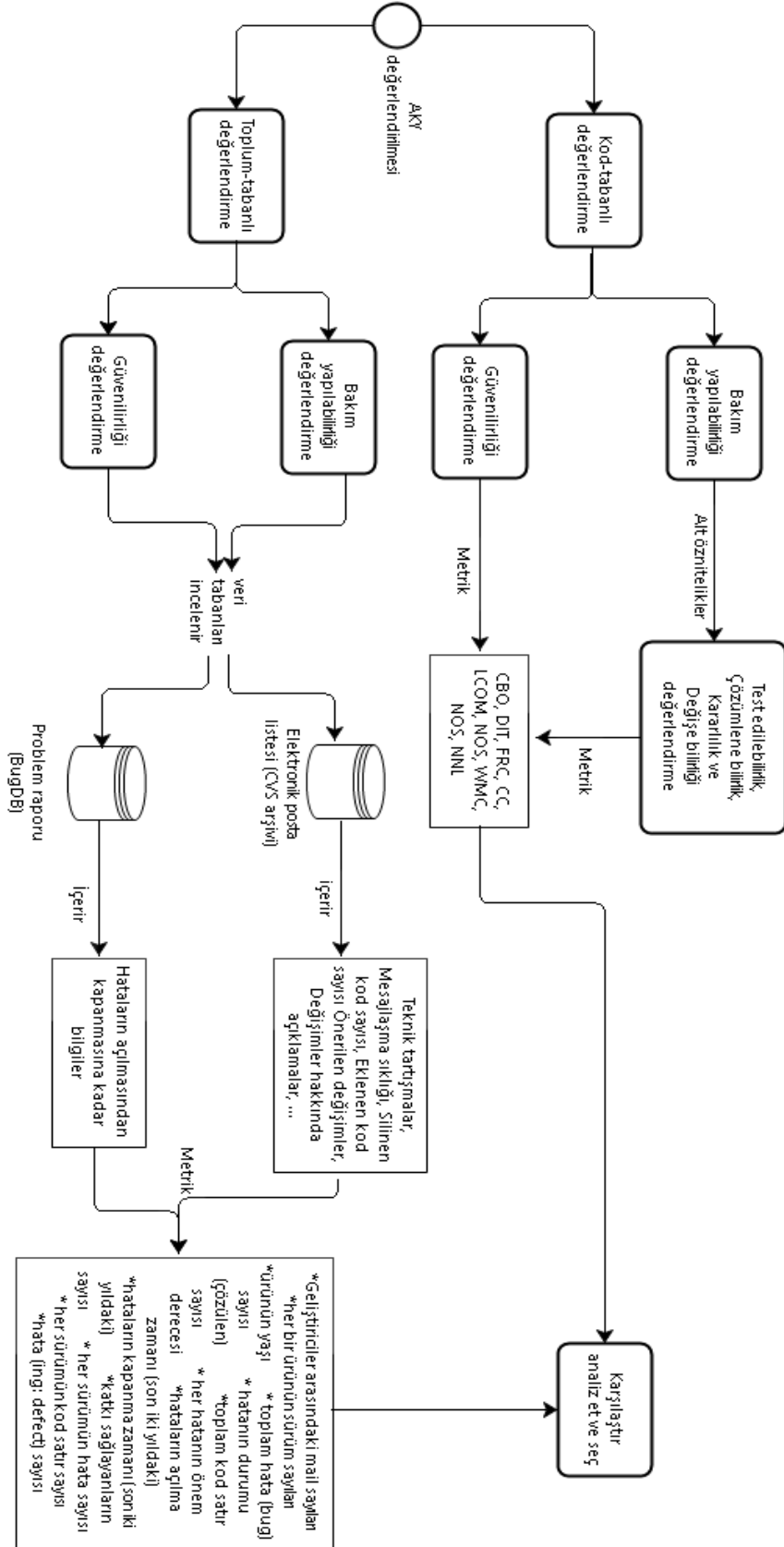
İki boyut için ayrı ayrı değerlendirmeler yapıp kullanıcı ihtiyacını karşılayan en iyi ürünler seçildikten sonra, bu aşamada (Şekil 4.1, **4. aşama**) iki boyutlu değerlendirme sonuçları kullanılarak en iyi ürün seçilir. Bunu yaparken genel değerlendirme için adımlar belirlenmeli ve en iyi ürünün seçimi için uygulanmalıdır. Belirlenen adımlar kullanıcı ihtiyaçlarına veya ölçülmek istenen özneliğe göre değişebilir. Mesela bir özneliği ölçerken toplum-tabanlı değerlendirmede kullanılan metrikler, daha belirleyici olabilir. Bu durumda toplum-tabanlı metriklerden elde edilen sonuçlar, kod-tabanlı metriklerden elde edilen sonuçlara göre seçimde daha fazla önem arz edecektir.

5. BAKIM YAPILABİLİRLİK VE GÜVENİLİRLİK İÇİN DEĞERLENDİRME YÖNTEMİ

Bu tez çalışmasında Bölüm 4'te önerilen süreç, bakım yapılabilirlik ve güvenilirlik için uygulanacaktır. Seçilen AKY ürünleri bu iki özneliğe göre hem kod-tabanlı hem de toplum-tabanlı olarak iki boyutlu değerlendirilecektir. Şekil 5.1'de bu iki özneliğin değerlendirilme aşamaları özetlenmiştir. Şekilde görüldüğü gibi iki ayrı boyutta farklı yollar izlenmiştir. Kısaca özetleyecek olursak:

Kod-tabanlı değerlendirmede bakım yapılabilirliğin ölçülmesi için alt öznelikler (test edilebilirlik, kararlılık, çözümlenebilirlik ve değişebilirlik) kullanılmıştır. Bu öznelikler literatürde araştırılıp kullanılan metrikler üzerinden ölçülmüştür. Güvenilirlik ise direkt olarak literatürde kullanılan metriklerle ilişkilendirilerek ölçülmüştür.

Toplum-tabanlı değerlendirme yapılırken öncelikle bu iki öznelikte ölçülmek istenen bilgi ihtiyaçları belirlenmiş ve bu bağlamda şekilde gösterilen veri tabanları (Problem raporu veri tabanı, BugDB ve e-posta listesi, CVS arşivi) kullanılarak derinlemesine analiz edilmiştir. Daha sonra şekilde bu veri tabanlarının içerdiği veriler belirtilmiştir. Veri tabanlarındaki veriler kullanılarak metrikler belirlenmiş ve bu metrik değerleri üzerinden ölçümler yapılmıştır.



Şekil 5.1 AKY'ların bakım yapılabirliğini ve güvenilirliğini değerlendirmek için geliştirilen metodun aşamaları

5.1. Değerlendirme Yapılacak AKY'ların Seçilmesi

Öncelikle ihtiyaç dâhilinde birbirine alternatif olabilecek AKY ürünleri araştırılmıştır. Değerlendirme için bu metodun Java tabanlı kod inşa araçlarına (İng: Java built tool) uygulanması kararlaştırılmıştır. Bu bağlamda ilk önce İnternet arama motorları (Google, Yandex, Yahoo, vb.) kullanılarak piyasada var olan Java dilinde yazılmış kod inşa araçları araştırılmıştır. Araştırma yapılırken ürünlerin işlevsellik yönünden birbirine alternatif olabilecek ürünler olmasına dikkat edilmiş ve üç tane AKY ürünü seçilmiştir: Apache Ant, Apache Maven ve Apache Archiva. Daha sonra kullanıcılara AKY ürünleri sağlayan İnternet sitelerine de (Apache, SourceForge, Debian, Savannah, vb.) bakılarak seçilen ürünlerin popülerliği ve işlevsel birbirine alternatif olabilme dereceleri doğrulanmıştır. Çizelge 5.1'de seçilen AKY ürünlerinin bilgileri verilmiştir.

Çizelge 5.1 Belirlenen aday AKY ürünlerinin özellikleri

Ürün özellikleri	Apache Archiva	Apache Maven	Apache Ant
İnternet sitesi	http://archiva.apache.org/index.cgi	http://maven.apache.org/	http://ant.apache.org
Ürün türü	Kod inşa aracı	Kod inşa aracı	Kod inşa aracı
Programlama dili	JAVA	JAVA	JAVA
Üretim yılı	2006	2002	2000
Mail arşivi	http://archiva.apache.org/mail-lists.html	http://maven.apache.org/mail-lists.html	http://ant.apache.org/mail.html
Problem veri tabanı	http://issues.apache.org/jira/browse/MRM	https://issues.apache.org/jira/browse/MNG	https://bz.apache.org/bugzilla/buglist.cgi?product=Ant

Seçilen araçların piyasaya sürülmesinde temel amaçlar farklılık gösterebilir. Maven ve Ant, Java projelerinin iskeletinin oluşturulmasını ve böylece projelerin yönetilmesinin ve geliştirilmesinin kolaylaşmasını sağlayan proje yönetim araçları olarak üretilmiştir. Ayrıca, projelere belirli standartlar getirerek JAR dosyalarını projeler arasında paylaşmak, projelerin içeriklerini anlaşılır bir yapıda tutmak ve projeleri kolay bir şekilde yayınlamak için kullanılır. Archiva, temel amaç olarak projelerde sıklıkla kullanılan kütüphaneler için bir depo oluşturmak ve bu kütüphaneye projede çalışan herkesin kolay erişebilmesini sağlamak amacıyla üretilmiştir. Ayrıca yazılım inşasının her aşamasındaki süreçleri otomatikleştirmesi temel üretim amaçları arasındadır. Bu inşa araçları genel üretim amacı dışında

birçok ortak özelliklere sahiptirler ve her birinin yaptığı çoğu işi bir diğer aracın yapması mümkündür. Örneğin, seçilen bu üç ürün işlevsel olarak; proje geliştirilirken proje içinde standart oluşturulmasını, geliştirme sürecinin basitleştirilmesini, dokümantasyonların etkili bir şekilde oluşturulmasını, projelerde ihtiyaç duyulan kütüphanelere otomatik olarak erişilebilmesini ve tümleşik geliştirme ortamı (İng. Integrated Development Environment - IDE) bağımlılığını ortadan kaldırılmasını sağlamak gibi birbirine alternatif olabilecek özellikler içerirler.

5.2. Değerlendirme Yapılacak Kalite Özniteliğinin Belirlenmesi

Bir önceki aşamada birbirine alternatif olarak seçilen AKY ürünlerinin hangi kalite özniteliklerine göre değerlendirileceği kararlaştırılmıştır. Bu bağlamda geçmişten günümüze önerilen en bilinen kalite modelleri incelenmiş ve hepsinin ortak olarak önerdiği kalite öznitelikleri içerisinde, ölçülmeye ihtiyaç duyulanlar belirlenmiştir. Çizelge 4.1’de gösterildiği gibi, bütün modeller için önerilen öznitelikler içinden ihtiyacımıza göre güvenilirlik ve bakım yapılabilirlik seçilmiştir. Değerlendirilecek ürünlerin kod inşa aracı olması da bu iki özneneliğın seçilmesinde etkili olmuştur. Çünkü kod inşa aracının güvenilir olması ve aynı zamanda uzun ömürlü kullanım açısından bakım yapılabilirliğinin kolay olması beklenir. Bu sebeple seçilen üç Java tabanlı kod inşa aracı, bu iki öznenelik bakımından değerlendirilmiştir.

5.3. Değerlendirme Yapılacak Boyutun Belirlenmesi

Değerlendirme yapılacak aday ürünler ve bu ürünlerin hangi kalite özniteliklerine göre değerlendirileceği belirlendikten sonra, değerlendirme yapılacak boyut (kod-tabanlı veya toplum-tabanlı) seçilmiştir. Örneklenen seçimde her iki boyut ayrı ayrı değerlendirilmiştir.

5.3.1. Kod-tabanlı Değerlendirme

Aday AKY ürünleri ve değerlendirme yapılacak kalite öznitelikleri belirlendikten sonra kod-tabanlı değerlendirme yapmak için, metriklerin belirlenmesi gerekir. Bölüm 4.3.1 de bahsedildiği gibi bakım yapılabilirliği ve güvenilirliği ölçmek için metriklerin belirlenmesi aşamasında her bir öznenelik için araştırma soruları oluşturulmuştur:

Araştırma Sorusu 1: Hangi metrikler bakım yapılabilirliği veya bakım yapılabilirliğın alt özneneliklerini ölçmek için kullanılabilir?

Araştırma Sorusu 2: Hangi metrikler güvenilirliği veya güvenilirliğin alt özniteliklerini ölçmek için kullanılabilir?

Bu iki araştırma sorusu çerçevesinde iki özniteliği ölçmeye yönelik metrikler araştırılmıştır. Bakım yapılabilirliği ve güvenilirliği değerlendirmede kullanılacak metrikleri belirlemek için ilk olarak, ISO/IEC 25010 standardı içindeki metrikler gözden geçirilmiştir. Ne var ki bu metrikler, nesneye yönelik özellikleri olan Java inşa araçlarını değerlendirmek için uygun görülmemiştir. Standartta tanımlanan metrikler içinden seçim yapılamamış olmasının bir başka sebebi de tanımlanan metriklerin çoğunun geliştirme bağlamına ilişkin bilgi toplanmasını gerektiriyor olmasıdır. Bu AKY'lar dünya üzerinde farklı konumlarda ve ortamlardaki farklı onlarca geliştirici tarafından üretildiklerinden, bağlamla ilgili bilgilere ulaşmak maalesef mümkün değildir.

Yukarıda belirtilen sebeple literatürdeki çalışmalar araştırılmış ve bu iki özniteliği ölçmek içinde C&K metrik serisi, iç içe döngü sayısı, çevrimsel karmaşıklık ve açıklamaların sıklığı metriklerinin kullanılmasına karar verilmiştir. Bu metrikler Çizelge 5.2'de gösterildiği gibi iki öznitelik ile ayrı ayrı ilişkilendirilmiştir.

Metrikler belirlendikten sonra üç aday AKY ürününün kaynak kodlarının metrik değerlerini hesaplamak için bir kod analiz aracı olan Understand Scitool [53] kullanılmıştır.

Çizelge 5.2 Bakım yapılabilirlik ve güvenilirliği ölçmek için metrikler

Araştırma Sorusu	Ar. Sor 1: Hangi metrikler bakım yapılabilirliği (veya bakım yapılabilirliğin alt özniteliklerini) ölçmek için kullanılabilir?				Ar. Sor 2: Hangi metrikler güvenilirliği (veya güvenilirliğin alt özniteliklerini) ölçmek için kullanılabilir?
Metrik	Test edilebilirlik	Kararlılık	Değişebilirlik	Çözümlenebilirlik	CBO DIT LCOM
	NNL NOC RFC	CBO DIT NNL	CBO DIT LCOM	CC NOS WMC	NOC RFC WMC

Bakım yapılabilirliğin ölçülmesi için literatürdeki ilişkili çalışmalar [9][34][41][54][55][56][57][58] analiz edilmiştir. Bu çalışmalardan yola çıkarak bakım yapılabilirliği ölçmek için bakım yapılabilirliğin alt öznitelikleri kullanılmıştır. Bu alt

öznitelikler test edilebilirlik, çözümlenebilirlik, kararlılık ve değişebilirlik. Dolayısıyla bakım yapılabilirliği direkt ölçmek yerine belirlenen metrikler Çizelge 5.2’de gösterildiği gibi bu özneliğin alt öznitelikleriyle ilişkilendirilmiştir. Kod analiz aracı (Understand Scitool) kullanılarak metrik değerleri elde edildikten sonra bu metrik değerleri kullanılarak bakım yapılabilirliğin ölçülebilmesi için adımlar belirlenmiştir. Metriklerin sayısal değerleri ve Bölüm 2.4’deki tanımları kullanılarak, bakım yapılabilirliğin her bir alt özneliği için, seçilen ürünlere ağırlıklar verilmiştir. Ayrıca kullanıcı da ihtiyacı doğrultusunda önem derecesine göre, bu alt özniteliklere ağırlıklar vermiştir. Bütün bu ağırlık değerleri değerlendirme için belirlenen adımlara girdi alınmış ve sonuçta AKY’lar için Bakım Yapılabilirlik İndeksi (BYİ) elde edilmiştir. Burada yüzeysel olarak bahsedilen bu adımlar, Bölüm 6’da uygulaması ile birlikte detaylı olarak anlatılacaktır.

Çizelge 5.3 Güvenilirliğin değerlendirilmesi için kullanılan metriklerin eşik (İng. threshold) değerleri [37][38]

C&K Metrik	CBO	DIT	LCOM	NOC	RFC	WMC
Eşik (Threshold) değeri	3-9	1-6	1-3	1-3	6-36	6-30

Çizelge 5.4 Güvenilirliğin hesaplanması için kullanılan kurallar ve formüller
[37][38][39][40]

Kural 1:

Eğer (*alt eşik değer* ≤ *metrik değeri* ≤ *alt ve üst eşik değerlerin ortalaması*) ise,

Ağırlık =1

Kural 2:

Eğer (*alt ve üst eşik değerlerin ortalaması* ≤ *metrik değeri* ≤ *üst eşik değer*) ise,

Ağırlık =2

Kural 3:

Eğer (*Metrik değeri eşik değerlerinin dışında*) ise,

Ağırlık =7

Kural 4:

NOC için (*bu metrik diğer metriklerden ayrı değerlendirilecek*),

$G^E(Max)$ değeri hesaplanırken $\log((\text{üst eşik değeri})^2)$ değeri dikkate alınır.

$G^E(Min)$ değeri hesaplanırken $\log((\text{alt eşik değeri})^2)$ değeri dikkate alınır.

Kural 1'den kural 4'e doğru uygulandığında:

$$G^E(Max) = k * (1/AD(WMC) + AD(DIT) + AD(RFC) + AD(LCOM) + AD(CBO)) + (\log(\bar{U} - Lt(NOC)))^2 \quad (3)$$

$$G^E(Min) = k * (1/AD(WMC) + AD(DIT) + AD(RFC) + AD(LCOM) + AD(CBO)) + (\log(A - Lt(NOC)))^2 \quad (4)$$

$K = \text{sabit} = 1;$

$$G^E(Max) = 1 * (1/(1 + 1 + 1 + 1 + 1)) + (\log(3))^2 = 0,4276$$

$$G^E(Min) = 1 * (1/(2 + 2 + 2 + 2 + 2)) + (\log(1))^2 = 0,1000$$

Güvenilirlik eşik değeri (G^E) 0,1000 < G^E < 0,4276 arasındadır [37][38][39][40].

Dolayısıyla eğer güvenilirliğin eşik değeri bu değerler arasında ne kadar büyük ise ürün o kadar güvenilirdir. Eğer bu eşik değerleri dışında ise ürün hataya yatkındır.

Güvenilirliğin değerlendirilmesi aşamasında bakım yapılabilirlikle aynı metrik değerleri kullanılsa da farklı adımlar izlenmiştir. Güvenilirliğin ölçülebilmesi için literatürdeki çalışmalar araştırılmış [37][38][39][40] ve C&K metrik seti ile güvenilirliğin direkt olarak ölçülebildiği görülmüştür. İncelenen çalışmalar, yüksek atıf alan çalışmalara [59][60][61] dayanarak, güvenilirlikle C&K metrikleri arasında

ters orantılı bir ilişki olduğunu öngörmüştür. Değerlendirme yaparken bu [37][38][39] çalışmalar literatürde mevcut olan önceki çalışmaları [40][62][63][64][65][66][67][68][69] analiz etmişler ve C&K metrik serisi için Çizelge 5.3'de gösterildiği gibi yeni eşik (İng. threshold) değerleri hesaplamışlardır. Daha sonra güvenilirliğin hesaplanması için Çizelge 5.4'de gösterilen kuralları ve formülleri tanımlamışlardır [37][38][39]. Bu çalışmada güvenilirliğin değerlendirilmesi için aynı adımlar temel alınmıştır. Metrikler için Çizelge 5.3'de verilen eşik değerleri ve Çizelge 5.4'de verilen kurallar ve formüller uygulanarak AKY'lar için Güvenilirlik İndeksi (Gİ) elde edilmiştir. Burada yüzeysel olarak bahsedilen bu adımlar, Bölüm 6'da uygulaması ile birlikte detaylı olarak anlatılacaktır.

5.3.2. Toplum-tabanlı Değerlendirme

Aday AKY ürünleri ve değerlendirme yapılacak kalite öznitelikleri belirlendikten sonra toplum-tabanlı değerlendirme yapmak için bu belirlenen özniteliklerin ölçülmesinde bizlere yardım edecek bilgi ihtiyaçlarının belirlenmesi gerekir. Bu aşamada bilgi ihtiyaçlarının belirlenmesinden metriklerin elde edilmesine ve ölçümlerin yapılmasına kadarki süreçte, ölçümün amacına uygun ve gerçekçi olması için, uluslararası bir standart olan ISO/IEC 15939 ölçüm süreci kullanılmıştır. Çünkü standart hangi ölçüm bilgisinin gerekli olduğu, ölçümlerin ve analiz sonuçlarının nasıl uygulanacağı gibi detayların tanımlanması konusunda yol göstericidir.

Çizelge 5.5 Toplum-tabanlı değerlendirme için belirlenen bilgi ihtiyaçları ve metrikler

Bilgi ihtiyacı (İng: Information Need)	Tanımı ve amacı	AKY veri tabanlarından kullanılan metrikler
Yıllara göre elektronik posta yoğunluğu (yıllara göre ePY)	Ürünün durağan bir e-posta trafiğinin olup olmadığını, eğer mail trafiği durağanlaşmışsa bunun ne kadar zaman aldığını ölçer.	Geliştiriciler arasındaki toplam e-posta sayıları, geliştiricilerin sayısı, kod satır sayısı ve her bir ürünün sürüm sayıları
Hata çözme başarı indeksi (HÇBİ)	Geliştiricilerin raporlanan hataların çözümünde ne kadar başarılı olduğunu ölçer.	Toplam hata sayısı, hatanın durumu (çözülen), toplam kod satır sayısı
Hata ciddiyet indeksi (HCİ)	Diğer aday ürünlerle karşılaştırıldığında ürünün ne derece ciddi hatalarla karşılaştığını ölçer.	Her hatanın önem derecesi ve ürünün kod satır sayısı
Ortalama hata çözme süresine göre hata çözme başarı oranı (OHÇS)	Geliştiricilerin performansını ve oluşan sorunları, hangi ürün geliştiricilerin zamanında ve disiplin içinde çözdüğünü ölçer.	Hataların açılma zamanı (son iki yıldaki), hataların kapanma zamanı (son iki yıldaki), katkı sağlayanların sayısı (İng. number of contributors)
Ürün sürümlerine göre kümülatif hata yoğunluğu (SKHY)	Ürünlerin sürümlerine göre hataya eğilimli olup olmadığını ölçer.	Hata sayısı, toplam kod satır sayısı, her sürümün hata sayısı, her sürümün satır sayısı

Daha sonra Çizelge 5.5’de gösterildiği gibi bakım yapılabilirlik ve güvenilirliği ölçmek için beş tane bilgi ihtiyacı belirlenmiştir (Bilgi ihtiyaçlarının ISO/IEC 15939 temel alınarak oluşturulan ölçüm yapıları (İng. measurement construct) Ek 1’de verilmiştir). Bu bilgi ihtiyaçlarının ölçülebilmesi için her bir ürünün İnternet siteleri derinlemesine analiz edilmiş ve bu bilgilere nasıl ulaşılabileceği araştırılmıştır. Concurrent Version Control (CVS), Problem Reporting Database ve (BugDB) gibi

veri tabanları analiz edilmiş ve bilgi ihtiyaçlarını karşılamaya yönelik sonuçlar elde etmek için Çizelge 5.5'de verilen metrikler belirlenmiştir.

Bilgi ihtiyaçları ve metrikler belirlendikten sonra sonuçları elde etmek için adımlar belirlenmiştir. Öncelikle her bir bilgi ihtiyacını ölçmeye yönelik belirlenen metrik değerleri kullanılarak bilgi ihtiyaçları için indeks değerleri hesaplanmıştır. Genel sonuçların sağlıklı bir şekilde elde edilebilmesi için bilgi ihtiyaçları özelinde indeks değerlerinin hesaplama şeklinin, kod-tabanlı değerlendirmedeki hesaplama şekliyle tutarlı olması gerekmektedir. Diğer bir deyişle, kod-tabanlı değerlendirmede olduğu gibi en büyük indekse sahip ürün en tercih edilebilir ürün olmalıdır. Ne var ki HÇBİ ve OHÇS için indeks değerlerinin daha büyük olması ürünler için daha iyi bir durum iken, HÇİ, ePY ve SKHY için indeks değerlerinin daha küçük olması ürünler için daha iyi bir durumdur. Bu sebeple HÇİ, ePY ve SKHY için elde edilen indeks değerlerinin, genel sonuçların doğru hesaplanabilmesi için yeniden düzenlenmesi gerekmektedir. Bu düzenlemeyi yapmak amacıyla öncelikle, her bir ürün için hesaplanan HÇİ, ePY veya SKHY değerlerinin 1 ile bölünerek tersi alınmıştır. Denklem 5'deki gibi tersi alınarak elde edilen değerlerden en büyük olanının 10 tabanında logaritması alınmış ve elde edilen değer kendisinden büyük olan en küçük doğal sayıya yuvarlanmıştır. Daha sonra elde edilen bu en büyük doğal sayı (K), denklem 6'da yerine konularak her bir ürün için HÇİ, ePY veya SKHY bakımından yeni indeks değerleri hesaplanmıştır.

X_1, X_2, \dots, X_n değerlerinin sırayla 1., 2., ve n. ürün için HÇİ, ePY veya SKHY bakımından elde edilen indeks değerlerini simgelediği varsayıldığında:

$$K = \text{yukarıyuvarla}(\log_{10} \max(\frac{1}{x_1}, \frac{1}{x_2}, \dots, \frac{1}{x_n})) \quad (5)$$

L_i 'nin, HÇİ, ePY veya SKHY bakımından değerlendirme yapılan i . ürün için elde edilen yeni indeks değerini simgelediği varsayıldığında:

$$\text{Yeni indeks değeri} = L_i = \frac{1/x_i}{10^K} \quad (6)$$

Bilgi ihtiyaçları için indeksler elde edildikten sonra kullanıcı ihtiyacı doğrultusunda önem derecesine göre bu bilgi ihtiyaçlarına ağırlıklar verilmiştir. Hesaplanan indeks

değerleri ve bu ağırlık değerleri kullanılarak Toplum-tabanlı Değerlendirme İndeksi (TDİ) elde edilmiştir. Burada yüzeysel olarak bahsedilen bu adımlar, Bölüm 6'da uygulaması ile birlikte detaylı olarak anlatılacaktır.

5.4. Genel Sonuçların Elde Edilmesi

Her iki boyut için değerlendirme sonuçları ayrı ayrı elde edildikten sonra, kullanıcı boyutların önem derecesini dikkate alarak değerlendirme yapılan bu boyutlara ağırlıklar verir (Örneğin, kod-tabanlı değerlendirme %50 ve toplum-tabanlı değerlendirme %50). Daha sonra bu iki öznelik (bakım yapılabilirlik ve güvenilirlik) için ayrı ayrı, iki boyutun değerlendirme sonuçlarına göre en tercih edilebilir ürün seçilir. Bu genel sonuç şu formül kullanılarak hesaplanır:

$$\text{AKY iki boyutlu değerlendirme sonucu} = A * B + C * D$$

A: Kod-tabanlı boyut için hesaplanan genel değerlendirme indeksi

B: Kullanıcı tarafından kod-tabanlı boyuta verilen ağırlık

C: Toplum-tabanlı boyut için hesaplanan genel değerlendirme indeksi

D: Kullanıcı tarafından toplum-tabanlı boyuta verilen ağırlık

6. GELİŞTİRİLEN YÖNTEMİN UYGULANMASI VE SONUÇLARI

6.1. Kod-tabanlı Değerlendirme Sonuçları

Bu bölümde, geliştirilen metot bakım yapılabilirlik ve güvenilirlik için uygulanacak ve sonuçları gösterilecektir. Çizelge 5.1’de özellikleri gösterilen üç AKY ürününün (java tabanlı kod inşa aracı) kaynak kodlarının Çizelge 5.2’de listelenen metrik değerleri sınıf seviyesinde hesaplanmıştır. Bu hesaplama yapılırken Scitool firmasının Understand isimli kod analiz aracı kullanılmıştır. Her bir ürün için bu metrik değerlerinin ortalaması Çizelge 6.1’de verilmiştir.

Çizelge 6.1 Her bir AKY ürünü için metrik ölçüm sonuçları

Ürün	CBO	CC	DIT	LCOM	NNL	NOC	NOS	RFC	WMC
Archiva	3,241	1,618	1,399	2,338	0,477	1,424	21,401	12,443	17,901
Maven	2,039	2,259	1,519	3,162	0,681	1,224	28,152	7,081	20,623
Ant	3,312	2,005	1,886	2,739	0,697	1,632	21,713	22,142	26,354

6.1.1. Bakım Yapılabilirliğin Değerlendirilmesi ve Sonuçları

Bakım yapılabilirliği hesaplamak için; Çizelge 5.2’de verilen metriklerden bakım yapılabilirliğin alt öznelikleri ile ilişkisi olanlar, Bölüm 2.4’de verilen metriklerin açıklaması ve Çizelge 6.1’de verilen metrik değerlerinin ölçüm sonuçları kullanılmıştır. Bu değerler kullanılarak her bir alt öznelik bakımından her bir AKY ürünü 1’den 3’e doğru (1. en tercih edilebilir ürün) Çizelge 6.2’de gösterildiği gibi sıralanmıştır. Bu Çizelgede X, Y ve Z sırayla Archiva, Maven ve Ant AKY ürünlerini simgelemektedir. Çizelge 6.2’nin en sağındaki kolonda, her bir alt özneliğe göre ürünler en iyiden en kötüye doğru sıralanmıştır. Bunu yaparken Çizelgenin her bir satırındaki ürünlerin sıra değerleri dikkate alınmıştır. Örneğin, test edilebilirliği ölçmek için üç metrik değeri (RFC, NOC, NNL) kullanılmıştır. Bu üç metriğin ölçüm sonuçlarına göre, iki metrik için (RFC, NOC) Maven en iyi ürün olarak gözlemlenmiş ve bir tanesi için (NNL) Archiva en iyi ürün olarak gözlemlenmiştir. Dolayısıyla bu durumda test edilebilirlik için Maven en tercih edilebilir ürün olarak belirlenmiştir.

Sonuç olarak bakım yapılabilirliğin her bir alt özneliği için ürünler en iyiden en kötüye Çizelge 6.2'deki gibi sıralanmıştır.

Çizelge 6.2 Bakım yapılabilirliğin alt özneliklerine göre AKY ürünlerinin tercih edilme sıralaması

Alt öznelik	Sıra	WMC	DIT	RFC	NOC	NOS	NNL	CBO	LCOM	CC	Sonuç
Test edilebilirlik	1			Y	Y		X				Y (Maven)
	2			X	X		Y				X (Archiva)
	3			Z	Z		Z				Z (Ant)
Kararlılık	1		X				X	Y			X (Archiva)
	2		Y				Y	X			Y (Maven)
	3		Z				Z	Z			Z (Ant)
Değişebilirlik	1		X					Y	X		X (Archiva)
	2		Y					X	Z		Y (Maven)
	3		Z					Z	Y		Z (Ant)
Çözünümlenebilirlik	1	X				Y				X	X (Archiva)
	2	Y				Z				Z	Y (Maven)
	3	Z				X				Y	Z (Ant)

Ürünlere her bir alt öznelik için 1'den 3'e doğru sıralama verildikten sonra bu sıralama değerleri kullanılarak ürünlere Çizelge 6.3'te gösterildiği gibi ağırlıklar verilmiştir. Bu ağırlıklar verilirken ürünlere verilen sıra değerlerinin tersi kullanılmıştır. Mesela Çizelge 6.2'de 1. sırada olan ürüne ağırlık değeri olarak 3 ve 3. sırada olan ürüne ağırlık değeri olarak 1 verilmiştir. Değerlendirmenin bu kısmına kadar kullanıcı değerlendirmeye müdahale etmemiştir. Bu aşamadan sonra değerlendirici kendisine göre daha önemli gördüğü alt özneliklere ağırlıklar vermiştir. Çünkü bir değerlendirici için ürünün test edilebilirliği daha önemli olabilirken diğer bir değerlendirici için çözümlenebilirliği daha önemli olabilir. Bu doğrultuda sırayla çözümlenebilirlik, test edilebilirlik, değişebilirlik ve kararlılığa 1, 2, 3 ve 4 ağırlıkları atanmıştır. Burada bu ağırlıklandırma sisteminde verilen ağırlığın büyük olması, o özelliğin daha önemli olduğu anlamına gelmektedir.

Çizelge 6.3 Bakım yapılabilirlik indeksi (BYİ) ve hesaplanması

Bakım yapılabilirlik	Archiva ağırlık değeri $AD(X)$	Maven ağırlık değeri $AD(Y)$	Ant ağırlık değeri $AD(Z)$	Değerlendiricinin ağırlık değeri $AD(U)$
Test edilebilirlik	2	3	1	2
Kararlılık	3	2	1	4
Değişebilirlik	3	2	1	3
Çözümlenebilirlik	3	2	1	1
$AD(\dot{I}) * AD(U)$				
Test edilebilirlik	$2 * 2 = 4$	$3 * 2 = 6$	$1 * 2 = 2$	
Kararlılık	$3 * 4 = 12$	$2 * 4 = 8$	$1 * 4 = 4$	
Değişebilirlik	$3 * 3 = 9$	$2 * 3 = 6$	$1 * 3 = 3$	
Çözümlenebilirlik	$3 * 1 = 3$	$2 * 1 = 2$	$1 * 1 = 1$	
Toplam	28	22	10	
Toplam/ Mümkün olan en büyük ağırlık	28/48	22/48	10/48	
BYİ	0,583	0,458	0,208	

Bütün bu belirlenen ağırlıklar kullanılarak her bir ürün için Bakım Yapılabilirlik İndeksi (BYİ) hesaplanmıştır. BYİ hesaplanırken, her bir alt nitelik bakımından ürünler için belirlenen ağırlıklarla (Ağırlık Değeri(Ürün)- $AD(\dot{I})$), her bir alt özneliğe değerlendirici tarafından verilen ağırlıklar çarpılmıştır (Ağırlık Değeri(Kullanıcı)- $AD(U)$). Bu işlem her bir ürünün bütün alt öznelikleri için yapılmış ve sonuçlar toplanmıştır. Elde edilen değerler her bir ürün için elde edilebilecek en büyük değere (her alt öznelik için en büyük değer $max(AD(\dot{I}) * AD(U)) = 3 * 4 = 12$ ve toplam 4 alt nitelik olduğundan bakım yapılabilirlik için en büyük değer $12 * 4 = 48$ olur) bölünerek BYİ elde edilmiştir. Dolayısıyla Çizelge 6.3'te görüldüğü gibi Archiva,

Maven ve Ant AKY ürünleri için BYİ sırayla 0,583, 0,458 ve 0,208 olmuştur. (Bakım yapılabilirliğin her bir alt özneliği için ürünlerin genel değerlendirilmesi Ek-2’de radar diyagramı ile verilmiştir.)

6.1.2. Güvenilirliğin Değerlendirilmesi ve Sonuçları

Güvenilirliğin hesaplanması için bakım yapılabilirlikle aynı metriklerin ölçüm sonuçları (bkz. Çizelge 6.1) kullanılmasına rağmen farklı bir yol izlenmiştir. Güvenilirlik direkt olarak Çizelge 5.2’de gösterilen güvenilirlikle ilişkili metrikler kullanılarak değerlendirilmiştir. Bu değerlendirme için Çizelge 6.1’de gösterilen metrik ölçüm değerleri, Çizelge 5.3’te gösterilen eşik değerleri ve Çizelge 5.4’de gösterilen kurallar (kural 1’den kural 4’e) ve formüller kullanılarak hesaplanmıştır. Çizelge 6.4’de hangi ürünün hangi metrik değerine göre hangi kuralın uygulandığı ve bu kural çerçevesinde hangi ağırlık değerinin kullanıldığı gösterilmiştir.

Çizelge 6.4 Güvenilirlik indeksi (Gİ) hesaplanması için verilen ağırlıklar ve uygulanan kurallar

Ürün	CBO	DIT	LCOM	NOC	RFC	WMC	Gİ
Archiva	Ağırlık=1 (Kural1)	Ağırlık =1 (Kural 1)	Ağırlık =2 (Kural 2)	1,424 (Kural 4)	Ağırlık =1 (Kural 1)	Ağırlık =1 (Kural 1)	0,189
Maven	Ağırlık =7 (Kural 3)	Ağırlık =1 (Kural 1)	Ağırlık =7 (Kural 3)	1,224 (Kural 4)	Ağırlık =1 (Kural 1)	Ağırlık =2 (Kural 2)	0,062
Ant	Ağırlık =1 (Kural 1)	Ağırlık =1 (Kural 1)	Ağırlık =2 (Kural 2)	1,632 (Kural 4)	Ağırlık =2 (Kural 2)	Ağırlık =2 (Kural 2)	0,169

Çizelge 5.4’teki denklem (3) ve denklem (4) kullanılarak her bir ürün için, Güvenilirlik İndeksi (Gİ) şu şekilde hesaplanmıştır:

$$Archiva \ G\dot{I} = \frac{1}{(1 + 1 + 2 + 1 + 1)} + (\log(1,424))^2 = 0,189$$

$$Maven \ G\dot{I} = \frac{1}{(7 + 1 + 7 + 1 + 2)} + \log(1,224)^2 = 0,062$$

$$Ant \ G\dot{I} = \frac{1}{(1 + 1 + 2 + 2 + 2)} + \log(1,632)^2 = 0,169$$

Her bir AKY ürünü için Gİ incelendiğinde en güvenilir ürünün Archiva (Gİ=0,189) ve en çok hataya yatkın ürünün ise Maven (Gİ=0,062) olduğu görülmüştür.

6.2. Toplum-tabanlı Değerlendirme Sonuçları

Bakım yapılabilirlik ve güvenilirliğin toplum-tabanlı değerlendirilebilmesi için belirlenen bilgi ihtiyaçları, açıklamaları ve bu bilgi ihtiyaçlarının hesaplanmasında kullanılacak metrikler Çizelge 5.5’de özetlenmiştir. İzleyen alt kısımlarda her bir bilgi ihtiyacının gerekli metrikler kullanılarak nasıl ölçüldüğü anlatılacaktır.

6.2.1. Yıllara Göre Elektronik Posta Yoğunluğu (ePY)

AKY’ların geliştirilmesi süresince, kod geliştiricilerin kendi aralarındaki veya kullanıcılarla kod geliştiriciler arasındaki elektronik posta mesajları AKY ürünlerinin İnternet sitelerinin veri tabanlarında aylık olarak düzenli bir şekilde kaydedilir. Kod geliştirme yeteneği olan herkes bu mesajlaşma trafiğine katılabilir. Elektronik posta mesajları ürünün geçmişi, şu anki durumu ve geleceği hakkında bilgiler içerir. Çünkü bu mesajlar ürün kodunda yapılan değişimler, karşılaşılan hatalar, önerilen değişimler ve ürün hakkında teknik tartışmalar gibi birçok bilgi içerir. Bu çalışmada belirlenen AKY ürünlerinin elektronik posta listelerine erişebilmek için, Apache’nin elektronik posta listelerinin düzenli olarak kaydedildiği CVS [70] arşivi incelenmiştir.

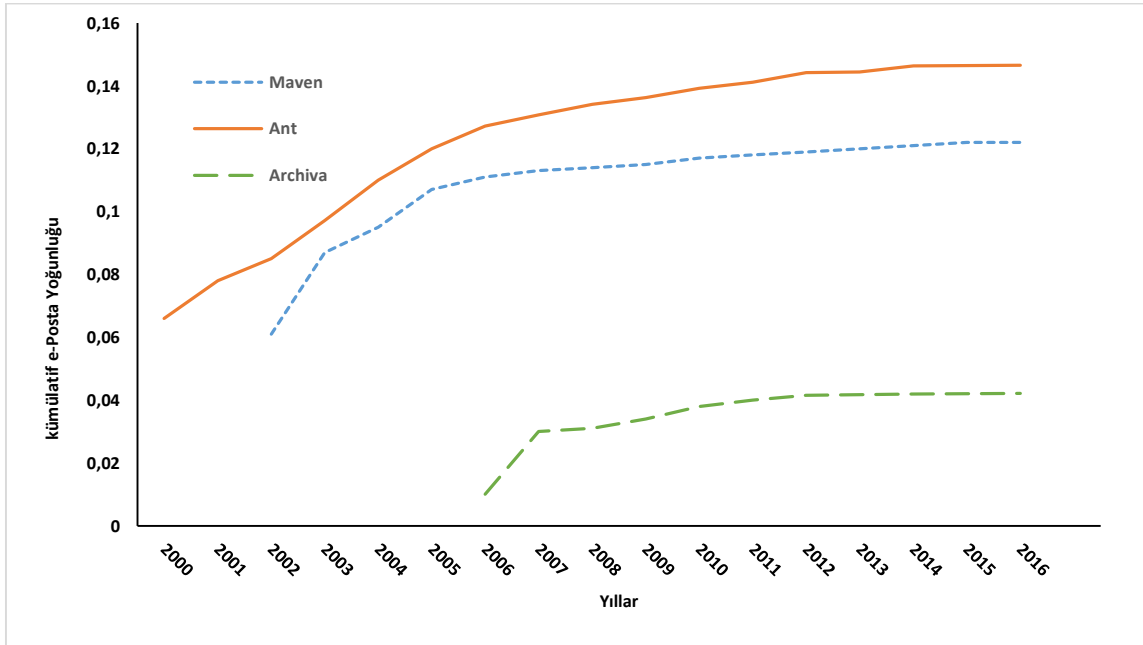
Bu bilgi ihtiyacını (yıllara göre ePY) hesaplamak için kod geliştiriciler arasındaki e-posta sayısı, her bir ürünün sürüm sayısı, kod satır sayısı ve kod geliştiricilerin sayısı kullanılmıştır. Ürünlerin sürüm sayıları, ürün üzerinde yapılan işlemlerin sayısı ile doğrudan ilişkilidir. Her bir yeni sürüm yeni özellikler ve önerilen yeni değişiklikler getireceği için, kod geliştiriciler arasındaki teknik tartışmaları da artıracaktır. Dolayısıyla kod geliştiriciler arasındaki elektronik posta sayıları da artacaktır. Kod geliştiriciler arasındaki elektronik posta sayıları temel alındığı için her bir üründeki kod geliştiricilerin sayısı da bu bilgi ihtiyacının hesaplanmasında önemlidir. Ayrıca her bir ürünün büyüklüğünün (kod satır sayısı) kod geliştiriciler arasındaki elektronik posta sayısı ile doğru olduğu varsayılabilir. Ürünün büyüklüğü arttığında ürünle ilgili problemler de artacağından e-posta sayısının da artması beklenir. Her bir ürün için ePY, toplam mail sayılarının toplam kod geliştirici sayılarına bölümünden elde edilen değerin (geliştirici başına mail sayısı), kod satır sayılarının toplam sürüm sayılarına bölümünden elde edilen değere (sürüm başına kod satır sayısı) bölünmesi ile hesaplanmıştır. Bu hesaplama sonucunda ePY için elde edilen değer ne kadar küçük ise bu, ürün için daha iyi bir durumdur. Ancak bölüm 5.3.2’de bahsedildiği gibi, hesaplamalar yapılırken büyük indeksteki ürünlerin daha tercih edilebilir olması

amaçlandığından, ePY için denklem 5 ve 6'nın uygulanması gerekmektedir. Denklemlerin uygulanması sonucunda değerlendirmede kullanılmak üzere elde edilen düzenlenmiş indeks değerleri Çizelge 6.5'deki gibi sırayla Archiva, Maven ve Ant için 0,6369, 0,1388 ve 0,1750 olmuştur.

Çizelge 6.5 E-posta yoğunluğu indeks değerleri

ePY	Archiva	Maven	Ant
Toplam e-posta satışı	17.395	112.776	87.438
Toplam sürüm sayısı	20	52	31
Kod geliştiricilerin sayısı	50	77	63
Kod Satır sayısı	439.346	716.983	387.863
e-posta yoğunluğu ($\frac{\#e-posta/\#kod\ geliştirci}{KSS/\#sürüm}$)	0,0157	0,0720	0,110
Düzenlenmiş indeks	0,636	0,138	0,090

Şekil 6.1'de yıllara göre e-posta yoğunluğu gösterilmiştir. Bu şekildeki her bir veri değeri elde edilirken her yıl için ayrı ayrı, Çizelge 6.5'de e-posta yoğunluğunu hesaplamak için verilen formül kullanılmış ve elde edilen sonuçlar kümülatif olarak toplanmıştır. Bu şekil Çizelge 6.5'de verilen indeks değerlerini desteklemek için elde edilmiştir.



Şekil 6.1 Yıllara göre kümülatif e-posta yoğunluğu

Şekil incelendiğinde, Archiva AKY ürününün yıllara göre kümülatif elektronik posta yoğunluğu iki yıldan sonra (üretildiği yıldan itibaren) istikrarlı bir tutum sergilemiş ve diğer iki ürüne göre hep düşük değerlerde seyretmiştir. Bu da yıllara göre bu ürün ile ilgili teknik tartışmaların diğer iki ürüne göre daha az olduğunun bir göstergesidir. Ant ürününün yıllara göre elektronik posta yoğunluğu diğer iki ürüne göre hep yükseklerde seyretmiştir. Yıllar içinde azalan bir seyir gösterse de ePY değerinin yüksek olması ürünle ilgili teknik tartışmaların diğer iki ürüne göre sürekli daha fazla olduğunun bir göstergesidir. Dolayısıyla bu bilgi ihtiyacı bakımından (ePY) Çizelge 6.5’de gösterildiği gibi, en iyi ürünün Archiva ve en kötü ürünün Ant olduğu görülmektedir.

6.2.2. Hata Çözme Başarı İndeksi (HÇBİ)

Yazılım hatası bilgisayar programının veya sistem akışının beklenmedik şekilde davranması veya yanlış ve beklenmedik sonuçlar üretmesidir [71]. Genellikle yazılımın kaynak kodunda veya tasarımında yapılan hatalardan meydana gelir.

Bu tez çalışmasında belirlenen üç aday AKY ürününün hata raporlarına ulaşmak için problem raporu veri tabanı (İng. bug reporting database-BugDB) kullanılmıştır. Bu veri tabanı kaydettiği her bir hata için hatanın açılmasından kapanmasına kadar birçok bilgiyi kaydeder (hata kimlik numarası, hatanın açıldığı ve kapandığı tarih, hatanın durumu (açıldı-çözüldü-kapandı), vb.).

Çizelge 6.6 Hata çözme başarı indeksi (HÇBİ)

Ürün	Kod satır sayısı	Toplam hata sayısı	Çözülen hata sayısı	Başarı oranı	HÇBİ (Başarı oranı/KLOC)
Archiva	439.346	1897	1663	%87,6	0,199
Maven	716.983	4914	4644	%94,5	0,131
Ant	387.863	5962	5135	%86,1	0,222

Çizelge 6.6’da, her bir ürün için toplam kod satır sayısı, toplam hata sayısı, toplam çözülen hata sayısı, hata çözme başarı oranı ve hata çözme başarı indeksi verilmiştir. Hata çözme başarı oranı, her bir ürün için çözülen hata sayısının toplam hata sayısına bölünmesiyle elde edilmiştir.

Hata çözüme başarı indeksi (HÇBİ), hata çözüme başarı oranının kilo kod satır sayısına (KKSS) bölünmesiyle hesaplanmıştır. Ürünlerin kod satır sayısı ne kadar fazla ise o kadar çok hatayla karşılaşılması ve hata çözüme başarı indeksinin (HÇBİ) düşük olması beklenir. Çizelgeye baktığımızda en uzun kod satır sayısına sahip olan Maven'in en düşük HÇBİ'ine (0,1319) ve en az kod satır sayısına sahip Ant ürününün en yüksek HÇBİ'ine (0,2220) sahip olduğu görülmektedir.

6.2.3. Hata Ciddiyet (İng. Bug Severity) İndeksi (HCİ)

Hata ciddiyeti yazılım ürünlerinde karşılaşılan hataların, ürün kalitesi üzerindeki negatif etki derecesine göre sınıflandırılmasıdır. Genellikle 5 seviyeden oluşur [72]:

Engelleyici (İng. blocker) hata: Programın normal işleyişini tamamen durdurur ve geçici çözümü yoktur.

Kritik (İng. critical) hata: Programdaki kritik fonksiyonları ve verileri etkiler. Geçici bir çözümü yoktur.

Büyük (İng. major) hata: Programdaki önemli fonksiyonları ve önemli verileri etkiler. Geçici bir çözümü vardır ancak kolay değildir.

Küçük (İng. minor) hata: Programdaki çok önemli olmayan fonksiyonları ve verileri etkiler. Geçici ve kolay bir çözümü vardır.

Önemsiz (İng. trivial) hata: Genellikle programdaki fonksiyonları ve verileri etkilemez. Geçici bir çözüm gerektirmez ve ürünün verimliliğini etkilemez.

Çizelge 6.7 Ürünlerin her bir hata seviyesindeki indeksleri ve HCİ

Ürün	Engelleyici	Kritik	Büyük	Küçük	Önemsiz	HCİ	Düzenlenmiş İndeks
Archiva	0,00004	0,00002	0,0034	0,005	0,00004	0,01904	0,520
Maven	0,0001	0,0002	0,005	0,0011	0,0002	0,0308	0,320
Ant	0,002	0,0021	0,0056	0,0049	0,0006	0,076	0,013

Bu bilgi ihtiyacının ölçülmesi için problem raporu veri tabanında (İng. bug reporting database-BugDB), bulunan her bir hatanın önem derecesi incelenmiştir. Ürünlerin her bir seviyedeki hata sayıları ayrı ayrı bulunmuş ve bu hata sayıları ürünlerin kod satır sayısına bölünerek Çizelge 6.7'de gösterildiği gibi değerler elde edilmiştir. Her

bir seviyedeki hatanın ürün üzerindeki etkisi farklı olacaktır. Bu yüzden hatalara önem derecelerine göre ağırlıklar verilmiştir (engelleyici = 9, kritik = 7, büyük = 5, küçük = 3, önemsiz = 1).

Her bir ürünün HCl değerinin hesaplanabilmesi için, Çizelge 6.7’de gösterilen hata önem derecesi değerleri ile önem derecesinin ağırlıkları çarpılmış ve çarpımlar aşağıda gösterildiği gibi toplanmıştır:

$$Archiva = (0,00004 * 9) + (0,00002 * 7) + (0,0034 * 5) + (0,0005 * 3) + (0,00004 * 1) = 0,01904$$

$$Maven = (0,0001 * 9) + (0,0002 * 7) + (0,005 * 5) + (0,0011 * 3) + (0,0002 * 1) = 0,0308$$

$$Ant = (0,002 * 9) + (0,0021 * 7) + (0,0056 * 5) + (0,0049 * 3) + (0,0006 * 1) = 0,076$$

HCl değerlerine baktığımızda Ant AKY ürününün piyasaya sürüldüğünden itibaren diğer ürünlere göre daha ciddi hatalarla karşılaştığı görülmektedir. Archiva AKY ürününün ise diğer ürünlere nazaran daha önemsiz hatalarla karşılaştığı gözlemlenmiştir. HCl için elde edilen değer ne kadar küçük ise bu, ürün için daha iyi bir durumdur. Ancak Bölüm 5.3.2’de bahsedildiği gibi, hesaplamalar yapılırken büyük indeksteki ürünlerin daha tercih edilebilir olması amaçlandığından HCl için denklem 5 ve 6’nın uygulanması gerekmektedir. Denklemlerin uygulanması sonucunda değerlendirmede kullanılmak üzere elde edilen düzenlenmiş indeks değerleri, çizelgedeki gibi sırayla Archiva, Maven ve Ant için 0,52, 0,32 ve 0,13 olmuştur.

6.2.4. Ortalama Hata Çözme Süresine Göre Hata Çözme Başarı Oranı (OHÇS)

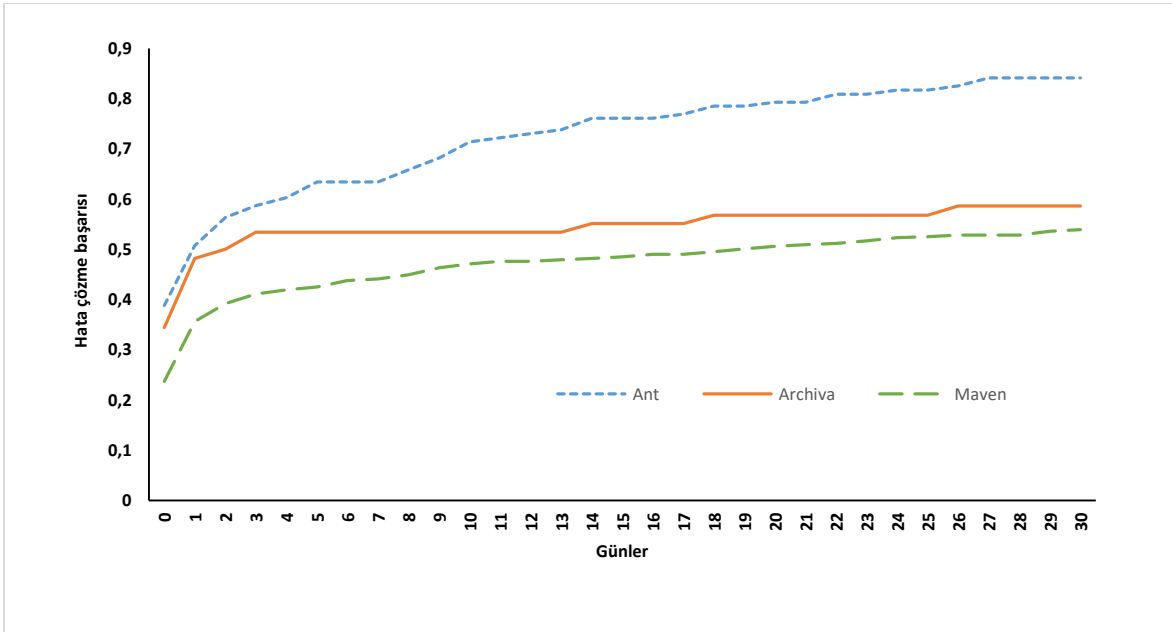
AKY ürünlerinin kalitesinin hesaplanmasında ürün bünyesinde çalışan kod geliştiriciler çok önemlidir. Çünkü hangi ürünün geliştirici takımı karşılaşılan problemleri çözmede daha çabuk ve disiplinli ise o ürün daha başarılı ve güvenilirdir. Bu bilgi ihtiyacı, direkt olarak ürünlerdeki kod geliştiricilerin performansını ölçmektedir.

Bu çalışmada OHÇS değerini ölçmek için son iki yıldaki hatalar göz önünde bulundurulmuştur. Dolayısıyla bu bize üründe güncel olarak çalışan kod geliştiricilerin başarısını yansıtacaktır. Bu yüzden son iki yılda açılan hataların açılış

tarihi, kapanış tarihi ve bu ürünlere katkı sağlayanların sayısı (İng. number of contributors) bu bilgi ihtiyacını ölçmek için kullanılacaktır.

Çizelge 6.8 Son iki yılda açılan hataların çözülmesi ile ilgili bilgiler

Ürün	Archiva	Maven	Ant
Son iki yıldaki OHÇS	112 gün	106 gün	21 gün
Katkı sağlayanların sayısı	50	77	63
Son iki yılda açılan hataların açıldığı günden itibaren 30 gün içinde çözülme oranı	0,550	0,450	0,850



Şekil 6.2 Son iki yılda açılan hataların açıldığı günden itibaren 30 gün içerisinde çözülme oranları

Çizelge 6.8'de son iki yılda açılan hataların her bir ürün için ortalama kaç günde çözüldüğü ve her bir üründeki katkı sağlayıcıların sayısı verilmiştir. Şekil 6.2'de her bir ürün için son iki yılda açılan toplam hata sayılarının açıldıktan itibaren 30 gün içerisinde çözülme oranları gösterilmektedir (örneğin, bir hata 10.Kasım.2015 de açılmış ve 11.Kasım.2015 kapatılmış ise bu 1 gün içinde çözülmüştür). Son iki yılda açılan hataların çözülme oranlarına baktığımızda, Ant ürününün açılan hataların

yüzde 40'ını (0,4), Archiva'nın yüzde 35'ini (0,35) ve Maven'in yüzde 25'ini (0,25) açıldığı gün çözdüğü görülmüştür.

Genel olarak baktığımızda, hem Çizelge 6.8'de hem de Şekil 6.2'de gösterildiği gibi, son iki yıl içinde açılan toplam hataların Ant ürünü için %85'i (0,85), Archiva ürünü için %55'i (0,55) ve Maven ürünü için %45'i (0,45) açıldıktan itibaren 30 gün içerisinde çözülmüştür. Dolayısıyla OHÇS bakımından indeks değerleri Archiva, Maven ve Ant için sırayla 0,55, 0,45 ve 0,85 olmuştur.

Sonuçlar incelendiğinde OHÇS bakımından Ant AKY ürünü en başarılı ürün olmuştur. Çizelge 6.8'de son iki yılda açılan hataların çözülmesi için gerekli ortalama süre bakımından Maven ürünü ile Archiva ürünü birbirine yakındır. Fakat Şekil 6.2 incelendiğinde Archiva ürününün kod geliştiricisi sayısının daha az olmasına rağmen genel anlamda hataları çözme oranında daha başarılı olduğu görülmüştür.

6.2.5. Ürün Sürümlerine Göre Kümülatif Hata Yoğunluğu (SKHY)

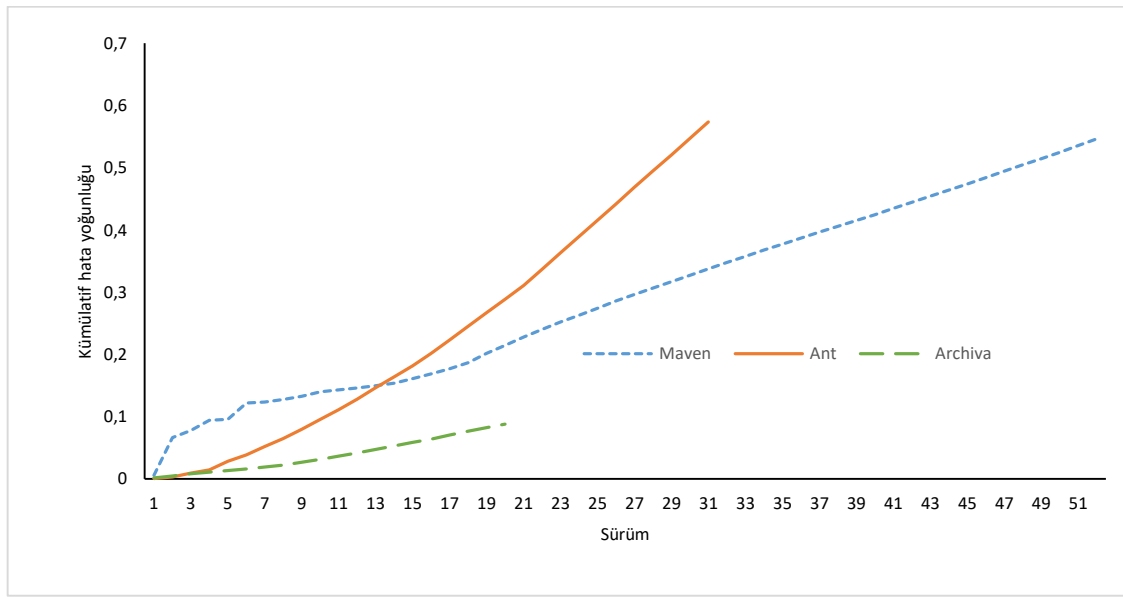
Hata yoğunluğu (İng. defect density) bir sistemde veya bileşende bulunan toplam hata sayısının o sistemin büyüklüğüne oranı olarak tanımlanır [23]. Hata yoğunluğunu azaltmak büyük projelerde zaman, maliyet ve kaliteyi dengede tutmada önemli bir yere sahiptir.

Burada belirlenen bilgi ihtiyacını (SKHY) ölçmek için, ürünlerin toplam hata sayıları ve son sürümlerinin toplam kod satır sayıları kullanılmıştır. Ürünler piyasaya sürüldüklerinden itibaren son sürümlerindeki toplam hata sayılarının toplam kod satır sayısına bölünmesi ile elde edilen hata yoğunluğu (HY) değerleri Çizelge 6.9'da verilmiştir. Çizelgedeki değerlere baktığımızda bu metrik değerine göre en tercih edilebilir ürün Archiva ve en hata eğilimli ürün ise Ant olarak görülmektedir. Ancak Bölüm 5.3.2'de bahsedildiği gibi tüm bilgi ihtiyaçları için elde edilen indekslerin belli bir standartta olması gerekir. SKHY için elde edilen değer ne kadar küçük ise bu, ürün için daha iyi bir durumdur. Hesaplamalar yapılırken büyük indeksteki ürünlerin daha tercih edilebilir olması amaçlandığından SKHY için denklem 5 ve 6'nın uygulanması gerekmektedir. Denklemlerin uygulanması sonucunda değerlendirmede kullanılmak üzere elde edilen düzenlenmiş indeks değerleri çizelgedeki gibi sırayla Archiva, Maven ve Ant için 0,232, 0,212 ve 0,065 olmuştur.

Çizelge 6.9 Hata yoğunluğu

Ürün	Archiva	Maven	Ant
Hata yoğunluğu	0,0043	0,0047	0,0153
Düzenlenmiş indeks	0,232	0,212	0,065

Bu durumu desteklemek için her bir ürünün hata yoğunluğu Şekil 6.3'deki gibi sürüm bazında incelenmiştir. Şekilde her bir ürün için üretildiği yıldan itibaren, her bir sürümde o ana dek bulunan hata sayısı, o sürümün toplam kod satır sayısına bölünmüştür. Her bir sürüm için önceki sonuçlar, kümülatif olarak toplanmıştır.



Şekil 6.3 Sürümlere göre kümülatif hata yoğunluğu

Şekil incelendiğinde sürümler bazında Ant yazılım ürününün hata yoğunluğunun yüksek seviyede seyrettiği gözlemlenmiştir. Dolayısıyla hem Çizelge 6.9'da hem de Şekil 6.3'de gösterildiği gibi, SKHY bakımından en iyi ürün Archiva ve en kötü ürün Ant olmuştur.

6.2.6. Toplum-tabanlı Değerlendirme İçin Genel Sonuçlar

Toplum-tabanlı değerlendirmede, bakım yapılabilirlik ve güvenilirlik için belirlenen bilgi ihtiyaçları, seçilen metrikler doğrultusunda ölçülmüştür. Ölçüm sonuçları değerlendirilerek her bilgi ihtiyacı için indeksler elde edilmiştir. Bu indeksler metrik değerlerine göre yapılan ölçüm sonucunda elde edildiğinden öznellik içermemektedir.

Genel sonuca ulaşabilmek için değerlendirici belirlediği bilgi ihtiyaçlarına kendi açısından önem derecesine göre ağırlıklar verir. Bir kullanıcı için üründe çalışan kod geliştiricilerin başarısı (OHÇS) daha önemliken diğer bir kullanıcı için ürünün piyasaya sürüldüğünden beri ne derece önemli hatalarla karşılaştığı (HCİ) bilgisi daha önemli olabilir. Bu çalışmada ePY, HÇBİ, HCİ, OHÇS ve SKHY bilgi ihtiyaçlarına sırayla 4, 2, 3, 1 ve 5 ağırlık değerleri verilmiştir (ağırlık ne kadar büyükse bilgi ihtiyacı kullanıcıya göre o kadar önemlidir).

Çizelge 6.10 Toplum-tabanlı değerlendirme indeksi (TDİ) ve hesaplanması

Bilgi ihtiyacı	Archiva ağırlık değeri $AD(X)$	Maven ağırlık değeri $AD(Y)$	Ant ağırlık değeri $AD(Z)$	Değerlendiricinin ağırlık değeri $AD(U)$
ePY	0,636	0,138	0,090	4
HÇBİ	0,199	0,131	0,222	2
HCİ	0,520	0,320	0,013	3
OHÇS	0,550	0,450	0,850	1
SKHY	0,232	0,212	0,065	5
$AD(i) * AD(U)$				
ePY	$0,636 * 4 = 2,544$	$0,138 * 4 = 0,552$	$0,090 * 4 = 0,360$	
HÇBİ	$0,199 * 2 = 0,398$	$0,131 * 2 = 0,262$	$0,222 * 2 = 0,444$	
HCİ	$0,520 * 3 = 1,560$	$0,320 * 3 = 0,960$	$0,013 * 3 = 0,039$	
OHÇS	$0,550 * 1 = 0,550$	$0,450 * 1 = 0,450$	$0,850 * 1 = 0,850$	
SKHY	$0,232 * 5 = 1,160$	$0,212 * 5 = 1,060$	$0,065 * 5 = 0,325$	
Toplam	6,212	3,284	2,018	
Toplam/ Mümkün olan en büyük ağırlık	$6,212/25$	$3,284/25$	$2,018/25$	
TDİ	0,248	0,131	0,080	

Belirlenen ağırlıklar sonucunda her bir ürün için Çizelge 6.10'da gösterildiği gibi Toplum-tabanlı Değerlendirme İndeksi (TDİ) hesaplanmıştır. TDİ hesaplanırken, her bir bilgi ihtiyacı bakımından ürünler için ölçülen ağırlıklarla (ağırlık değeri(Ürün)-AD(I)), her bir bilgi ihtiyacına değerlendirici tarafından verilen ağırlıklar çarpılmıştır (ağırlık değeri(Kullanıcı)-AD(U)). Bu işlem her bir ürünün bütün bilgi ihtiyaçları için yapılmış ve sonuçlar toplanmıştır. Elde edilen değerler her bir ürün için elde edilebilecek en büyük değere (her bilgi ihtiyacı için en büyük değer $max(AD(I) * AD(U)) = 1 * 5 = 5$ ve toplam 5 bilgi ihtiyacı olduğundan toplum-tabanlı değerlendirme için en büyük değer $5 * 5 = 25$ olur) bölünerek TDİ elde edilmiştir. Dolayısıyla Çizelge 6.10'da görüldüğü gibi Archiva, Maven ve Ant AKY ürünleri için TDİ sırayla 0,247, 0,129 ve 0,080 olmuştur. (Toplum-tabanlı değerlendirmede kullanılan bilgi ihtiyaçları için ürünlerin genel değerlendirmesi Ek-2'de radar diyagramı ile verilmiştir.)

6.3. Genel Değerlendirme Sonuçları

Bakım yapılabilirlik ve güvenilirlik için değerlendirmeler kod-tabanlı ve toplum-tabanlı olmak üzere ayrı ayrı yapılmış ve sonuçlar Çizelge 6.11'de gösterilmiştir. İki boyutlu değerlendirme sonucunda bakım yapılabilirlik ve güvenilirlik için ayrı ayrı en iyi ürünü belirlemeden önce, değerlendirici bu boyutlara da ağırlıklar verir. Çünkü değerlendirme boyutlarının önem derecesi kullanıcıya göre değişkenlik gösterebilir. Bu çalışmada kod-tabanlı değerlendirmeye %65 ve toplum-tabanlı değerlendirmeye %35 ağırlık verilmiştir.

Çizelge 6.11 Kod-tabanlı ve toplum-tabanlı değerlendirme sonuçları

Değerlendirme boyutu	İndeksler	Archiva	Maven	Ant	Değerlendiricinin ağırlık değeri
Kod-tabanlı	BYİ	0,583	0,458	0,208	%65
	Gİ	0,189	0,062	0,169	
Toplum-tabanlı	TDİ	0,248	0,131	0,080	%35

Çizelge 6.11'deki indeksler göz önüne alındığında, Bölüm 5.4'te verilen genel değerlendirme sonuçlarının elde edilmesi formülüne göre, sonuçlar aşağıdaki gibidir:

$$AKY \text{ iki boyutlu deęerlendirme sonucu} = A * B + C * D$$

A: Kod-tabanlı boyut için hesaplanan genel deęerlendirme indeksi

B: Kullanıcı tarafından kod-tabanlı boyuta verilen aęırlık

C: Toplum-tabanlı boyut için hesaplanan genel deęerlendirme indeksi

D: Kullanıcı tarafından toplum-tabanlı boyuta verilen aęırlık

Bakım yapılabilirlik için,

$$Apache Archiva = (0,583 * 0,65) + (0,248 * 0,35) = 0,464$$

$$Apache Maven = (0,458 * 0,65) + (0,131 * 0,35) = 0,342$$

$$Apache Ant = (0,208 * 0,65) + (0,080 * 0,35) = 0,163$$

Güvenilirlik için,

$$Apache Archiva = (0,189 * 0,65) + (0,248 * 0,35) = 0,208$$

$$Apache Maven = (0,062 * 0,65) + (0,131 * 0,35) = 0,085$$

$$Apache Ant = (0,169 * 0,65) + (0,080 * 0,35) = 0,137$$

Bu tez çalışmasında geliştirilen metodun deęerlendirme sonuçlarına göre, hem bakım yapılabilirlik (0,464) için hem de güvenilirlik (0,208) için en büyük indekse sahip Apache Archiva AKY ürünü en tercih edilebilir ürün olmuştur.

7. ÜRÜNLERİN GERİYE DÖNÜK DEĞERLENDİRİLMESİ VE SONUÇLARIN KARŞILAŞTIRILMASI

Bu bölümde önerilen metodun tutarlılığını ölçmek için geçmişe dönük değerlendirme yapılmıştır. Bir önceki bölümde kullanıcı belirlediği aday AKY ürünleri arasından ihtiyacına en uygun olanını ürünlerin güncel verilerini kullanarak seçmiştir. Bu bölümde ise aynı kullanıcının 1 Haziran 2013 yılında aynı aday AKY ürünleri arasından ihtiyaçları doğrultusunda seçim yaptığı varsayılmış ve sonuçları analiz edilmiştir. Değerlendirmenin yapılması için ürünlerin piyasaya sürüldüğü günden itibaren 1 Haziran 2013 yılına kadar olan verileri temel alınmıştır. Hem kod-tabanlı hem de toplum-tabanlı değerlendirmeler yapılırken güncel olarak yapılan değerlendirmelerde kullanılan metriklerle aynı metrikler kullanılmış ve aynı şekilde ölçüm yapılmıştır.

7.1. Geriye Dönük Kod-tabanlı Değerlendirme Sonuçları

Bu bölümde değerlendirme yapmak için her bir AKY ürününün 1 Haziran 2013 yılı itibarıyla son sürümlerinin kaynak kodları kullanılmıştır. Bölüm 6.1'deki ürünlerin güncel kaynak kodlarıyla yapılan ölçüm, bu bölümde aynı şekilde sınıf seviyesinde ve Understand Scitool aracı kullanılarak yapılmıştır. Her bir ürün için C&K metrik değerlerinin ortalaması Çizelge 7.1'de verilmiştir.

Çizelge 7.1 Her bir AKY ürünü için metrik ölçüm sonuçları (2013 verilerine göre)

	CBO	CC	DIT	LCOM	NNL	NOC	NOS	RFC	WMC
Archiva	3,121	1,575	1,151	1,794	0,462	1,442	21,667	12,564	17,106
Maven	3,032	1,690	1,294	1,357	0,554	1,397	18,164	9,920	20,210
Ant	3,224	1,982	2,101	2,120	0,677	1,721	20,913	25,289	26,281

7.1.1. Bakım Yapılabilirliğin Geriye Dönük Değerlendirilmesi ve Sonuçları

Bu bölümde bakım yapılabilirliğin hesaplanması için Çizelge 5.2’de verilen metriklerden bakım yapılabilirliğin alt öznelikleri ile ilişkili olanlar, Bölüm 2.4’de verilen metriklerin açıklaması ve Çizelge 7.1’de verilen metrik değerlerinin ölçüm sonuçları kullanılmıştır. Bölüm 6.1.1’de yapılan değerlendirmenin aynısı uygulanarak her bir alt öznelik için en tercih edilebilir ürünler 1 den 3’ doğru Çizelge 7.2’de sıralanmıştır (bkz. Bölüm 6.1.1).

Çizelge 7.2 Bakım yapılabilirliğin alt özneliklerine göre AKY ürünlerinin tercih edilme sıralaması (2013 yılı verilerine göre)

Alt öznelik	Sıra	WMC	DIT	RFC	NOC	NOS	NNL	CBO	LCOM	CC	Sonuç
Test edilebilirlik	1			Y	Y		X				Y (Maven)
	2			X	X		Y				X (Archiva)
	3			Z	Z		Z				Z (Ant)
Kararlılık	1		X				X	Y			X (Archiva)
	2		Y				Y	X			Y (Maven)
	3		Z				Z	Z			Z (Ant)
Değişebilirlik	1		X					Y	Y		Y (Maven)
	2		Y					X	X		X (Archiva)
	3		Z					Z	Z		Z (Ant)
Çözümenebilirlik	1	X				X				X	X (Archiva)
	2	Y				Z				Z	Z (Ant)
	3	Z				Y				Y	Y (Maven)

Bölüm 6.1.1’de kullanılan ağırlıklandırma algoritması bu bölümdeki değerlendirmede de kullanılmış ve bakım yapılabilirliğin alt özneliklerine kullanıcı tarafından yine ağırlık değerleri verilmiştir (Aynı kullanıcının aynı ihtiyaçla değerlendirme yaptığı varsayıldığı için sırayla çözümlenebilirlik, test edilebilirlik, değişebilirlik ve kararlılığa 1, 2, 3 ve 4 ağırlıkları atanmıştır).

Çizelge 7.3 Geriye dönük bakım yapılabilirlik indeksi (GD-BYİ) ve hesaplanması

Bakım yapılabilirlik	Archiva ağırlık değeri $AD(X)$	Maven ağırlık değeri $AD(Y)$	Ant ağırlık değeri $AD(Z)$	Değerlendiricinin ağırlık değeri $AD(U)$
Test edilebilirlik	2	3	1	2
Kararlılık	3	2	1	4
Değişebilirlik	2	3	1	3
Çözömlenebilirlik	3	1	2	1
$AD(i) * AD(U)$				
Test edilebilirlik	$2 * 2 = 4$	$3 * 2 = 6$	$1 * 2 = 2$	
Kararlılık	$3 * 4 = 12$	$2 * 4 = 8$	$1 * 4 = 4$	
Değişebilirlik	$2 * 3 = 6$	$3 * 3 = 9$	$1 * 3 = 3$	
Çözömlenebilirlik	$3 * 1 = 3$	$1 * 1 = 1$	$2 * 1 = 1$	
Toplam	25	24	11	
Toplam/ Mümkün olan en büyük ağırlık	25/48	24/48	11/48	
GD-BYİ	0,520	0,500	0,229	

Çizelge 7.3'de gösterilen bütün ağırlık değerleri kullanılarak her bir ürün için 1 Haziran 2013 yılına ait Geriye Dönük Bakım Yapılabilirlik İndeksi (GD-BYİ) hesaplanmıştır. Hesaplamanın nasıl yapıldığı Bölüm 6.1.1'de detaylı bir şekilde anlatılmıştır. Sonuçlara bakıldığında Archiva, Maven ve Ant AKY ürünleri için GD-BYİ değerleri sırayla 0,520, 0,500 ve 0,229 olmuştur.

7.1.2. Güvenilirliğin Geriye Dönük Değerlendirilmesi ve Sonuçları

Güvenilirliğin hesaplanması için Bölüm 6.1.2'deki metrikler ve yöntem kullanılmıştır. Metrik değerleri olarak Çizelge 7.1'de verilen ölçüm sonuçları kullanılmıştır. Bölüm 6.1.2'de olduğu gibi Çizelge 5.3'deki eşik değerleri ve Çizelge 5.4'deki kurallar

(kural1'den kural 4'e) ve formüller kullanılmıştır. Çizelge 7.4'de hangi ürünün hangi metrik değerlerine göre hangi kuralın uygulandığı ve bu kural çerçevesinde hangi ağırlık değerinin kullanıldığı gösterilmiştir. Bütün bu değerler kullanılarak her bir ürün için Geriye Dönük Güvenilirlik İndeksi (GD-Gİ) elde edilmiştir.

Çizelge 7.4 Geriye dönük güvenilirlik indeksi (GD-Gİ) hesaplanması için verilen ağırlıklar ve uygulanan kurallar

Product	CBO	DIT	LCOM	NOC	RFC	WMC	Gİ
Archiva	Ağırlık=1 (Kural1)	Ağırlık =1 (Kural 1)	Ağırlık =2 (Kural 2)	1,442 (Kural 4)	Ağırlık =1 (Kural 1)	Ağırlık =1 (Kural 1)	0,224
Maven	Ağırlık =1 (Kural 1)	Ağırlık =1 (Kural 1)	Ağırlık =1 (Kural 1)	1,397 (Kural 4)	Ağırlık =1 (Kural 1)	Ağırlık =2 (Kural 2)	0,187
Ant	Ağırlık =1 (Kural 1)	Ağırlık =1 (Kural 1)	Ağırlık =2 (Kural 2)	1,721 (Kural 4)	Ağırlık =2 (Kural 2)	Ağırlık =2 (Kural 2)	0,180

Bu kurallar çerçevesinde Çizelge 5.4'deki denklem (3) ve (4) kullanılarak her bir ürün için Geriye Dönük Güvenilirlik İndeksi (GD-Gİ) şu şekilde hesaplanmıştır:

$$\text{For Archiva} = 1/(1 + 1 + 1 + 1 + 1) + (\log(1,44))^2 = 0,224$$

$$\text{For Maven} = 1/(2 + 1 + 1 + 1 + 1) + (\log(1,397))^2 = 0,187$$

$$\text{For Ant} = 1/(2 + 1 + 2 + 2 + 1) + (\log(1,721))^2 = 0,180$$

Her bir AKY ürünü için GD-Gİ incelendiğinde en güvenilir ürünün Archiva (GD-Gİ=0,224) ve en çok hataya yatkın ürünün ise Ant (GD-Gİ=0,180) olduğu görülmüştür.

7.2. Geriye Dönük Toplum-tabanlı Değerlendirme Sonuçları

Bu bölümde, bakım yapılabilirlik ve güvenilirliğin toplum-tabanlı değerlendirmesinin yapılabilmesi için, Çizelge 5.5'de verilen bilgi ihtiyaçları kullanılmıştır. Bu bilgi ihtiyaçlarının değerlendirilmesinde Bölüm 6.2'deki ile aynı yol izlenmiştir. Ancak bilgi ihtiyaçlarının ölçülmesi için kullanılan metrikler hesaplanırken ürünün piyasaya sürüldüğü tarihten 1 Haziran 2013 tarihine kadar olan veriler temel alınmıştır. İzleyen alt bölümlerde, belirlenen bilgi ihtiyaçlarının geriye dönük ölçüm sonuçları anlatılacaktır.

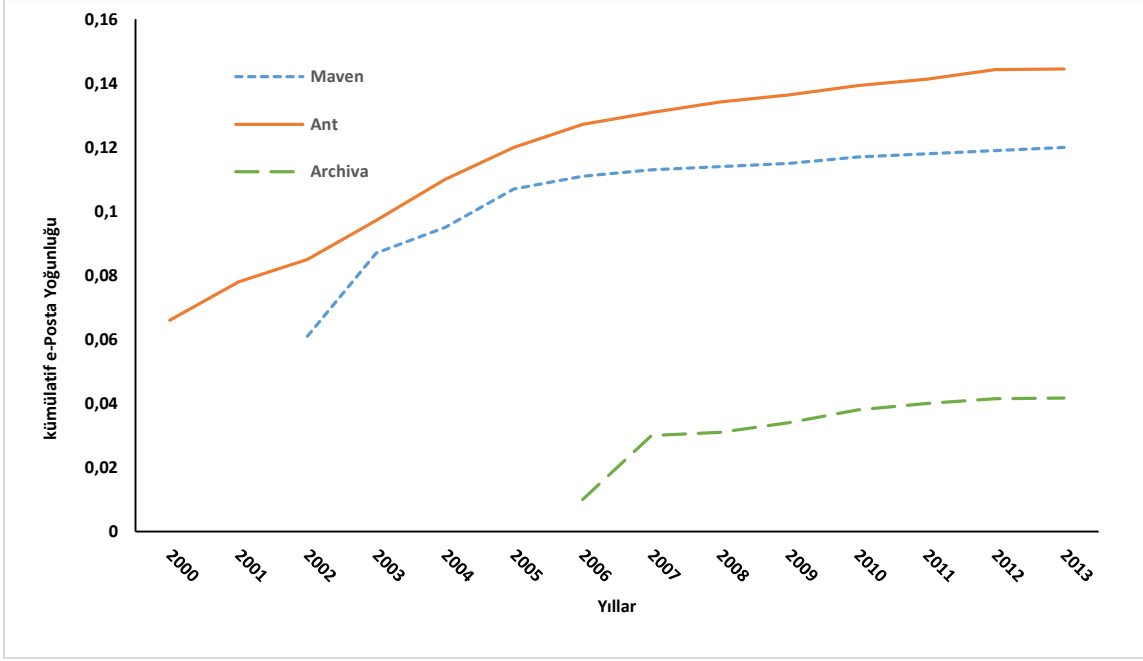
7.2.1. Yıllara Göre Geriye Dönük Elektronik Posta Yoğunluğu (GD-ePY)

Bu bilgi ihtiyacının karşılanması için CVS arşivindeki e-posta listeleri kullanılmıştır (detaylı bilgi için bkz. Bölüm 6.2.1). Bölüm 6.2.1’de güncel e-posta arşivi ile yapılan ölçümler bu bölümde 1 Haziran 2013 tarihine kadar olan e-posta arşivi temel alınarak yapılmıştır. Her bir ürün için GD-ePY, 1 Haziran 2013 yılına kadarki veriler temel alınarak, toplam mail sayılarının toplam kod geliştirici sayılarına bölümünden elde edilen değer (geliştirici başına mail sayısı), kod satır sayısının toplam sürüm sayısına bölümünden elde edilen değere (sürüm başına kod satır sayısı) bölünmesi ile hesaplanmış ve sonuçlar Çizelge 7.5’de gösterilmiştir. Bu hesaplama sonucu ePY için elde edilen değer ne kadar küçük ise bu, ürün için daha iyi bir durumdur. Ancak bölüm 5.3.2’de bahsedildiği gibi hesaplamalar yapılırken büyük indeksteki ürünlerin daha tercih edilebilir olması gerektiğinden GD-ePY için denklem 5 ve 6’nın uygulanması gerekmektedir. Denklemlerin uygulanması sonucu hesaplamada kullanılmak üzere elde edilen düzenlenmiş indeks değerleri Çizelge 7.5’deki gibi sırayla Archiva, Maven ve Ant için 0,8196, 0,1086 ve 0,0957 olmuştur.

Çizelge 7.5 Geriye dönük e-posta yoğunluğu indeks değerleri

GD-ePY	Arhiva	Maven	Ant
Toplam e-posta sayısı	13.795	90.194	86.117
Toplam sürüm sayısı	17	43	25
Kod geliştiricilerin sayısı	44	61	55
Kod satır sayısı	432.791	690.804	374.548
e-posta yoğunluğu($\frac{\#e-posta/\#kod\ geliştirici}{KSS/\#sürüm}$)	0,0122	0,092	0,1044
Düzenlenmiş indeks	0,819	0,108	0,095

Şekil 7.1’de yıllara göre e-posta yoğunluğu gösterilmiştir. Bu şekildeki her bir veri değeri elde edilirken, her yıl için ayrı ayrı Çizelge 7.5’de e-posta yoğunluğunu hesaplamak için verilen formül kullanılmış ve elde edilen sonuçlar kümülatif olarak toplanmıştır. Bu şekil Çizelge 7.5’de verilen indeks değerlerini desteklemek için oluşturulmuştur.



Şekil 7.1 Yıllara göre kümülatif e-posta yoğunluğu (2013 yılına kadar olan verilere göre)

Şekil incelendiğinde, Archiva AKY ürününün sürüm bazında elektronik posta yoğunluğunun iki yıldan sonra (üretildiği yıldan itibaren) istikrarlı bir tutum sergilediği ve diğer iki ürüne göre hep düşük değerlerde seyrettiği görülmektedir. Bu durum sürüm sayısına bağlı olarak, bu ürün ile ilgili teknik tartışmaların diğer iki ürüne göre daha az olduğu şeklinde yorumlanabilir. Ant ürününün sürüm bazında elektronik posta yoğunluğu, diğer iki ürüne göre hep yüksek değerlerde seyretmiştir. Bu durum ürünün sürümleri boyunca teknik tartışmaların her zaman var olduğunun bir göstergesidir. Dolayısıyla bu bilgi ihtiyacı bakımından (GD-ePY) Çizelge 7.5’de gösterildiği gibi, en iyi ürünün Archiva ve en kötü ürünün Ant olduğu görülmektedir.

7.2.2. Geriye Dönük Hata Çözme Başarı İndeksi (GD-HÇBİ)

Her bir ürünün hata raporlarına ulaşmak için Bölüm 6.2.2’deki gibi BugDB veri tabanı kullanılmıştır. Çizelge 7.6’da 1 Haziran 2013 yılı itibariyle her bir ürünün kod satır sayıları, toplam hata sayıları, toplam çözülen hata sayıları, geriye dönük hata çözme başarı oranı ve geriye dönük hata başarı indeksleri verilmiştir. Öncelikle ürünlerin hata çözme başarı oranı hesaplanmıştır. Bu hesaplama yapılırken ürünlerin 1 Haziran 2013 tarihine kadar çözülen hata sayısı o yıla kadarki toplam hata sayısına bölünmüş ve hata çözme başarı oranı için yüzdelik değer elde edilmiştir. Daha sonra

GD-HÇBİ, hata çözme başarı oranınının 1 Haziran 2013 yılındaki kilo kod satır sayısına bölünmesiyle hesaplanmıştır. Ürünlerin kod satır sayısı ne kadar fazla ise o kadar çok hatayla karşılaşması ve hata çözme başarı indeksinin (HÇBİ) düşük olması beklenir. Çizelgeye baktığımızda en uzun kod satır sayısına sahip olan Maven'in en düşük HÇBİ'ine (0,1415) ve en az kod satır sayısına sahip Ant ürününün en yüksek HÇBİ'ine (0,232) sahip olduğu görülmektedir.

Çizelge 7.6 Geriye dönük hata çözme başarı indeksi (GD-HÇBİ)

Ürün	Kod satır sayısı	Toplam hata sayısı	Çözülen hata sayısı	Hata çözme başarı oranı	GD-HÇBİ
Archiva	432.791	1728	1564	%90,5	0,209
Maven	690.804	4295	4199	%97,7	0,141
Ant	374.548	5748	5003	%87.0	0,232

7.2.3. Geriye Dönük Hata Ciddiyet İndeksi (GD-HCİ)

Bu bilgi ihtiyacının ölçülmesi için problem raporu veri tabanında (BugDB) 1 Haziran 2013 tarihine kadar kaydedilen her bir hatanın önem derecesi incelenmiştir. Ürünlerin her bir seviyedeki hata sayıları ayrı ayrı bulunmuş ve bu hata sayıları ürünlerin 1 Haziran 2013 itibariyle mevcut kod satır sayısına bölünerek Çizelge 7.7'de gösterildiği gibi değerler elde edilmiştir. Her bir seviyedeki hatanın ürün üzerindeki etkisi farklı olacaktır. Bu yüzden hatalara önem derecesine göre ağırlıklar verilmiştir (engelleyci = 9, kritik = 7, büyük = 5, küçük = 3, önemsiz = 1).

Her bir ürünün GD-HCİ değerinin hesaplanabilmesi için, Çizelge 7.7'de gösterilen hata önem derecesinin değeri ile önem derecesinin ağırlığı çarpılmıştır ve çarpımlar aşağıda gösterildiği gibi toplanmıştır:

$$Archiva = (0,0001 * 9) + (0,0002 * 7) + (0,0030 * 5) + (0,0004 * 3) + (0,00008 * 1) = 0,018$$

$$Maven = (0,0002 * 9) + (0,0003 * 7) + (0,004 * 5) + (0,001 * 3) + (0,0002 * 1) = 0,0271$$

$$Ant = (0,0007 * 9) + (0,0049 * 7) + (0,009 * 5) + (0,0004 * 3) + (0,0001 * 1) = 0,0866$$

Çizelge 7.7 Ürünlerin her bir hata seviyesindeki indeksleri ve GD-HCİ (2013 yılına kadar olan verilere göre)

Ürün	Engelleyici	Kritik	Büyük	Küçük	Önemsiz	GD-HCİ	Düzenlenmiş indeks
Archiva	0,0001	0,0002	0,0030	0,0004	0,00008	0,0185	0,540
Maven	0,0002	0,0003	0,004	0,001	0,0002	0,0271	0,369
Ant	0,0007	0,0049	0,009	0,0004	0,0001	0,0866	0,115

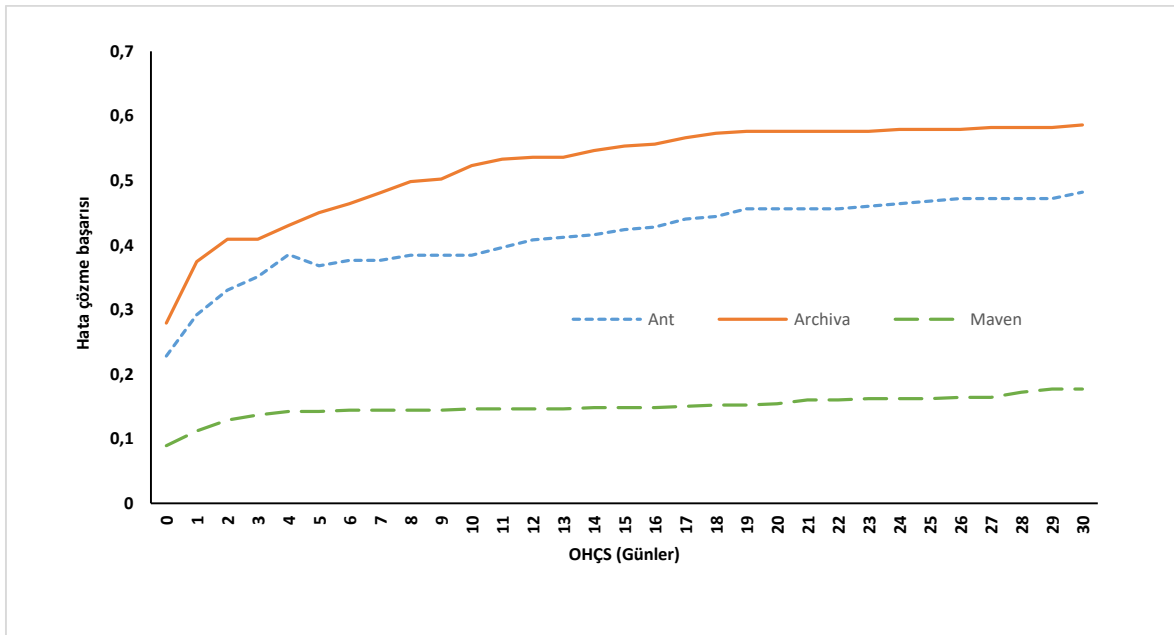
GD-HCİ değerlerine baktığımızda Ant AKY ürününün piyasaya sürüldüğünden itibaren diğer ürünlere göre daha ciddi hatalarla karşılaştığı görülmektedir. Archiva AKY ürününün ise diğer ürünlere nazaran daha önemsiz hatalarla karşılaştığı gözlemlenmiştir. GD-HCİ için elde edilen değer ne kadar küçük ise bu ürün için daha iyi bir durumdur. Ancak Bölüm 5.3.2’de bahsedildiği gibi hesaplamalar yapılırken büyük indeksteki ürünlerin daha tercih edilebilir olması amaçlandığından GD-HCİ için denklem 5 ve 6’nın uygulanması gerekmektedir. Denklemlerin uygulanması sonucunda değerlendirmede kullanılmak üzere elde edilen düzenlenmiş indeks değerleri çizelgedeki gibi sırayla Archiva, Maven ve Ant için 0,54, 0,36 ve 0,11 olmuştur.

7.2.4. Ortalama Hata Çözme Süresine Göre Geriye Dönük Hata Çözme Başarı Oranı (GD-OHÇS)

Bu bilgi ihtiyacının ölçülmesi için 1 Haziran 2013 tarihinden önceki iki yıla ilişkin hatalar göz önünde bulundurulmuştur. Bu hatalar bize 2013 yılı itibariyle üründe çalışan kod geliştiricilerin hata çözme süresine başarısını yansıtacaktır. Bu bilgi ihtiyacının ölçülmesi için 2013 yılından önceki iki yıllık dönemde açılan hataların açılış tarihi, kapanış tarihi ve o yıl itibarıyla ürünlere katkı sağlayanların sayısı (İng. number of contributors) metrikleri kullanılmıştır.

Çizelge 7.8 2013 yılından önceki iki yıllık dönemde açılan hataların çözülmesi ile ilgili bilgiler

Ürün	Archiva	Maven	Ant
Son iki yıldaki OHÇS	118 gün	251 gün	192 gün
Katkı sağlayanların sayısı	44	61	55
Son iki yılda açılan hataların açıldığı günden itibaren 30 gün içinde çözülme oranı	0,600	0,200	0,500



Şekil 7.2 2013 yılından önceki iki yıllık dönemde açılan hataların açıldığı günden itibaren 30 gün içerisinde çözülme oranları

Çizelge 7.8’de son iki yılda açılan hataların her bir ürün için ortalama kaç günde çözüldüğü ve her bir ürünlerdeki katkı sağlayıcıların sayısı verilmiştir. Şekil 7.2’de her bir ürün için son iki yılda açılan toplam hata sayılarının açıldıktan itibaren 30 gün içerisinde çözülme oranları gösterilmektedir (örneğin, bir hata 10.Kasım.2012 de açılmış ve 11.Kasım.2012 kapatılmış ise 1 gün içinde çözülmüştür). Son iki yılda açılan hataların çözülme oranlarına baktığımızda, Ant ürününün açılan hataların %25’ini (0,25), Archiva’nın %30’unu (0,3) ve Maven’in %10’unu (0,1) açıldığı gün çözdüğü görülmüştür.

Genel olarak baktığımızda, hem Çizelge 7.8’de hem de Şekil 7.2’de gösterildiği gibi, Ant ürünü için son iki yıl içinde açılan toplam hataların %50’sinin (0,5), Archiva ürünü için %60’ının (0,6) ve Maven ürünü için %20’sinin (0,2) açıldıktan itibaren 30 gün içerisinde çözüldüğü gözlemlenmiştir. Dolayısıyla OHÇS bakımından indeks değerleri Archiva, Maven ve Ant için sırayla 0,6, 0,2 ve 0,5 olmuştur.

Sonuçlar incelendiğinde GD-OHÇS bakımından Archiva AKY ürünü en başarılı ürün olmuştur. Şekil 7.2 incelendiğinde Ant ürününün kod geliştiricisi sayısının daha az olmasına rağmen genel anlamda Maven ürününden hataları çözme oranı bakımından daha başarılı olduğu görülmüştür.

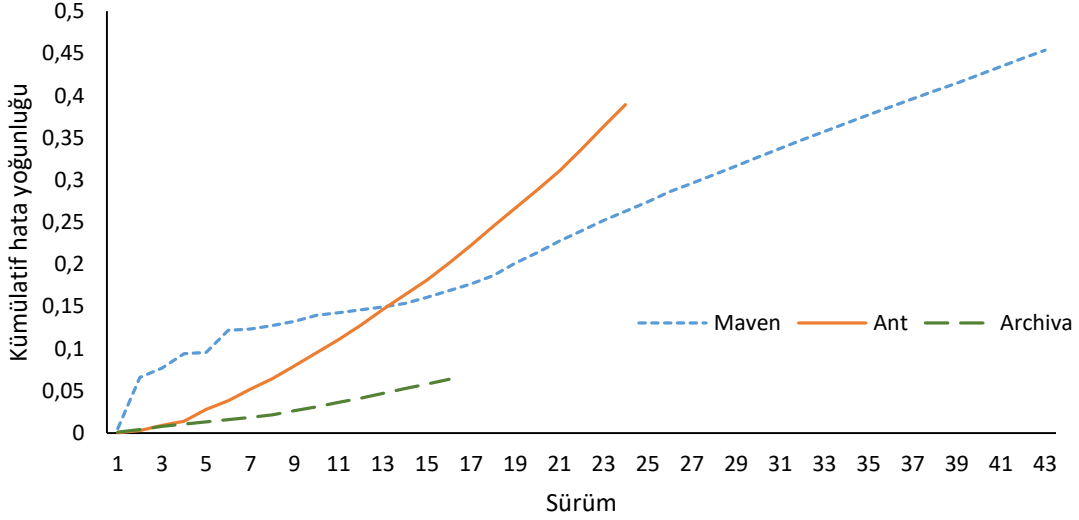
7.2.5. Ürün Sürümlerine Göre Geriye Dönük Kümülatif Hata Yoğunluğu (GD-SKHY)

Bu bölümde GD-SKHY bilgi ihtiyacını ölçmek için, ürünlerin 1 Haziran 2013 tarihine kadarki toplam hata sayıları ve bu tarihteki son sürümlerinin toplam kod satır sayısı kullanılmıştır. Ürünlerin hata yoğunluğu (HY) değeri, 1 Haziran 2013 tarihi itibarıyla toplam hata sayılarının yine o yıldaki son sürümlerinde bulunan toplam kod satır sayısına bölünmesiyle elde edilmiş ve sonuçlar Çizelge 7.9’da verilmiştir. Çizelgedeki değerlere baktığımızda hata yoğunluğu bakımından en tercih edilebilir ürünün Archiva ve en hata eğilimli ürünün ise Ant olduğu görülmüştür. Ancak Bölüm 5.3.2’de bahsedildiği gibi tüm bilgi ihtiyaçları için elde edilen indekslerin belli bir standartta olması gerekir. SKHY için elde edilen değer ne kadar küçük ise bu, ürün için daha iyi bir durumdur. Hesaplamalar yapılırken büyük indeksteki ürünlerin daha tercih edilebilir olması amaçlandığından SKHY için denklem 5 ve 6’nın uygulanması gerekmektedir. Denklemlerin uygulanması sonucunda değerlendirmede kullanılmak üzere elde edilen düzenlenmiş indeks değerleri tablodaki gibi sırayla Archiva, Maven ve Ant için 0,256, 0,161 ve 0,065 olmuştur.

Çizelge 7.9 2013 yılına kadar olan veriler temel alınarak hesaplanan hata yoğunluğu

Ürün	Archiva	Maven	Ant
Hata yoğunluğu (HY)	0,0039	0,0062	0,0153
Düzenlenmiş indeks	0,256	0,161	0,065

Bu durumu desteklemek için her bir ürünün hata yoğunluğu Şekil 7.3'deki gibi sürüm bazında incelenmiştir. Şekilde her bir ürün için üretildiği yıldan itibaren (1 Haziran 2013 yılına kadar), her bir sürümde o ana dek bulunan hata sayısı, o sürümün toplam kod satır sayısına bölünmüştür. Her bir sürüm için önceki sonuçlar, kümülatif olarak toplanmıştır.



Şekil 7.3 2013 yılına kadar olan veriler temel alınarak hesaplanan, sürümlere göre kümülatif hata yoğunluğu

Şekil incelendiğinde sürümler bazında Ant yazılım ürününün hata yoğunluğunun yüksek değerlerde seyrettiği gözlemlenmiştir. Hem çizelgedeki HY değerleri hem de şekildeki sürümler bazında hata yoğunlukları incelendiğinde, SKHY bakımından en iyi ürün Archiva ve en kötü ürün Ant olmuştur.

7.2.6. Geriye Dönük Toplum-tabanlı Değerlendirme için Genel Sonuçlar

Geriye dönük toplum-tabanlı değerlendirmenin genel sonuçlarını elde etmek için belirlenen bilgi ihtiyaçları, seçilen metriklerle 1 Haziran 2013 tarihine kadarki veriler temel alınarak ölçülmüş ve her bir bilgi ihtiyacı bakımından ürünler için ağırlıklar elde edilmiştir. Geriye dönük değerlendirme yapan kullanıcı ile güncel verilerle değerlendirme yapan kullanıcı aynı kişi olduğu ve aynı ihtiyaçla değerlendirme yaptığı için, bilgi ihtiyaçlarına yine aynı ağırlıklar verilmiştir (GD-ePY, GD-HÇBİ, GD-HCİ, GD-OHÇS, GD-SKHY bilgi ihtiyaçlarının ağırlıkları sırayla 4, 2, 3, 1 ve 5'tir).

Çizelge 7.10 Geriye dönük toplum-tabanlı değerlendirme indeksi (GD-TDİ) ve hesaplanması

Bilgi ihtiyacı	Archiva ağırlık değeri $AD(X)$	Maven ağırlık değeri $AD(Y)$	Ant ağırlık değeri $AD(Z)$	Değerlendiricinin ağırlık değeri $AD(U)$
GD-ePY	0,819	0,108	0,095	4
GD-HÇBİ	0,209	0,141	0,232	2
GD-HCİ	0,540	0,369	0,115	3
GD-OHÇS	0,600	0,200	0,500	1
GD-SKHY	0,256	0,161	0,065	5
$AD(i) * AD(U)$				
GD-ePY	$0,819 * 4 = 3,276$	$0,108 * 4 = 0,432$	$0,095 * 4 = 0,380$	
GD-HÇBİ	$0,209 * 2 = 0,418$	$0,141 * 2 = 0,282$	$0,232 * 2 = 0,464$	
GD-HCİ	$0,540 * 3 = 1,620$	$0,369 * 3 = 1,107$	$0,115 * 3 = 0,345$	
GD-OHÇS	$0,600 * 1 = 0,600$	$0,200 * 1 = 0,200$	$0,500 * 1 = 0,500$	
GD-SKHY	$0,256 * 5 = 1,280$	$0,161 * 5 = 0,805$	$0,065 * 5 = 0,325$	
Toplam	7,194	2,826	2,014	
Toplam/ Mümkün olan en büyük ağırlık	7,194/25	2,826/25	2,014/25	
GD-TDİ	0,287	0,113	0,080	

Belirlenen ağırlıklar sonucunda Çizelge 7.10'da gösterildiği gibi her bir ürün için Geriye Dönük Toplum-tabanlı Değerlendirme İndeksi (GD-TDİ) hesaplanmıştır. GD-TDİ değerinin nasıl hesaplandığı Bölüm 6.2.6'da detaylı olarak anlatılmıştır. Dolayısıyla Çizelgede görüldüğü gibi Archiva, Maven ve Ant AKY ürünleri için GD-TDİ değerleri sırayla 0,287, 0,113 ve 0,080 olmuştur.

7.3. Geriye Dönük Genel Değerlendirme Sonuçları

Bakım yapılabilirlik ve güvenilirlik için kod-tabanlı ve toplum-tabanlı değerlendirmeler 1 Haziran 2013 tarihine kadarki veriler temel alınarak geriye dönük olarak ölçülmüş ve sonuçlar Çizelge 7.11’de verilmiştir. Geriye dönük değerlendirme yapan kullanıcı ile güncel verilerle değerlendirme yapan kullanıcının aynı kişi olduğu ve aynı ihtiyaçla değerlendirme yaptığı varsayılmıştır. Bu sebeple güncel değerlendirmede olduğu gibi kod-tabanlı değerlendirmeye %65 ve toplum tabanlı değerlendirmeye %35 ağırlık vermiştir.

Çizelge 7.11 Geriye dönük kod-tabanlı ve toplum-tabanlı değerlendirme sonuçları

Değerlendirme boyutu	İndeks sonuçları	Archiva	Maven	Ant	Değerlendiricinin ağırlık değeri
Kod-tabanlı	GD-BYİ	0,520	0,500	0,229	%65
	GD-Gİ	0,224	0,187	0,110	
Toplum-tabanlı	GD-TDİ	0,287	0,113	0,080	%35

Çizelge 7.11’deki indeksler incelendiğinde, Bölüm 5.4’te verilen geriye dönük genel değerlendirme sonuçlarının elde edilmesi formülüne göre, sonuçlar aşağıdaki gibidir:

Bakım yapılabilirlik için,

$$Apache Archiva = (0,520 * 0,65) + (0,287 * 0,35) = 0,438$$

$$Apache Maven = (0,500 * 0,65) + (0,113 * 0,35) = 0,364$$

$$Apache Ant = (0,229 * 0,65) + (0,080 * 0,35) = 0,176$$

Güvenilirlik için,

$$Apache Archiva = (0,224 * 0,65) + (0,287 * 0,35) = 0,246$$

$$Apache Maven = (0,187 * 0,65) + (0,113 * 0,35) = 0,160$$

$$Apache Ant = (0,110 * 0,65) + (0,080 * 0,35) = 0,099$$

Bu tez çalışmasında geliştirilen metodun geriye dönük değerlendirme sonuçlarına göre, hem bakım yapılabilirlik (0,438) için hem de güvenilirlik (0,246) için en büyük indekse sahip Apache Archiva AKY ürünü en tercih edilebilir ürün olmuştur.

7.4. Güncel ve Geriye Dönük Değerlendirme Sonuçlarının Karşılaştırılması

Önerilen metodun tutarlılığını ölçmek için geçmişe dönük olarak hem kod-tabanlı hem de toplum-tabanlı değerlendirme yapılmıştır. Bu bölümde şu sorunun cevabı aranmıştır; kullanıcı bu tez çalışmasında önerilen metodu kullanarak 2013 yılındaki verilerle, belirlediği aday ürünler arasından değerlendirerek seçseydi sonuçlar nasıl değişirdi? Bölüm 6'da güncel verilerle yapılan ölçümler sonucunda ihtiyacı karşılayan en iyi ürün seçilmiştir. Bölüm 7'deki ölçümlerde ise 1 Haziran 2013 tarihine kadar olan veriler temel alınarak ürün seçimi yapılmıştır.

Yukarıda bahsi geçen iki bölümdeki değerlendirmeler sonucunda elde edilen sonuçlar, aşağıda karşılaştırılmış ve analiz edilmiştir.

Çizelge 7.12 2013 ve 2017 yıllarındaki kod-tabanlı ve toplum-tabanlı değerlendirme sonuçları

Değerlendirme boyutu	İndeks sonuçları	Archiva	Maven	Ant	Değerlendiricinin ağırlık değeri
Kod-tabanlı	BYİ	0,583	0,458	0,208	%65
	GD-BYİ	0,520	0,500	0,229	
	Gİ	0,189	0,062	0,169	
	GD-Gİ	0,224	0,187	0,110	
Toplum-tabanlı	TDİ	0,248	0,131	0,080	%35
	GD-TDİ	0,287	0,113	0,080	

Çizelge 7.12'de her bir ürün için 2013 ve 2017 yılında yapılan kod-tabanlı ve toplum-tabanlı ölçüm sonuçları verilmiştir. Çizelgeye baktığımızda bakım yapılabilirlik indeksi (BYİ) bakımından 2013 yılındaki verilere göre Archiva ürünü artış göstermiş ve diğer iki üründe düşüş olmuştur. 2013 yılındaki verilere göre BYİ bakımından en tercih edilebilir ürün Archiva olmuş ve 2017 yılındaki verilere göre sonuç değişmemiştir. Güvenilirlik indeksi (Gİ) bakımından 2013 yılındaki verilere göre de Archiva ürünü en tercih edilebilir ürün olmuş ve 2017 yılına gelindiğinde sonuç değişmemiştir. Toplum-tabanlı değerlendirme indeksi bakımından sonuçlar, 2013

yıldaki verilere göre pek deęişmemiş ve her iki ölçüm sonuçlarına göre TDİ en fazla olan ürün Archiva olmuştur.

Çizelge 7.12'de verilen indeksler göz önüne alındığında ve Bölüm 5.4 de verilen genel değerlendirme sonuçlarının elde edilmesi formülüne göre, 2013 ve 2017 yıllarında yapılan ölçüm sonuçların karşılaştırması Çizelge 7.13'de verilmiştir.

Çizelge 7.13 2013 ve 2017 yıllarında yapılan genel değerlendirme sonuçlarının karşılaştırılması

Öznitelik	Yıllar	Archiva	Maven	Ant
Bakım yapılabirlik	2013	0,438	0,364	0,176
	2017	0,464	0,342	0,163
Güvenilirlik	2013	0,246	0,160	0,099
	2017	0,208	0,085	0,137

Çizelgede gösterildiği gibi hem 2013 yılında hem de 2017 yılında bu tez çalışmasında önerilen metot kullanılarak yapılan ölçümler sonucunda, bakım yapılabirlik ve güvenilirlik bakımından en tercih edilebilir AKY ürünü Archiva olmuştur.

8. ÖNERİLEN METODUN OSMM VE OPENBRR KULLANILARAK DOĞRULANMASI

8.1. Modellerin Tanımı

8.1.1. Açık Kaynak Olgunluk Modeli (İng. Open Source Maturity Model-OSMM)

Açık kaynak olgunluk modeli [8] AKY'ların değerlendirilmesinde en çok kullanılan modellerden bir tanesidir. Capgemini [73] kuruluşu tarafından ilk kez 2003 yılında önerilmiş ve günümüzdeki AKY değerlendirme modellerinin temellerini oluşturmuştur.

Capgemini danışmanlık hizmeti veren dünya çapında bir kuruluştur. Teknolojinin hızla geliştiği ve sınırlarının kaybolduğu düşünüldüğünde şirketlerin karmaşık ve beklenilmeyen değişikliklere karşı hızlı ve sezgisel yanıt vermesini sağlar ve böylece sürekli gelişmesine ve değişmesine yardımcı olur. Şirketleri, ihtiyaçlarını karşılayan doğru teknolojiyi kullanma konusunda yönlendirir ve rekabet dünyasında bir adım önde olmalarını sağlar.

AKY'ların kullanımı Bölüm 2.1.3'de bahsedilen sebeplerden dolayı sürekli artmıştır. Kullanımın artması AKY'ların sayısının artmasına sebep olmuş ve birbirine alternatif olarak piyasaya sürülen yazılımların sayısı da artmıştır. Bu duruma, Capgemini şirketi sessiz kalmamış ve şirketlere AKY'ların seçiminde yardımcı olacak Açık Kaynak Olgunluk Modeli (OSMM) geliştirerek danışmanlık hizmeti vermiştir.

Bu modelin (OSMM) temel amacı AKY'ların herhangi bir organizasyon için uygun olup olmadığını belirlemektir. Bu model kullanıcı ile danışman arasındaki işbirliğine dayalı bir modeldir. Model kullanıcılardan gelen geri beslemelerle (İng. Feedback) sürekli güncellenmiştir. Modelde AKY'ların değerlendirilmesinde Bölüm 3'de bahsedildiği gibi herhangi bir kalite modeline bağlı kalınmamış ve değerlendirmede kullanılmak üzere müşteri görüşlerini ön planda tutan iki tane ana kriter belirlenmiştir: Ürün indikatörü (İng. Product indicator) ve uygulama indikatörü (İng. Application indicator). Model ürün indikatöründe 12 tane ve uygulama indikatöründe 15 tane olmak üzere, toplam 27 tane alt indikatörden oluşur.

8.1.1.1 Ürün İndikatörü

Ürün indikatörü, ürünün kalitesi ve olgunluğunu öngörüsöl (kullanım öncesi) olarak ölçen indikatördür. Bu tez çalışmasında önerilen metot, ürüne yönelik belirlenen kalite öznitelikleri bakımından öngörüsöl sonuçlar verdiđinden, dolaylı bu tez çalışmanın doğrulanması aşamasında ürün indikatörü kullanılmıştır.

Ürün indikatörü Çizelge 8.1’de görüldüğü gibi hiyerarşik yapıdan oluşmaktadır. Ürün indikatörünün altında 4 alt grup vardır ve bu 4 alt grubun altında AKY ölçümlerinde kullanılacak toplam 12 kriter vardır. Kullanıcı her bir AKY ürünü için bu 12 kritere ayrı ayrı 1 ile 5 arasında skorlar verir ve bu skorları her bir ürün için toplayarak en olgun ürünü seçer (toplamı büyük olan en olgundur). Bu kriterlere hangi şartlara göre skorlar verildiđi Çizelge 8.1’de gösterilmiştir. Ürün indikatörü altındaki 4 alt grup ve bu 4 alt grubun altında bulunan toplam 12 kriterin açıklamaları şu şekildedir:

1- Ürün (İng. product): Ürünün geliştirilmesi aşamasındaki özelliklere odaklanır (kod geliştiricilerin kararlılığı, başarısı vb.). Ürün grubunun ölçümünde 5 tane alt kriter kullanılır:

- **Ürün yaşı (İng. age):** AKY ürünlerinin aktif geliştirilme süresi ne kadar fazla ise üründe çalışan kod geliştiricilerin üründe çalışmayı aniden durdurma olasılığı da o kadar düşüktür ve ürün daha olgundur. AKY’larda ürünlerin piyasaya sürüldüğü ilk yıl çok önemlidir. Çünkü ilk yıl beklentileri karşılayamayıp kod geliştiricilerin ilgisini çekmez ve yeterli sponsor bulamazsa, daha sonra beklentileri karşılaması çok zor olacaktır.
- **Ürün lisansları (İng. licensing):** AKY’lar Bölüm 2.1.4’de bahsedildiđi gibi birçok lisansa sahiptir ve bu lisanslar AKY’larla neler yapabileceğimiz veya yapamayacağımız ile ilgili bilgiler verir. Bazı lisanslar çok kısıtlayıcı olduđu için kod geliştiricilerin ilgisini çekmeyebilir. Bu durum ürünlerin yeterli olgunluk seviyesine ulaşamamasına sebep olur.
- **İnsan hiyerarşisi (İng. human hierarchies):** AKY projelerinin sadece bir kurucu lider tarafından kontrol edilmesi, ürünlerin yazılım dünyasında yaşamlarının kısa olmasına sebep olmaktadır. Dolayısıyla kurucu liderler proje kontrolü konusunda bazı yetkilerini diđer aktif üyelere devretmeli ve üyeler arasında bir iş bölümü olmalıdır.

- **Satış noktası (İng. selling points):** AKY ürünleri ihtiyaçlara göre değiştirilip satışı yapılabilir. Çünkü bazı lisanslar buna izin vermektedir. Bazıları satışı bireysel seviyede bazıları da şirket seviyesinde yapmaktadır. Bu kriter tez çalışmasında kullanılan AKY ürünlerinin özelliklerini yansıtmamaktadır. Çünkü değerlendirme kullanıcı tarafından ürünleri satmak için değil, ihtiyaçları doğrultusunda kullanmak için yapılmıştır.
 - **Kod geliştirici topluluğu (İng. developer community):** Genellikle AKY projelerinde ne kadar çok kod geliştirici varsa ürün daha olgun bir ürün anlamına gelmektedir. Çünkü bir AKY projesinde bir kod geliştiricinin ilgisini başka projeler çekebilir ve geliştirici projeden ayrılabilir. Bu durumda projede çalışan 2 kod geliştirici olduğu varsayıldığında, projede çalışan kişilerin yarısı projeden ayrılmış olur. Dolayısıyla sayı ne kadar artarsa üründen çalışan kişilerin ayrılması daha az tehlike arz edecektir.
- 2- Entegrasyon (İng. integration):** Ürünlerin diğer ürünlerle birlikte çalışabilme derecesini ve bunun yanında ürünlerin modülerlik derecesini ölçer. Entegrasyon grubu ölçümünde 3 tane alt kriter kullanılır:
- **Modülerlik (İng. modularity):** Büyük çaplı AKY projelerinde ürünler, ana yapısı korunarak her biri farklı işlevler yapabilen çeşitli modüllere ayrılabilir. Bu durum ürünün büyüklüğünün ve çok yönlü olduğunun bir göstergesidir. Bu tarz AKY ürünleri kullanıcılar tarafından daha fazla ilgi görmektedir.
 - **Diğer ürünlerle işbirliği (İng. collaboration with other products):** AKY ürününün yeterli kitleye ulaştıktan sonra, diğer ürünlerle uyum içinde çalışması önem kazanmaktadır. Dolayısıyla ürünler diğer ihtiyaç duyulan ürünlerle ne kadar uyum içinde çalışıyorsa o kadar kaliteli ve olgundur.
 - **Standartlar (İng. standarts):** Ticari yazılımlarda olduğu gibi AKY ürünlerinde de yaygın standartların kullanılması ürünü bir adım öne çıkarmaktadır. Ürünün sadece bir kaç belirli ortamda desteklenen standartları kullanması, ürünün geçerliliği ve olgunluğu için dezavantajdır. Örneğin, bazı AKY ürünleri veri tabanına direkt bağlanırken, bazıları Windows ve Linux işletim sisteminin desteklediği ODBC kullanarak bağlanır ve bazıları da sadece Windows işletim sisteminin desteklediği OLEDB kullanarak bağlanmaktadır. Dolayısıyla bu şekil kısıtlamaların olması, ürünün kalitesi için bir dezavantajdır.

3- Kullanım (İng. use): Ürünün günlük kullanımı durumunda kullanıcının desteklenme derecesini ölçer. Örneğin, kullanıcıya sunulan destek seçeneklerinin sayısının gözden geçirilmesini mümkün kılar. Kullanım grubu ölçümünde 2 tane alt kriter kullanır:

- **Destek (İng. support):** Bazı AKY ürünlerinde kullanıcıların destek almak için gönderdikleri e-postalar sadece bir tane kod geliştircide toplanır. Bu da e-postalara geç cevap verilmesine veya hiç cevap verilmemesine sebep olur. Bazı AKY ürünlerinde ise gelen e-postalara cevap vermede bir iş bölümü vardır. Bu durumda gelen e-postalara cevap verilmemesi veya geç cevap verilmesi ihtimali daha düşüktür. Gelen e-postalara cevap verilme olasılığının yüksek olması kullanıcıların ciddiye alındığının, ürünlerin kaliteli ve olgun olduğunun göstergesidir.
- **Dağıtım kolaylığı (İng. ease of deployment):** AKY ürünlerinin nasıl kullanılacağına ilişkin bilgilere kolay ulaşılabilmesi ürünün popüler, kaliteli ve olgun olduğunun göstergesidir. Özellikle bağımsız kişiler veya kurumlar tarafında ürünle ilgili kullanım kılavuzu hazırlanmış ise bu ürünün kalitesi için çok önemli bir etkidir. Bu sayede yeni kullanıcılar fazla çaba sarf etmeden ihtiyaçlarını karşılayacaklardır.

4- Kabul (İng. acceptance): Ürünlerin kullanıcılar tarafından kabul görme ve popülerlik derecesini ölçer. Bu grup ürünün büyümesi ve önemli bir ürün haline gelmesi ile ilgilidir. Kabul grubu ölçümünde 2 tane alt kriter kullanır:

- **Kullanıcı topluluğu (İng. user community):** Bazı ürünlerin tartışma platformları çok sessizdir. Bu durum ürünün yeterli olgunluk seviyesine ulaşmadığının göstergesidir. Bazı ürünlerin tartışma platformları ise kullanıcı istekleri veya önerileri ile doludur. Bu durum ürünlerin popülerliğinin ve kullanıcı sayısının çok olduğunun göstergesidir. Ürün ile ilgili istekler ve öneriler çoğalınca, üründeki kod geliştirciler daha çok çalışacak veya çalışan sayısını artıracaktır. Bu da ürünlerin gelişimine katkı sağlayacaktır.
- **Pazara girme (İng. market penetration):** Kendini ispatlamış sistemler her zaman ürün kalitesi ve olgunluğu için önemlidir. Kullanıcılar rahatça istekte bulunup problemleri tartışabileceği sitelerdeki (Örn. Apache) piyasaya sürülen ürünlere daha çok talepte bulunmaktadır.

Çizelge 8.1. OSMM ürün indikatörleri ve her indikatör için skor verme kriterleri [8]

Ürün indikatörü	Skor: 1	Skor: 3	Skor: 5
Ürün grubu			
Ürün yaşı	< 2 ay	1-2 Yıl	> 3 yıl
Ürün lisansı	Ürün lisansı bilinmiyor olması veya net olmaması	Biliniyor olması fakat yapılabilecekler sınırlı olması (GPL, LGPL)	Kullanıcının ihtiyaçlarını karşılayan birçok imkân sunması
İnsan hiyerarşisi	Kurucu tek üyenin her şeyden sorumlu olması	Belirli grubun her şeyden sorumlu olması	Organizasyon özelliği göstermesi ve iş dağılımının detaylı belirtilmiş olması
Satış noktası	Bireysel	Kulüp	Organizasyon
Kod geliştirici topluluğu	Üründe kod geliştirici olarak ürüne dâhil olunamaması	Üründe kod geliştirici olunma imkânı olması ama üye olmak için belli prosedürün olmaması	Üründe kod geliştirici olma imkânı olması, ayrıca nasıl ve hangi rollerde olunacağını belli olması
Entegrasyon grubu			
Modülerlik	Modüllere ayırlamaması	Ana yapısını koruyarak modüllere limitli bir şekilde ayrılması	Yaşamını modüler olarak devam ettirebilmesi
Diğer ürünlerle işbirliği	Ürünün sadece kendi ana işlemlerini yerine getirmesi	Ürünlerin sadece belirli protokollerle başka ürünlerle etkileşim halinde olması	Ürünün ihtiyaç duyduğu bitin diğer ürünlere etkileşim halinde olması
Standartlar	Geçersiz standartları kullanması	Popüler olmayan standartları kullanması	En güncel endüstri standartlarını kullanması
Kullanım grubu			
Destek	Sadece kod geliştiriciler tarafından desteklenmesi	Hem kod geliştiriciler hem de kullanıcılar tarafından desteklenmesi	Kod geliştirici ve kullanıcıların yanında bağımsız bir şirket tarafından profesyonel olarak desteklenmesi
Dağıtım kolaylığı	Ürünle ve kullanımla ilgili dokümanların olmaması	Dokümanların sadece ürünlere odaklanması, kullanım ile ilgili dokümanların ve kursların bulunmaması	Ürünle ilgili dokümanların yanı sıra kullanım kılavuzunun olması veya çeşitli bağımsız kuruluşlar tarafından ürünlerin kullanımı ile ilgili eğitimlerin olması
Kabul grubu			
Kullanıcı topluluğu	Kullanıcı grubunun ürüne katkı sağlama konusunda ilgisiz olması (pasif kullanıcı grubu)	Kullanıcı grubunun ürüne tek yönüyle katkı sağlama konusunda daha aktif olması	Ürünün kullanıcı grubuna göre desteklenecek bölümlere ayrılması ve her bölümünün destekleyici kullanıcı grubunun olması
Pazara girme	Ürünün bilinmeyen bir ürün olması	Kendini ispatlamış ürünlere bir alternatif olması	Kendi alanında market lideri olması

AKY'ların değerlendirilmesinde OSMM kullanılırken Çizelge 8.1'de gösterilen bazı kriterlere verilecek skorların sınırları net olmayabilir. Bu yüzden değerlendirmenin doğru ve güvenilir olabilmesi için değerlendirmeyi yapacak kişinin AKY altyapısının

olması ve AKY'lara hâkim olması gerekmektedir. Ayrıca OSMM modelinin uygulanışının daha objektif olabilmesi için bu konuda birden fazla uzman kişi tarafından bağımsız değerlendirme yapılması da sonuçların doğruluğunu arttıracaktır.

Ayrıca Çizelge 8.1'de gösterilen bazı kriterler değerlendirme yapılacak AKY'nin özelliğini yansıtmayabilir. Bu durumda bu kriterin skor değerinin karşılaştırma yapılacak bütün ürünler için 3 olarak verilmesi gerekir (bu değer OSMM için eşik değeridir ve altındaki değerler yeterli olgunluğa ulaşmamış ürünleri temsil eder). Bunu yapmadaki amaç, bütün ürünlere aynı değeri vererek o kriterin ürünü pozitif veya negatif olarak etkilemesini önlemektir.

8.1.1.2 Uygulama İndikatörü

Uygulama indikatörleri AKY ürünleri için kullanım sonrası yapılan ölçüm sonuçlarıdır. Bu ölçüm işlemi ile ürünlerin daha önce öngörülen kullanıcı ihtiyaçlarını karşılayıp karşılamadığı ölçülür. Uygulama indikatörü ürünlerin kullanım aşaması sonrasında kullanıcı görüşleriyle ilgilendiği için, bizim geliştirdiğimiz metodun doğrulanmasında kullanılmamıştır. Çünkü bizim geliştirdiğimiz metot ürünler kullanılmadan değerlendirme yapmakta ve alternatifler arasından hangi ürünün kullanıcı ihtiyaçlarını daha çok karşıladığını belirlemektedir. Uygulama indikatörü ölçüm için skor verilmek üzere toplam 15 tane alt kriterden oluşur. Kullanıcı bu kriterlere her bir ürün için skorlar verir. Ayrıca kullanıcı bu kriterlere tez çalışmasında önerilen metotta yapıldığı gibi, önem derecesine göre skorlar verir. Daha sonra ürünün ağırlık değeri ile kullanıcının kriterlere verdiği ağırlıklar çarpılır ve sonuçlar her bir ürün için toplanır. OSMM'una göre toplamı daha büyük olan ürün uygulama indikatörü bakımından daha olgun bir üründür. Yukarıda bahsedilen 15 kriter şunlardır: Kullanılabilirlik, arabirim (İng. interfacing), performans, güvenilirlik, güvenlik, kanıtlanmış teknoloji (İng. Proven technology), satıcı bağımsızlığı (İng. Vendor independence), platform bağımsızlığı (İng. Platform independence), destek (İng. support), raporlama (İng. reporting), yönetim (İng. administration), öneri (İng. advice), eğitim (İng. training), çalışan kadrosu (İng. Staffing), uygulama (İng. implementation).

8.1.2. Açık Kaynak İş Hazırlık Oranı (İng. Open Business Readiness Rating- OpenBRR)

OpenBRR modeli SpikeSource [74], Center for Open Source Investigation at Carnegie Mellon West [75] ve Intel Corporation [76] kuruluşları tarafından geliştirilmiştir. OpenBRR modeli kullanıcının gereksinimlerini en iyi karşılayan AKY ürününün belirlenmesi amacıyla geliştirilmiştir. Modelde AKY'ların değerlendirilmesi için kullanılabilirlik (İng. usability), kalite (İng. quality), fonksiyonellik (İng. functionality), güvenlik (İng. security), performans (İng. performance), ölçeklenebilirlik (İng. scalability), mimari (İng. architecture), destek (İng. support), dokümantasyon (İng. documentation), benimseme (İng. adoption), topluluk (İng. community), profesyonellik (İng. professionalism) olmak üzere 12 tane kategori belirlenmiştir.

OpenBRR modeli AKY'ların değerlendirmesini 4 aşamada yapar:

- 1- Hızlı değerlendirme aşaması (İng. quick assessment):** Değerlendirilecek ürünler temel kullanım amacı dikkate alınarak hızlıca gözden geçirilir, ihtiyaç doğrultusunda olmayan ürünler elenir ve uygun adayların listesi oluşturulur.
- 2- Hedef kullanım değerlendirmesi (İng. target usage assessment):** Bu aşamada kullanıcı tarafından OpenBRR modelinde bulunan 12 kategoriye önem derecesine göre 1 den 12'ye doğru sıralama verilir (sayılar büyüdükçe önem derecesi artar). Bu 12 kategoriden önemli görülen en fazla 7 tane olmak üzere değerlendirme kategorisi seçilir.
- 3- Veri toplama ve işleme (İng. Data collection and processing):** Bu aşama en çok zaman alan aşamadır. Çizelge 8.2'de bulunan her bir kategorideki metrikler için veri toplanır. Bu veriler ışığında çizelgede belirlenen kurallara göre ürünlere 1 ile 5 arasında skorlar verilir.
- 4- Veri dönüşümü (İng. data translation):** Bu aşamada her bir kategorideki skorlar toplanır ve kullanıcının ihtiyacını karşılayan en iyi AKY ürünü belirlenir.

Bu tez çalışmasında çizelgede listelenen 12 kategoriye, tez çalışmasında önerilen metotla karşılaştırmaya en uygun olacak şekilde sıralama verilmiştir. OpenBRR modeline göre en çok 7 tane kategori seçilmesini gerektiğinden, ayrıca tez çalışmasında önerilen metotla karşılaştırılma ve ürünün kullanım öncesi değerlendirilmesine uygunluk açılarından en uygun 5 kategori seçilmiştir: kalite, topluluk, destek, dokümantasyon ve profesyonellik (önem sırasına göre verilmiştir).

Bu 5 kategorinin ölçülebilmesi için gerekli metrikler ve skor verme kriterleri Çizelge 8.2'de verilmiştir (diğer bütün kategorilerin skor verme kriterleri için bkz. [10]).

Çizelge 8.2. OpenBRR değerlendirme kategorileri, değerlendirme metrikleri ve skor verme kriterleri [10]

Kategori	Metrik	Tanım	Skor				
			5-mükemmel	4-iyi	3-kabul edilebilir	2-zayıf	1-kabul edilemez
Kalite	Son 12 aydaki küçük çaplı sürüm sayısı	Planlanan güncellemeleri denetler	2		1 veya 3		0 veya > 3
	Son 12 aydaki büyük çaplı sürüm sayısı	Zorunlu olarak yapılan güncellemelerdir (Programın kilitlenmesi, güvenlik sorunları vb.)	3-4		1-2 veya 5-6		0 veya > 6
	Son 6 ayda açılan hata sayısı	Ürün kullanımının kalitesini ölçer.	< 50	50-100	100-500	500-1000	> 1000
	Son 6 ayda çözülen hata sayısı	Açılan hataların ne kadar çabuk çözüldüğünü ölçer.	> %75	%60-75	%45-60	%25-45	<% 25
	Açılan kritik hata sayısı	Ürünün kalitesini etkileyen hataları ölçer.	0	1 - 5	5 - 10	10 - 20	> 20
	Son 6 aydaki açılan kritik hataların çözülme süresi	Ürünlerin kritik hatalar karşısında başarısını ölçer.	< 1 hafta	1 - 2 hafta	2 - 3 hafta	3 - 4 hafta	> 4 hafta
Destek	Son 6 aydaki e-posta sayıları	İnsanların destek amaçlı ilk başvurdukları yer e-posta listesi olduğundan ürünlerin destek seviyesini ölçer.	> 720 mesaj her ay	300-720 mesaj her ay	150-300 mesaj her ay	30-150 mesaj her ay	< 30 mesaj her ay
	Profesyonel desteğin kalitesi	Ürünlerin sorunlarının giderilmesinde profesyonel desteğin kalitesini ölçer.	Kurulum + sorun giderme + özelleştirme desteği		Sadece kurulum desteği		Her hangi bir destek yok.
Dokümantasyon	Çeşitli dokümantasyonların varlığı	Ürünlerde bulunan dokümanların kalitesini ölçer.	Kurulum, kullanıcı, yönetici, eğitici, geliştirici dokümanları var. (çeşitli formatlarda)	Kurulum, kullanıcı, yönetici rehberleri var (çeşitli formatlarda)	Kurulum ve kullanım kılavuzu var.	Sadece metin-tabanlı kurulum dokümanı var.	Uygun dokümanlar yok.
	Kullanıcıların katkısı	Kullanıcıların geribildirimlerle ürüne katkısını ölçer.	Kullanıcıların katkı yapmasına izin verilir ve yapılan katkılar uzmanlar tarafından düzenlenir.		Kullanıcıların katkı yapmasına izin verilir ve yapılan katkılar uzmanlar tarafından düzenlenmez.		Kullanıcılar katkı yapmaz.
Topluluk	Son 6 aydaki e-posta sayıları	İnsanların destek amaçlı ilk başvurdukları yer e-posta listesi olduğundan ürünlerin destek seviyesini ölçer.	> 720 mesaj her ay	300-720 mesaj her ay	150-300 mesaj her ay	30-150 mesaj her ay	< 30 mesaj her ay
	Son 6 aydaki kod geliştiricilerin sayısı	Kod geliştiricinin sayısı ne kadar fazla ise ürünün olgunluk seviyesi o kadar fazladır.	> 50	20-50	10-20	5-10	< 5
Profesyonellik	Proje yönetimi	Projenin iyi yönetilebilme ve proje için kaynak toplanabilme derecesini ölçer.	Bağımsız kuruluşlar tarafından destek verilir.		Kod geliştirici gruplar tarafından destek verilir.		Üründeki kod geliştiriciler tarafından bireysel destek verilir.
	Ana geliştirici takıma üye olma zorluğu	Olgun projeler üye seçiminde daha seçicidir. Ama yeni projeler bu konuda pek seçici olmazlar.	Başka projelerde kendini ispatladıktan (örn. her zaman aktif) sonra üye olunabilir.		Oldukça zordur öncesinde ürüne katkıda bulunmak gerekir.		Herkes üye olabilir.

8.2. Modellerin Uygulanması

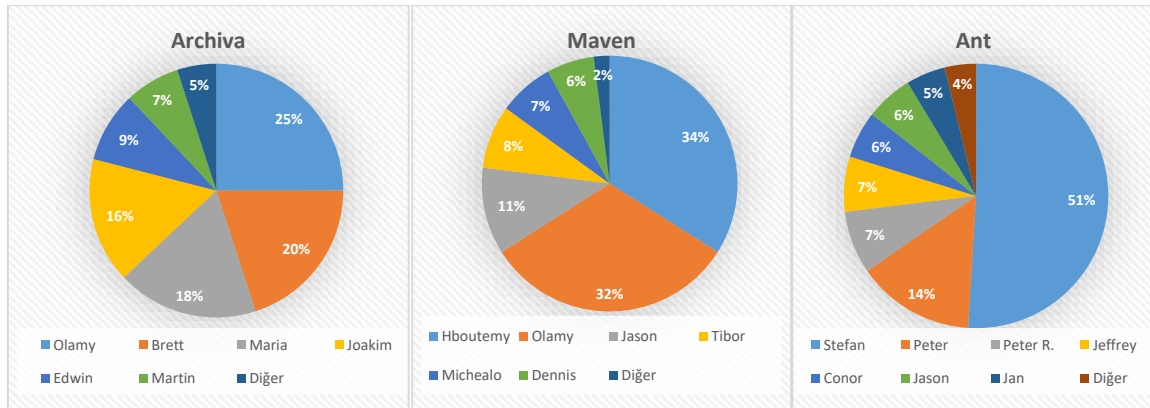
8.2.1. OSMM'nin Uygulanması

Bu tez çalışması için belirlenen üç aday AKY ürünü (Archiva, Maven ve Ant) OSMM'ine göre değerlendirilerek sonuçları verilmiştir. Bu değerlendirmeyi yaparken her bir kritere şu şekilde skor verilmiştir:

Ürün yaşı: Çizelge 8.1'deki kurallara göre, ürünler 3 yaşından büyük ise skor değeri 5 olması gerekmektedir. Dolayısıyla Çizelge 5.1'deki ürünlerin üretildiği yıllara baktığımızda hepsinin 3 yaşından büyük olduğu görülmektedir. Dolayısıyla bu kritere göre tüm ürünlerin skoru 5'tir.

Ürün lisansı: Çizelge 8.1'deki kurallara baktığımızda, ürünler Open Source Initiative (OSI) tarafından onaylanmış belirli bir lisansa sahipse skor değeri olarak 3 verilmesi gerekmektedir. Değerlendirme yapılan AKY ürünleri OSI tarafından onaylanmış Apache 2.0 lisansına sahiptir. Dolayısıyla hepsinin ağırlık skoru 3 olacaktır.

İnsan hiyerarşisi: Çizelge 8.1'deki kurallara göre, projenin bütün işlerini sadece kurucu lider (1 kişi) yapıyorsa skor değerinin 1 olması gerekir, birden fazla kişi tarafından yapılıyorsa (kulüp özelliği gösteriyorsa) skor değerinin 3 olması gerekmektedir. Birden fazla kişi tarafından yapılıyor ve herkesin iş dağılımı belli (organizasyon özelliği gösteriyorsa) ise skor değerinin 5 olması gerekmektedir. Bu kriteri ölçmek için her bir üründeki üyelerin ürüne sağladığı katkılar yüzdelik olarak incelenmiştir (Şekil 8.1).



Şekil 8.1 Her bir üründeki üyelerin ürüne sağladığı katkıların yüzdelik gösterimi

Kod geliştiricilerin ürünlere sağladığı katkılar incelendiğinde Archiva ürünü için kod geliştiricilerin iş dağılımının daha iyi yapıldığı ve organizasyon özelliği gösterdiği görülmektedir (skor: 5). Maven ürünü için birden fazla kişi katkı sağladıysa da iş dağılımı Archiva'nın gerisinde kalmıştır. Yapılan işlerin büyük bir bölümü 2 üyeye yüklenmiş ve bu yüzden kulüp özelliği göstermiştir (skor: 3). Ant ürününde birden fazla kişi katkı sağlamış olmasına rağmen iş yükünün büyük kısmı tek bir üyeye yüklenmiştir. Dolayısıyla tam olarak kulüp özelliği göstermemiştir (skor: 2).

Satış noktası: Belirlenen üç aday AKY ürününün herhangi bir kâr amacı olmadığından bu kriter ürünlerin özelliğini yansıtmamaktadır. OSMM'ine göre, eğer bir kriter değerlendirme yapılan ürünün özelliğini yansıtmıyorsa her bir ürüne skor değeri olarak 3 verilmesi gerekmektedir. Dolayısıyla bu kriter açısından tüm ürünlerin skoru 3'tür.

Kod geliştirici topluluğu: Değerlendirme yapılan üç aday AKY ürününde ürünlere kod geliştirici olarak katkı sağlayan herkes topluluğa üye olabilmektedir. Archiva ve Ant ürünlerinde üye olmak için prosedürle belirlenmiş herhangi bir dokümantasyon bulunmamaktadır (skor: 3). Maven ürününde nasıl üye olunacağı ve ne şartlarda olunacağı ürünün web sitesinde verilmiştir (skor: 5).

Modülerlik: Değerlendirme yapılan üç aday AKY ürünü de modüler yapıya ayrılmadan, tek bir ürün olarak yaşamını sürdürmektedir. Dolayısıyla Çizelge 8.1'deki kritere göre üç ürün için de skor değeri olarak 1 verilmiştir.

Diğer ürünlerle işbirliği: Değerlendirme yapılan üç aday AKY ürününün web sitelerinden elde ettiğimiz bilgilere göre, ürünler belirli protokoller gerektirmeden birlikte çalışması gereken ürünlerle (Netbeans, Eclipse, vb.) uyum içinde çalışabilmektedir. Dolayısıyla Çizelge 8.1'deki kriter göz önüne alındığında üç ürün için de skor değeri olarak 5 verilmiştir.

Standartlar: Değerlendirme yapılan üç aday AKY ürününün web sitelerinden elde ettiğimiz bilgilere göre ürünler son endüstri standartlarında geliştirilmiştir. Dolayısıyla Çizelge 8.1'deki kriter göz önüne alındığında üç ürün içinde skor değeri olarak 5 verilmesi uygun olacaktır.

Destek: Değerlendirme yapılan üç aday AKY ürünlerine e-posta yoluyla hem kod geliştiriciler hem de kullanıcılar tarafından destek verilmektedir. Dolayısıyla Çizelge 8.1'deki kurallara göre, ürünlere skor değeri 3 olarak verilmiştir.

Dağıtım kolaylığı: Değerlendirme yapılan üç aday AKY ürününden Maven ve Ant ürünlerinin sitelerinde ilgili dokümanlar mevcuttur fakat kullanım veya ürünle ilgili eğitici dokümanlar mevcut değildir. Archiva ürününün sitesinde ise bütün bu dokümanların yanı sıra kullanım kılavuzu ve çeşitli eğitim videoları mevcuttur. Çizelge 8.1'deki kurallar incelendiğinde Ant ve Maven için skor değerinin 3, Archiva için ise 5 olarak verilmesi uygun görülmüştür.

Kullanıcı topluluğu: Değerlendirme yapılan üç aday AKY ürünü için kullanıcıların aktifliğini ölçebilmek amacıyla kullanıcılardan gelen e-posta sayıları incelenmiştir. Ürünlerin CVS arşivleri incelendiğinde Archiva'nın 85.923 adet, Maven'in 80.926 adet ve Ant'in ise 73.019 adet kullanıcıdan gelen e-posta sayısına sahip olduğu görülmüştür. Bu sayılar her bir ürünlerdeki kullanıcıların çeşitli öneriler ve fikirlerle ürünlerde ne derece aktif olduğunu gösterir. Bazı ürünlerde kullanıcılar gruplara ayrılarak her grup ürünün belli bir bölümü için önerilerini sunar ve bu gruplar ürün üyeleri tarafından yönetilir. Ancak bu durum bu çalışmada değerlendirme yapılan ürünler için geçerli değildir. Dolayısıyla Çizelge 8.1'deki kurallara göre üç AKY ürünü içinde skor değeri olarak 3 verilmiştir.

Pazara girme: Eğer herhangi bir ürünün kurulu bir sistem alt yapısı var ise, olgunluğa erişmesi daha kolaydır. Değerlendirme yapılan AKY ürünlerinin üçü de Apache altyapısına sahip ve kendi alanında market lideri olmuş ürünlerdir. Dolayısıyla Çizelge 8.1'deki kurallara baktığımızda üç AKY ürünü için de skor değeri olarak 5 verilmesi uygun görülmüştür.

Yukarıda her bir ürün için her bir kriterin açıklaması verilmiş ve her bir ürün için skor değerlerinin nasıl verildiği açıklanmıştır. Çizelge 8.3'de her bir ürüne verilen skor değerleri listelenmiş ve skor değerleri toplanarak modele (OSMM) göre en iyi ürün belirlenmiştir.

Çizelge 8.3 OSMM'ine göre AKY'ların değerlendirme sonuçları (ürün indikatörleri)

Ürün indikatörü	Archiva	Maven	Ant
Ürün grubu			
Ürün yaşı	5	5	5
Ürün lisansı	3	3	3
İnsan hiyerarşisi	5	3	2
Satış noktası	3	3	3
Kod geliştirici topluluğu	3	5	3
Entegrasyon grubu			
Modülerlik	1	1	1
Diğer ürünlerle işbirliği	5	5	5
Standartlar	5	5	5
Kullanım grubu			
Destek	3	3	3
Dağıtım kolaylığı	5	3	3
Kabul grubu			
Kullanıcı topluluğu	3	3	3
Pazara girme	5	5	5
Toplam	46	44	41

8.2.2. OpenBRR Modelinin Uygulanması

Bu tez çalışması için belirlenen üç aday AKY ürünü (Archiva, Maven ve Ant) OpenBRR modeline göre değerlendirilerek sonuçları verilmiştir. Her bir ürün için Çizelge 8.2'de listelenen metrikler için veriler toplanmış ve değerleri hesaplanmıştır. Hesaplanan metrik değerleri ve Çizelge 8.2'deki kurallar temel alınarak ürünlere verilen skorlar Çizelge 8.4'de gösterilmiştir. Bu skor değerleri toplanarak bu model bakımından en iyi ürün seçilmiştir (toplamı en büyük olan en iyi üründür).

Çizelge 8.4'de OpenBRR modeline göre hesaplanan metrik değerleri ve değerlendirme sonuçları

Kategori	Metrik	Archiva		Maven		Ant	
		Hesaplanan metrik değeri	Ağırlık	Hesaplanan metrik değeri	Ağırlık	Hesaplanan metrik değeri	Ağırlık
Kalite	Son 12 aydaki küçük çaplı sürüm sayısı	2	5	1	3	3	3
	Son 12 aydaki büyük çaplı sürüm sayısı	1	3	0	1	1	3
	Son 6 ayda açılan hata sayısı	16	5	83	4	52	4
	Son 6 ayda çözülen hata sayısı	6	2	47	2	13	2
	Açılan kritik hata sayısı	12	2	12	2	21	1
	Son 6 aydaki açılan kritik hataların çözüme süresi	10 gün	4	17 gün	3	15 gün	3
Destek	Son 6 aydaki e-posta sayıları	496	4	560	4	132	2
	Profesyonel desteğin kalitesi	Sadece kurulum desteği	3	Sadece kurulum desteği	3	Sadece kurulum desteği	3
Dokümantasyon	Çeşitli dokümantasyonların varlığı	Kurulum ve kullanım kılavuzu var.	3	Sadece metin-tabanlı kurulum dokümanı var	2	Sadece metin-tabanlı kurulum dokümanı var	2
	Kullanıcıların katkısı	Kullanıcıların katkı yapmasına izin verilir ve yapılan katkılar uzmanlar tarafından düzenlenir.	5	Kullanıcıların katkı yapmasına izin verilir ve yapılan katkılar uzmanlar tarafından düzenlenir.	5	Kullanıcıların katkı yapmasına izin verilir ve yapılan katkılar uzmanlar tarafından düzenlenir.	5
Topluluk	Son 6 aydaki e-posta sayıları	496	4	560	4	132	2
	Son 6 aydaki kod geliştiricilerin sayısı	18	3	25	4	21	3
Profesyonellik	Proje yönetimi	Kod geliştirici gruplar tarafından destek verilir.	3	Kod geliştirici gruplar tarafından destek verilir.	3	Kod geliştirici gruplar tarafından destek verilir.	3
	Ana geliştirici takıma üye olma zorluğu	Oldukça zordur öncesinde ürüne katkıda bulunmak gerekir.	3	Oldukça zordur öncesinde ürüne katkıda bulunmak gerekir.	3	Oldukça zordur öncesinde ürüne katkıda bulunmak gerekir.	3
TOPLAM			49		43		39

8.3. Sonuçların Karşılaştırması

Bu bölümde, tez çalışmasında önerilen metodu kullanarak elde ettiğimiz sonuçlarla, AKY'ların değerlendirilmesi için literatürde en çok bilenen metotlardan OSMM ve OpenBRR kullanılarak elde edilen sonuçlar karşılaştırılmıştır.

Literatürdeki metotlar direkt ürün kalitesine ve olgunluğuna yönelik tek bir ürün seçimi yapmaya yöneliktir. Fakat bu tez çalışmasında önerilen metot bakım yapılabilirlik ve güvenilirlik bakımından en iyi ürünlerin belirlenmesinde her bir öznitelik için ayrı ayrı sonuçlar hesaplanmıştır. Bu sebeple bu tez çalışmasında

değerlendirilen bu iki öznelik için sonuçların karşılaştırması Çizelge 8.5'de ayrı ayrı verilmiştir.

Çizelge 8.5 Önerilen metot ile OSMM ve OpenBRR modellerinin değerlendirme sonuçlarının karşılaştırılması

	Metotlar	Archiva	Maven	Ant
Bakım yapılabilirlik	Önerilen metot	0,464	0,342	0,163
	Önerilen metodu geriye dönük uygulanması	0,438	0,364	0,176
	OSMM	46	42	41
	OpenBRR	49	43	39
Güvenilirlik	Önerilen metot	0,208	0,085	0,1378
	Önerilen metodu geriye dönük uygulanması	0,246	0,1608	0,0995
	OSMM	46	42	41
	OpenBRR	49	43	39

Çizelge incelendiğinde, bu tez çalışmasında önerilen metodun sonuçlarına göre Apache Archiva ürünü hem bakım yapılabilirlik ve hem de güvenilirlik bakımından en tercih edilebilir üründür. OSMM ve OpenBRR sonuçları incelendiğinde elde edilen değerlendirme sonuçlarının tez çalışmasında elde edilen sonuçları desteklediği görülmüştür.

9. SONUÇLAR VE TARTIŞMA

9.1. Genel Sonuçlar

AKY'ların popüleriği özellikle 2010 yılından sonra büyük bir artış göstermiştir. Bu yüzden birbirine alternatif olarak üretilen AKY ürünlerinin sayısı artmış ve kullanıcıların ihtiyaçlarını karşılayan en uygun ürünü seçmeleri zor bir süreç haline gelmiştir. Yanlış yazılım ürünü seçiminin ihtiyaç sahiplerine maddi olarak büyük zararlara yol açtığı düşünöldüğünde doğru ürünü seçme işi daha da önem kazanmıştır.

Bu tez çalışmasında doğru AKY seçiminin öneminden yola çıkılmış ve literatüde AKY seçimine yönelik metotlar analiz edilmiştir. Bu metotlar incelenerek eksik yönleri tespit edilmiş ve bu kapsamda bu metotların iyi yönleri temel alınarak AKY'ları hem kod-tabanlı hem de toplum-tabanlı olarak iki boyutta değerlendiren genel bir metot önerilmiştir.

Önerilen bu metot, aday AKY ürünleri olarak belirlenen üç tane Java tabanlı kod inşa aracının bakım yapılabirliğini ve güvenilirliğini ölçmek için uygulanmış ve uygulama sonuçları incelenmiştir. Bakım yapılabirlik ve güvenilirliğin kod-tabanlı değerlendirmesi yapılırken oluşturulan araştırma soruları ile bu iki özneliği ölçmeye yönelik metrikler araştırılmış ve literatürde en çok kullanılan C&K metrik seti, iç içe döngü sayısı, çevrimsel karmaşıklık ve açıklamaların sıklığı metrikleri kullanılmıştır. Toplum tabanlı değerlendirme yapılırken ISO/IEC 15939 ölçüm standardı temel alınmıştır. Bu standart dâhilinde, bakım yapılabirlik ve güvenilirliği ölçmeye yönelik bilgi ihtiyaçları belirlenmiştir. Bu bilgi ihtiyaçlarını ölçmek için metrikler araştırılmış ve bu metrikleri ölçmek için BugDB, CVS arşivi ve Bugzilla veri tabanları derinlemesine analiz edilip gerekli bilgiler elde edilmiştir.

Ölçümler yapılırken kod-tabanlı değerlendirmede metrik değerlerine göre ürünlere ağırlıklar verilmiştir (1 ile 3 arasında). Bu ağırlık değerleri kullanılarak hem bakım yapılabirlik hem de güvenilirlik için ayrı ayrı indeks değerleri elde edilmiştir. Toplum-tabanlı değerlendirme yapılırken belirlenen bilgi ihtiyaçlarını ölçmeye yönelik metrikler kullanılarak her bir bilgi ihtiyacı için indeks değerleri elde edilmiştir. İki boyut için her bir ürün bakımından ayrı ayrı indeks değerleri elde edildikten sonra

değerlendirici tarafından ölçüm boyutlarına ağırlık verilmiştir. Bütün bu indeks değerleri girdi olarak alınmış ve her bir ürün için bir indeks değeri elde edilmiştir. En büyük indekse sahip ürün en tercih edilebilir ürün olarak belirlenmiştir. Ürünlerin seçiminde kullanılmak üzere en son elde edilen indeks değerleri, ürünler için sabit indeks değerleri değildir. Bu indeks değerleri ürünlerin karşılaştırıldığı diğer ürünlere ve değerlendirme yapan kullanıcının ihtiyacına göre değişebilir. Bu sebeple elde edilen indeks değerleri, karşılaştırılan ürünler için tercih edilebilirlik sırası oluşturan, görece indeks değerleri olarak algılanmalıdır.

Java tabanlı kod inşa araçlarının bakım yapılabilirliğini ve güvenilirliğini ölçmek için önerilen metodun tutarlılığını değerlendirmek amacıyla, bu AKY'lar üzerinde geriye dönük değerlendirme de yapılmıştır. AKY'ların 1 Haziran 2013 tarihine kadarki verileri kullanılarak aynı ihtiyaçlara sahip kullanıcı tarafından seçim yapılmış ve sonuçlar güncel değerlendirme sonuçları ile karşılaştırılmıştır. Hem güncel verilerle yapılan ölçüm sonuçları hem de 2013 yılı temel alınarak yapılan ölçüm sonuçlarına göre kullanıcı ihtiyaçlarını karşılayan en iyi ürün, Apache Archiva AKY yazılım ürünü olarak gözlemlenmiştir.

Önerilen metodun ölçüm sonuçlarının doğrulanması için, literatürde AKY'ları değerlendirmeye yarayan ve en popüler metotlardan olan OSMM ve OpenBRR kullanılmıştır. Hem güncel olarak elde edilen hem de geriye dönük elde edilen sonuçlar, bu metotların sonuçları ile karşılaştırılmıştır. OSMM ve OpenBRR sonuçlarına göre de en iyi ürünün Apache Archiva olduğu gözlemlenmiştir.

9.2. Çalışmaya ve Sonuçlara Yönelik Kısıtlar

Tez çalışmasında önerilen metodun geliştirilmesi aşamasında literatürde önerilen metotların eksik yönlerinden yola çıkılmıştır. Literatürdeki metotlar ürünlerin olgunluğunu ve kalitesini ölçmeye yönelik kalite modelleri önermişlerdir. Bazı çalışmalar değerlendirme aşamasında öznelliği ön plana çıkarmış bazıları ise kullanıcı görüşlerini hesaba katmadan nesnel olarak değerlendirme yapmıştır. Tez çalışmasında önerilen metotta en güncel kalite modeli olan ISO/IEC 25010 kalite modeli kullanılmış ve değerlendirmenin bazı aşamalarında nesnel bazı aşamalarında ise öznel davranılarak iki boyutlu metot geliştirilmiş ve literatürdeki açık kapatılmaya çalışılmıştır. Kod-tabanlı tarafta ölçüm yapılırken literatürde en sık

referans edilen metrikler kullanılmıştır. Toplum-tabanlı-tarafta ise bilgi ihtiyaçlarının ve metriklerin belirlenmesinde ISO/IEC 15939 ölçme süreci modeli kullanılmıştır.

Bu tez çalışmasının zayıf yönü olarak, önerilen metotta kullanılan ağırlıklandırma yöntemi gösterilebilir. Kod-tabanlı ve toplum-tabanlı olarak ayrı ayrı en iyi ürün belirlendikten sonra, değerlendiriciden boyutlara ağırlık vermesi beklenmiştir. Dolayısıyla hem kod-tabanlı değerlendirmede hem de toplum-tabanlı değerlendirmede farklı ürünler en tercih edilebilir ürünler olarak gözlemlenseydi, değerlendiricinin boyutlara vereceği ağırlıklar daha da büyük önem kazanacaktı. Başka bir deyişle değerlendirmede öznellik büyük bir pay sahibi olacaktı. Dolayısıyla kullanıcının boyutlara yanlış ağırlıklar vermesi belki de yanlış ürünü seçmesine sebep olabilecekti.

Bu tez çalışmasında önerilen metot sadece çalışmada belirlenen üç Java tabanlı kod inşa aracı için uygulanmış ve elde edilen sonuçlar literatürde kullanılan popüler iki metotla doğrulanmıştır. Daha geniş kapsamda geçerliliğin, metodu farklı ürünlerde uygulamak ve sonuçları analiz etmek suretiyle sınanması gereklidir.

Bu tez çalışmasında yapılan ölçümlerin aynı ihtiyaca sahip farklı kişiler tarafından yapıldığında aynı sonuçları vermesi beklenmektedir. Çünkü ölçümler için araştırılan ve elde edilen metrikler tamamen nesnel metriklerdir ve ölçüm sonuçları kişiye göre değişen metrikler değildir. Fakat ölçüm yapan kişilerin ihtiyaçlarının farklı olması durumunda elde edilen sonuçların da farklılık göstermesi olası bir durumdur. Çünkü metodun bazı aşamalarında kullanıcı öncelikleri de dikkate alınmaktadır. Dolayısıyla önerilen metodu uygulayacak kişilerin ihtiyaçlarını tam olarak bilmesi ve AKY altyapıları hakkında bilgi sahibi olması gerekmektedir.

Bu tez çalışmasında kullanılan 4 aşamalı metodun en fazla zaman alan aşaması 3. aşama olan metriklerin belirlenmesi, toplanması ve ölçüm yapmak için adımların belirlenmesi aşamasıdır (Şekil 4.1). Dolayısıyla, metot herhangi bir ürünün bakım yapılabilirlik ve güvenilirliğini ölçmek için kullanılacaksa yaklaşık 5 iş günü kadar zaman alacaktır. Çünkü bu çalışmada hem kod-tabanlı boyutta hem de toplum-tabanlı boyutta kullanılacak metrikler, metriklerin elde edilme şekilleri, ölçme şekilleri ve ölçümde takip edilecek adımlar açık bir şekilde tanımlanmıştır. Bu çalışmada önerilen metodun bakım yapılabilirlik ve güvenilirlik için uygulanması yaklaşık 30 iş günü bir zaman almıştır. Çünkü hem kod-tabanlı hem de toplum-tabanlı boyutta

metriklerin sıfırdan belirlenmesi, elde edilmesi, ölçülmesi ve ölçüldükten sonra nasıl adımlar izleneceği zaman alan bir süreçtir. Önerilen metot bu iki öznelik dışında farklı bir özneliği ölçmek için uygulanacaksa araştırma, tanımlama ve uygulama eforunu göz önünde bulundurmak gerekmektedir.

Ayrıca bu çalışmada kod-tabanlı değerlendirme için literatürde en yaygın uygulanan [77] C&K metrik seti kullanılmıştır. Bu boyutta değerlendirmenin doğrulanması için nesneye yönelik başka metriklerle de ölçüm yapılması faydalı olacaktır.

9.3. Gelecek Çalışmalar

Bu tez çalışmasında bütün kalite özneliklerine uygulanabilecek genel bir değerlendirme metodu önerilmiştir. Gelecekte bu metodun, en bilinen kalite modelleri tarafından tanımlanmış olan (bakım yapılabilirlik ve güvenilirlik dışında) başka özneliklere de uygulanarak değerlendirme metotları önerilmesi planlanmaktadır.

AKY'ların bakım yapılabilirlik ve güvenilirliğini değerlendirmek üzere önerilen metodun kod-tabanlı boyutunda, C&K metriklerine ek olarak MOOD metrik setine göre ölçümler yapılması ve elde edilen sonuçların doğruluğunun sınanması hedeflenmektedir.

Önerilen modeli doğrulama aşamasında OSMM modelinin önerdiği uygulama indikatörleri, ürünlerin kullanıldıktan sonra değerlendirilmesine imkân sunduğundan dolayı bu çalışmada önerilen metodun karşılaştırılmasında kullanılmamıştır. Çünkü önerilen metotta ürünlerin kullanım öncesinde değerlendirmesi yapılmaktadır. Bununla birlikte toplum-tabanlı değerlendirmede kullanılan veri tabanları daha derinlemesine analiz edilip ürünlerin kullanım aşamasına ilişkin metrikler elde edilebilir. Gelecekte bu metriklerin elde edilerek önerilen modelin OSMM'nin uygulama indikatörlerine göre karşılaştırılmasının yapılması hedeflenmektedir.

AKY'ların bakım yapılabilirlik ve güvenilirliğini değerlendirmek üzere önerilen metodun toplum-tabanlı boyutunda ise ürünlerin veri tabanları kullanılarak yeni metrikler elde edilmesi hedeflenmektedir. Örneğin, ürünlerin e-posta arşivinde listelenen e-posta listelerinin, Metin Madenciliği (İng. Text Mining) kullanılarak içeriklerine inilmesi ve bu teknik konuşmalardan ürünlerin geleceği ile ilgili yeni metrikler elde edilmesi mümkün olabilir.

Bu tez alıřmasında nerilen metodun uygulanmasını kolaylařtırmak adına bir ara (İng. tool) geliřtirilmesi de hedeflenenler arasındadır. Bu ara sayesinde belirlenen aday AKY rnlerini deęerlendirmek iin manuel olarak yapılan tanımlama ve hesaplama iřlemlerinin otomatik olarak yapılması amalanmaktadır.

KAYNAKLAR

- [1] Open-source software, URL: https://en.wikipedia.org/wiki/Open-source_software. [son ziyaret tarihi: 22-Nisan-2017].
- [2] Black Duck Software. URL : <https://www.blackducksoftware.com/>. [son ziyaret tarihi: 16-Mayıs-2017].
- [3] North Bridge. URL: Available: <http://www.northbridge.com/>. [son ziyaret tarihi: 21-Mayıs-2017]
- [4] Marc Andreessen on Why Software Is Eating the World: URL: <http://online.wsj.com/article/SB10001424053111903480904576512250915629460.html>. [Son ziyaret tarihi: 11-Mayıs-2017].
- [5] K. Noyes, Senior U.S. Correspondent, PCWorld, Nisan 17, 2013, URL: <http://www.pcworld.com/article/2035651/open-source-is-taking-over-the-software-world-survey-says.html>. [son ziyaret tarihi: 13-Nisan-2017]
- [6] GitHub - Cracking the Code to GitHub's Growth. URL: <https://growthhackers.com/growth-studies/github>. [Son ziyaret tarihi: 11-Mayıs-2017].
- [7] P. Maki-Asiala ve M. Matinlassi, "Quality assurance of open source components: Integrator point of view," Bildiri Kitabı - *International Computer Software and Applications Conference*, 2006, vol. 2, sf. 189–192.
- [8] F.-W. Duijnhouwer ve C. W. Capgemini, "Open Source Maturity Model The Usefulness of a Maturity Model Open Source in a Corporate Setting."
- [9] I. Samoladas, G. Gousios, D. Spinellis, ve I. Stamelos, "The SQO-OSS quality model: measurement based open source software evaluation," *Open Source Dev. Communities Qual.*, vol. 275, sf. 1–11, 2008.
- [10] Business Readiness Rating for Open Source. URL: '<http://www.openbrr.org>. [son ziyaret tarihi: 15-Mayıs-2017]
- [11] Method for Qualification and Selection of Open Source Software (QSOS), version 1.6. URL: <http://www.qsos.org/>. [son ziyaret tarihi: 15-Mayıs-2017]
- [12] Ø. Hauge, T. Østerlie, C.-F. Sørensen, ve M. Gereaa, "An Empirical Study on Selection of Open Source Software - Preliminary Results," *Emerg. Trends Free. Source Softw. Res. Dev. 2009. FLOSS '09. ICSE Work.*, sf. 42–47, Mayıs 2009.
- [13] The Open Source Definition | Open Source Initiative. URL: <https://opensource.org/docs/osd>. [Son ziyaret tarihi: 11-Mayıs-2017].
- [14] M. Raşit Özdaş, "Kamuda Açık Kaynak Kodlu Yazılım Kullanımı," no. T.C kalkınma bakanlığı bilgi toplum dairesi, 2012.
- [15] İ. Eser, "Kamuda Özgür Yazılım Kullanımı: Dünya ve Türkiye'deki Örnekleri Üzerinden Bir Değerlendirme," *Radyo ve Telev. Üst Kurulu*, 2011.
- [16] EU. URL: <http://eu.conecta.it/index.htm>. [Son ziyaret tarihi: 11-Mayıs-2017].
- [17] Open source, open standards and re-use: government action plan - GOV.UK. URL: <https://www.gov.uk/government/publications/open-source-open-standards-and-re-use-government-action-plan>. [Son ziyaret tarihi: 11-Mayıs-2017].
- [18] Technology Research | Gartner Inc. URL: <http://www.gartner.com/technology/home.jsp>. [Son ziyaret tarihi: 11-Mayıs-2017].
- [19] Pardus. URL: <http://www.pardus.org.tr/>. [Son ziyaret tarihi: 11-Mayıs-2017].
- [20] TR Açık kaynak kod platform, URL: <https://acik-kaynak.org.tr>. [son ziyaret tarihi: 16-Mayıs-2017]
- [21] CyberSource. URL: <https://www.cybersource.com/>. [Son ziyaret tarihi: 11-Mayıs-2017].
- [22] Açık Kaynak Lisansları. URL: Available: <http://www.yusufaytas.com/acik-kaynak-lisanslari/>. [Son ziyaret tarihi: 11-Mayıs-2017].

- [23] Software Quality – Software Testing Fundamentals. URL: <http://softwaretestingfundamentals.com/software-quality/>. [Son ziyaret tarihi: 13-Mayıs-2017].
- [24] Software Quality Models and Philosophies. URL: [www.bth.se/com/besq.nsf/\(WebFiles\)/CF1C3230DB425EDCC125706900317C44/\\$FILE/chapter_1.pdf](http://www.bth.se/com/besq.nsf/(WebFiles)/CF1C3230DB425EDCC125706900317C44/$FILE/chapter_1.pdf). [Son ziyaret tarihi: 13-Mayıs-2017].
- [25] Quality Models in Software Engineering. URL: <https://msritse2012.wordpress.com/2013/01/27/quality-models-in-software-engineering/>. [Son ziyaret tarihi: 13-Mayıs-2017].
- [26] R. B. Grady, *Practical software metrics for project management and process improvement*. Prentice Hall, 1992.
- [27] R. G. Dromey, “A model for software product quality,” *IEEE Trans. Softw. Eng.*, vol. 21, no. 2, sf. 146–162, 1995.
- [28] ISO9126 - Software Quality Characteristics. URL: <http://www.sqa.net/iso9126.html>. [Son ziyaret tarihi: 13-Mayıs-2017].
- [29] K. Kurtel, “Yazılım Bakım Altarakteristiklerinin ISO Tabanlı Modeller Kullanılarak Ölçülmesi.”, 2009
- [30] ISO/IEC 25010, URL: ‘<http://iso25000.com/index.php/en/iso-25000-standards/iso-25010>.’ [Son ziyaret tarihi: 29-Mart-2017].
- [31] ISO/IEC 15939:2007 - Systems and software engineering -- Measurement process. URL: <https://www.iso.org/standard/44344.html>. [Son ziyaret tarihi: 29-Mart-2017].
- [32] P. Goodman, *The Practical Implementation of Software Metrics*. McGraw-Hill, 1993.
- [33] G. G. Schulmeyer, *Handbook of Software Quality Assurance*. Prentice Hall, 1987.
- [34] S. R. Chidamber ve C. F. Kemerer, “A Metrics Suite for Object Oriented Design,” *IEEE Trans. Softw. Eng.*, vol. 20, no. 6, sf. 476–493, 1994.
- [35] E. Ural, T. Umut, ve B. Feza, “Nesneye Dayalı Yazılım Metrikleri ve Yazılım Kalitesi,” *Yazılım Kalitesi ve Yazılım Geliştirme Araçları Sempozyumu*, 2008.
- [36] Depth of Inheritance Tree (DIT). URL: http://support.objecteering.com/objecteering6.1/help/us/metrics/metrics_in_detail/depth_of_inheritance_tree.htm. [Son ziyaret tarihi: 13-Mayıs-2017].
- [37] P. Antony, “Predicting Reliability of Software Using Thresholds of CK Metrics.,” *Int. J. Adv. Netw.*, vol. 1785, no. 6, sf. 1778–1785, 2013.
- [38] Johny Antony P & Harsh Dev, “ESTIMATING RELIABILITY OF SOFTWARE SYSTEM USING OBJECT-ORIENTED METRICS,” *Int. J. Comput. Sci. Eng. Inf. Technol. Res.*, vol. 3, no. 2, sf. 283–294, 2013.
- [39] S. K. Misra, E. Sikkim, ve B. Roy, “Assessment of Object Oriented Metrics for Software Reliability,” vol. 4, no. 1, sf. 432–435, 2015.
- [40] L. Dr. Rosenberg, T. Hammer, ve J. Shaw, “Software metrics and reliability,” *Proc. 9th Int. Symp. Softw. Reliab. Eng.*, sf. 1–8, 1998.
- [41] M. Hanefi Calp ve N. Arici, “Nesne Yönelimli Tasarım Metrikleri ve Kalite Özellikleriyle İlişkisi,” *Politek. Derg. J. Polytech. Cilt Digit. Object Identifier*, vol. 14141, no. 10, sf. 9–14, 2011.
- [42] Law of Demeter & Objective Sense of Style. URL: <https://www.slideshare.net/vladimirtsukur/law-of-demeter-objective-sense-of-style>. [Son ziyaret tarihi: 13-Mayıs-2017].
- [43] J. M. Bieman ve B.-K. Kang, “Cohesion and reuse in an object-oriented system,” in *Proceedings of the 1995 Symposium on Software reusability - SSR '95*, 1995, sf. 259–262.
- [44] CBO coupling between object. URL: <http://stackoverflow.com/questions/27515541/cbo-coupling-between-object>. [Son ziyaret tarihi: 13-Mayıs-2017].
- [45] Cyclomatic complexity - Wikipedia. URL:

- https://en.wikipedia.org/wiki/Cyclomatic_complexity. [Son ziyaret tarihi: 13-Mayıs-2017].
- [46] K.-J. Stol ve M. Ali Babar, "A Comparison Framework for Open Source Software Evaluation Methods," Springer, Berlin, Heidelberg, 2010, sf. 389–394.
- [47] I. Stamelos, L. Angelis, A. Oikonomou, ve G. L. Bleris, "Code quality analysis in open source software development," *Inf. Syst. J.*, vol. 12, no. 1, sf. 43–60, Ocak. 2002.
- [48] D. A. Wheeler, "How to evaluate open source software / free software programs," sf. 1–15, 2003.
- [49] W. J. Sung, J. H. Kim, ve S. Y. Rhew, "A Quality Model for Open Source Software Selection," *Sixth Int. Conf. Adv. Lang. Process. Web Inf. Technol. (ALPIT 2007)*, sf. 515–519, 2007.
- [50] V. Garousi, "Investigating the success factors of open-source software projects across their lifetime," *J. Softw. Eng. Stud.*, 2009.
- [51] A. MOCKUS, R. T. FIELDING, ve J. HERBSLEB, "Two Case Studies of Open Source Software Development Apache and Mozilla."
- [52] A. S. Jadhav ve R. M. Sonar, "Evaluating and selecting software packages: A review," *Information and Software Technology*, vol. 51, no. 3. sf. 555–563, 2009.
- [53] Scitools | Build Notes. URL: <https://scitools.com/build-notes/>. [Son ziyaret tarihi: 22-Mayıs-2017].
- [54] M. Dagpinar ve J. H. Jahnke, "Predicting maintainability with object-oriented metrics - An empirical comparison," *Bildiri Kitabı - Working Conference on Reverse Engineering, WCRE*, 2003, vol. 2003–Ocak, sf. 155–164.
- [55] S. K. Dubey ve A. Rana, "Assessment of maintainability metrics for object-oriented software system," *ACM SIGSOFT Softw. Eng. Notes*, vol. 36, no. 5, p. 1, Sep. 2011.
- [56] A. D. Bakar, A. B. M. Sultan, H. Zulzalil, ve J. Din, "Review on Maintainability Metrics in Open Source software," 2012.
- [57] J. Saraiva, S. Soares, ve F. Castor, "Towards a catalog of Object-Oriented Software Maintainability metrics," in *International Workshop on Emerging Trends in Software Metrics, WETSoM*, 2013, sf. 84–87.
- [58] J. D. A. G. Saraiva, M. S. De França, S. C. B. Soares, F. J. C. L. Filho, ve R. M. C. R. De Souza, "Classifying metrics for assessing Object-Oriented Software Maintainability: A family of metrics' catalogs," in *Journal of Systems and Software*, 2015, vol. 103, sf. 85–101.
- [59] V. R. Basili, L. C. Briand, ve W. L. Melo, "A validation of object-oriented design metrics as quality indicators," *IEEE Trans. Softw. Eng.*, vol. 22, no. 10, sf. 751–761, 1996.
- [60] T. Gyimóthy, R. Ferenc, ve I. Siket, "Empirical validation of object-oriented metrics on open source software for fault prediction," *IEEE Trans. Softw. Eng.*, vol. 31, no. 10, sf. 897–910, Kasım. 2005.
- [61] W. T. Tsai, D. Zhang, Y. Chen, H. Huang, R. Paul, ve N. Liao, "A software reliability model for Web services," 2004, sf. 144–149.
- [62] S. Benlarbi, K. El Emam, N. Goel, ve S. Rai, "Thresholds for Object-Oriented Measures," *Proc. 11th Int. Symp. Softw. Reliab. Eng.*, no. Mart, p. 24, 2000.
- [63] S. Herbold, J. Grabowski, ve S. Waack, "Calculation and optimization of thresholds for sets of software metrics," *Empir. Softw. Eng.*, vol. 16, no. 6, sf. 812–841, Aralık. 2011.
- [64] D. E. Chandra ve P. E. Linda, "Class Break Point Determination Using CK Metrics Thresholds," *Glob. J. Comput. Sci. Technol.*, vol. 10, no. 14, sf. 73–77, 2010.
- [65] B. M. Goel ve P. K. Bhatia, "Analysis of reusability of object-oriented systems using object-oriented metrics," *ACM SIGSOFT Softw. Eng. Notes*, vol. 38, no. 4, p. 1, Temmuz. 2013.
- [66] M. Hitz ve B. Montazeri, "Measuring Product Attributes of Object-Oriented Systems," *Angew. Inform.*, vol. 989, sf. 124–136, 1995.
- [67] S. G. Anuradha Tomar, *International Journal of Engineering Research and Technology IJERT*, no. Vol.1-Sayı 5, (Temmuz-2012). ESRSA Publ, 2012.

- [68] P. Mago, J. & Kaur, "Analysis of quality of the design of the object oriented software using fuzzy logic," in *International Conference on Recent Advances and Future Trends in Information Technology (iRAFIT2012) Proceedings published in International Journal of Computer Applications® (IJCA)*, 2012, sf. 21–25.
- [69] J. D. Musa, A. Iannino, ve K. Okumoto, *Software Reliability -- Measurement, prediction, application*. McGraw-Hill, 1987.
- [70] Mailing List Archive: Apache: CVS. URL: <https://lists.gt.net/apache/cvs/>. [Son ziyaret tarihi: 26-Mayıs-2017].
- [71] Software bug. Wikipedia. URL: https://en.wikipedia.org/wiki/Software_bug. [Son ziyaret tarihi: 16-Mayıs-2017].
- [72] Software Testing Fundamental, URL: '<http://softwaretestingfundamentals.com/defect-severity/>.' [Son ziyaret tarihi: 16-Haziran-2017].
- [73] Home | Capgemini Worldwide. URL: <https://www.capgemini.com/>. [Son ziyaret tarihi: 22-Temmuz-2017].
- [74] SpikeSource. URL: <https://www.crunchbase.com/#/home/index>. [Son ziyaret tarihi: 22-Temmuz-2017].
- [75] Intel Corporation. URL: <https://www.intc.com/investor-relations/default.aspx>. [Son ziyaret tarihi: 22-Temmuz-2017].
- [76] Center for Open Source Investigation- Carnegie Mellon University. URL: <http://www.cmu.edu/silicon-valley/research/cosi/>. [Son ziyaret tarihi: 22-Temmuz-2017].
- [77] R. Jabangwe, J. Börstler, D. Smite, ve C. Wohlin, "Empirical evidence on the link between object-oriented measures and external quality attributes: a systematic literature review," *Empir. Softw. Eng.*, sf. 1–54, 2014.

EK 1

E-posta yoğunluğu (ePY) için ölçüm yapısı (İng. measurement construct)

Bilgi ihtiyacı	Güvenilirlik ve bakım yapılabilirliğin değerlendirilmesi için e-posta yoğunluğunun hesaplanması
Ölçülebilir Kavram	Sürüm bazında e-posta yoğunluğu
Varlık	1. E-posta 2. Geliştirici sayısı 3. Kaynak kod 4. Sürümler
Özellik	1. CVS arşivindeki e-posta listesi 2. Ürün internet sitesindeki geliştirici sayısı 3. SVN kod havuzundaki ürün kaynak kodu 4. Ürün İnternet sitesindeki sürüm listeleri
Temel Ölçüler	1. Geliştiriciler arasındaki e-posta sayısı 2. Ürünlerde çalışan geliştiricilerin sayısı 3. Ürünlerin son sürümündeki kod satır sayısı 4. Ürünlerin sürüm sayısı
Ölçüm Yöntemi	1. Geliştiriciler arasındaki toplam e-posta sayıları sayılarak hesaplanır 2. Geliştiricilerin sayısı toplanarak hesaplanır 3. Son sürümdeki kod satır sayısı hesaplanır 4. Her bir üründe bulunan toplam sürüm sayıları sayılarak hesaplanır
Ölçüm yönteminin Türü	1. Objektif 2. Objektif 3. Objektif 4. Objektif
Ölçek	1. Sıfırdan sonsuza tamsayılar 2. Sıfırdan sonsuza tamsayılar 3. Sıfırdan sonsuza tamsayılar 4. Sıfırdan sonsuza tamsayılar
Ölçek Türü	1. Oransal 2. Oransal 3. Oransal 4. Oransal
Ölçüm Birimi	1. E-posta 2. Geliştirici sayısı 3. Satır sayısı 4. Sürüm
Türetilmiş Ölçüler	E-posta yoğunluğu
Ölçüm Fonksiyonu	$\left(\frac{\text{Geliştiriciler arasındaki e - posta sayısı}}{\text{Geliştirici sayısı}} \right) * (\text{Ürünlerin sürüm sayısı/kod satır sayısı})$
Gösterge	E-posta yoğunluğu
Model	E-posta yoğunluğunun sıfıra yakın olması beklenir.
Karar Kriteri	E-posta yoğunluğunun fazla olması üründeki teknik tartışmaların fazla olduğunu gösterir.

Hata çözüme başarı indeksi (HÇBİ) için ölçüm yapısı

Bilgi ihtiyacı	Güvenilirlik ve bakım yapılabilirliğin değerlendirilmesi için hata çözüme başarı oranının hesaplanması
Ölçülebilir Kavram	Hata çözüme başarı oranı
Varlık	1. Toplam hatalar 2. Çözülen hatalar 3. Kaynak kod
Özellik	1. BugDB veri tabanındaki hata listeleri 2. BugDB veri tabanındaki hata listeleri 3. SVN kod havuzundaki ürün kaynak kodu
Temel Ölçüler	1. Ürünlerin toplam hata sayıları 2. Ürünlerin toplam çözülen hata sayıları 3. Ürünlerin son sürümündeki kod satır sayısı
Ölçüm Yöntemi	1. Ürünlerde karşılaşılan toplam hata sayıları sayılarak hesaplanır. 2. Ürünlerde çözülen toplam hata sayıları sayılarak hesaplanır. 3. Son sürümdeki kod satır sayısı hesaplanır.
Ölçüm yönteminin Türü	1. Objektif 2. Objektif 3. Objektif
Ölçek	1. Sıfırdan sonsuza tamsayılar 2. Sıfırdan sonsuza tamsayılar 3. Sıfırdan sonsuza tamsayılar
Ölçek Türü	1. Oransal 2. Oransal 3. Oransal
Ölçüm Birimi	1. Hata sayısı 2. Hata sayısı 3. Hata sayısı
Türetilmiş Ölçüler	Hata çözüme başarı oranı
Ölçüm Fonksiyonu	$(\text{Ürünlerin toplam çözülen hata sayıları} / \text{Ürünlerin toplam hata sayıları}) / \text{kilo kod satır sayısı}$
Gösterge	Hata çözüme başarı oranı
Model	Hata çözüme başarı oranının %100'e yakın olması beklenir
Karar Kriteri	Hata çözüme başarı oranı ne kadar yüksek ise ürün karşılaşılan hatalar karşısında o kadar başarılıdır ve daha tercih edilebilir bir üründür.

Hata ciddiyet indeksi (HCI) için ölçüm yapısı

Bilgi ihtiyacı	Güvenilirlik ve bakım yapılabilirliğin değerlendirilmesi için hata ciddiyet indeksinin hesaplanması
Ölçülebilir Kavram	Hata ciddiyet indeksi
Varlık	1. Kaynak kod 2. Hatalar
Özellik	1. SVN kod havuzundaki ürün kaynak kodu 2. BugDB veri tabanındaki hata listeleri
Temel Ölçüler	1. Ürünlerin son sürümündeki kod satır sayısı 2. Ürünlerdeki her bir önem derecesindeki hataların sayıları
Ölçüm Yöntemi	1. Son sürümdeki kod satır sayısı hesaplanır. 2. Ürünlerde her bir seviyede bulunan hatalar sayılarak hesaplanır.
Ölçüm yönteminin Türü	1. Objektif 2. Objektif
Ölçek	1. Sıfırdan sonsuza tamsayılar 2. Sıfırdan sonsuza tamsayılar
Ölçek Türü	1. Oransal 2. Oransal
Ölçüm Birimi	1. Satır sayısı 2. Hata sayısı
Türetilmiş Ölçüler	Hata ciddiyet indeksi
Ölçüm Fonksiyonu	$\frac{\text{Ürünlerdeki her bir önem derecesindeki hataların sayıları}}{\text{ürünün kod satır sayısı}}$
Gösterge	Hata ciddiyet indeksi
Model	Hata ciddiyet indeksinin sıfıra yakın olması beklenir
Karar Kriteri	Ürünlerde bulunan hatalar ne derece ciddi hatalarsa ürün o derece tercih edilemez bir üründür.

Ortalama hata çözme süresi (OHÇS) için ölçüm yapısı

Bilgi ihtiyacı	Güvenilirlik ve bakım yapılabilirliğin değerlendirilmesi için ortalama hata çözme süresine göre hata çözme başarı oranının hesaplanması
Ölçülebilir Kavram	Ortalama hata çözme süresine göre hata çözme başarı oranı
Varlık	<ol style="list-style-type: none"> 1. Hatalar 2. Hatalar 3. Hatalar
Özellik	<ol style="list-style-type: none"> 1. BugDB veri tabanındaki hata listeleri 2. BugDB veri tabanındaki hata listeleri 3. BugDB veri tabanındaki hata listeleri
Temel Ölçüler	<ol style="list-style-type: none"> 1. Son iki yılda açılan hataların açılma zamanı 2. Son iki yılda açılan hataların kapanma zamanı 3. Son iki yılda açılan toplam hata sayısı
Ölçüm Yöntemi	<ol style="list-style-type: none"> 1. Son iki yılda açılan toplam hatalar sayılarak hesaplanır. 2. Son iki yılda kapanan toplam hatalar sayılarak hesaplanır. 3. Son iki yılda açılan toplam hatalar sayılarak hesaplanır.
Ölçüm yönteminin Türü	<ol style="list-style-type: none"> 1. Objektif 2. Objektif 3. Objektif
Ölçek	<ol style="list-style-type: none"> 1. Sıfırdan sonsuza tamsayılar 2. Sıfırdan sonsuza tamsayılar 3. Sıfırdan sonsuza tamsayılar
Ölçek Türü	<ol style="list-style-type: none"> 1. Oransal 2. Oransal 3. Oransal
Ölçüm Birimi	<ol style="list-style-type: none"> 1. Gün 2. Gün 3. Hata
Türetilmiş Ölçüler	Ortalama hata çözme süresine göre hata çözme başarı oranı
Ölçüm Fonksiyonu	<p>Gün bazında kapanan hata sayısı =Son iki yılda açılan hataların kapanma tarihi - Son iki yılda açılan hataların açılma tarihi=T1</p> <p>Son iki yılda açılan toplam hata sayısı=T2</p> <p>Hata oranı =T1/T2</p>
Gösterge	Kümülatif ortalama hata çözme süresine göre hata çözme başarı oranı grafiği
Model	Her bir ürünün son iki yılda açılan hatalarının ortalama çözülme süresinin sifıra yakın olması beklenir
Karar Kriteri	Ortalama hata çözme süresi ne kadar düşükse ürünlerde çalışan kod geliştiricilerin başarısı o kadar yüksektir.

Hata yoğunluğu (HY) için ölçüm yapısı

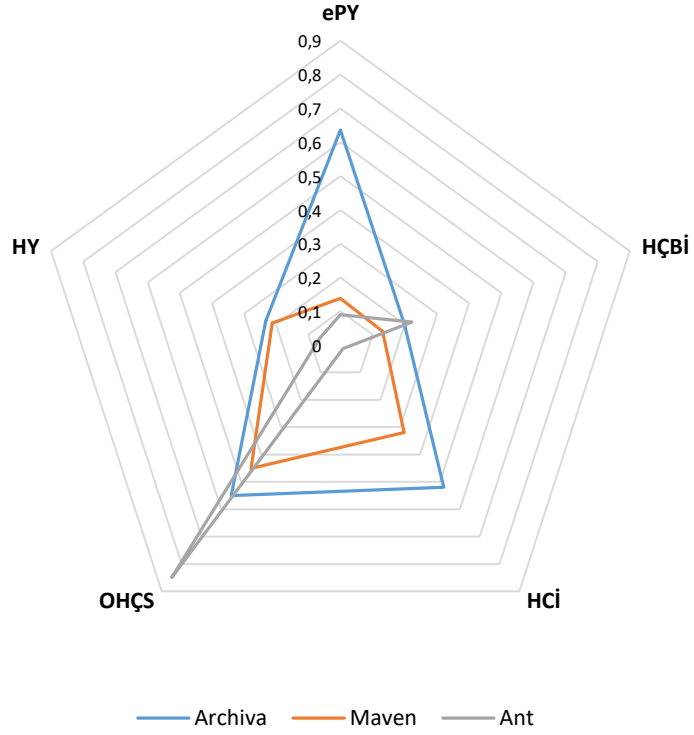
Bilgi ihtiyacı	Güvenilirlik ve bakım yapılabilirliğin değerlendirilmesi için hata yoğunluğunun hesaplanması
Ölçülebilir Kavram	Hata yoğunluğu
Varlık	1. Hatalar 2. Kaynak kod
Özellik	1. Ürün İnternet sitesindeki kod satır sayısı 2. SVN kod havuzundaki ürün kaynak kodu
Temel Ölçüler	1. Hata sayısı 2. Her ürünün kod satır sayısı
Ölçüm Yöntemi	1. Toplam hata sayıları sayılarak hesaplanır 2. Her bir üründe bulunan kod satır sayısı hesaplanır.
Ölçüm yönteminin Türü	1. Objektif 2. Objektif
Ölçek	1. Sıfırdan sonsuza tamsayılar 2. Sıfırdan sonsuza tamsayılar
Ölçek Türü	1. Oransal 2. Oransal
Ölçüm Birimi	1. Hata sayısı 2. Satır sayısı
Türetilmiş Ölçüler	Hata yoğunluğu
Ölçüm Fonksiyonu	Her bir üründe bulunan toplam hata sayısı/o üründeki kod satır sayısı
Gösterge	Hata yoğunluğu
Model	Hata yoğunluğunun sıfıra yakın olması beklenir
Karar Kriteri	Hata yoğunluğu ne az ise ürün o kadar tercih edilebilirdir.

EK 2



Yukarıdaki şekilde değerler hesaplanırken her bir metrik değeri için ürünlere Çizelge 6.2’de verilen ağırlıklar dikkate alınmıştır. Ürün bir metrik bakımından en iyi ürün ise ağırlık olarak 3, en kötü ürün ise 1 ve orta derece iyi ise 2 ağırlık verilmiştir. Örneğin, çözümlenebilirliği ölçmek için üç metrik değeri (WMC, NOS, CC) kullanılmıştır. Bu üç metriğin ölçüm sonuçlarına göre Archiva’nın ağırlık değerlerine baktığımızda, iki metrik için (WMC, CC) en iyi ürün olarak gözlemlenmiş (her bir metrik değeri için ağırlık değeri 3) ve bir tanesi için (NOS) ise en kötü ürün olarak (ağırlık değeri 1) olarak gözlemlenmiştir. Dolayısıyla test edilebilirlik için Archiva’nın toplam ağırlık değeri 7 ($3+3+1=7$) olmuştur.

Bilgi ihtiyaçları bakımından toplum-tabanlı deęerlendirme sonuları



Yukarıdaki Őekilde deęerlerler hesaplanırken Blm 6.2'deki her bir bilgi ihtiyaı bakımından rnler iin hesaplanan indeks deęerleri dikkate alınmıŐtır. Őekil incelendięinde, OHS bilgi ihtiyaı iin Ant rnnn aık bir Őekilde daha iyi olduęu gzlemlenmiŐtir. Ancak genel anlamda baktıęımızda Archiva rnnn dięer iki rne gre daha iyi olduęu gzlemlenmiŐtir.

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı: Nebi YILMAZ

Doğum Yeri: Kilis

Medeni Hali: Bekâr

E-posta: nebiyilmaz10@gmail.com

Adresi: Hacettepe Üniversitesi Bilgisayar Mühendisliği Bölümü, 121 nolu ofis

Eğitim

Lise: Hacı Mehmet Koçarslan Anadolu Lisesi, Kilis, 2006

Lisans: Gaziantep Zirve Üniversitesi, Bilgisayar Mühendisliği, Gaziantep, 2014

Yabancı Diller

İngilizce

İş Deneyimi

Araştırma Görevlisi, Hacettepe Üniversitesi, 2014-2017

Deneyim Alanları

Yazılım Mühendisliği, Yazılım Kalite

Tezden Üretilmiş Projeler ve Bütçesi

-

Tezden Üretilmiş Yayınlar

Yılmaz, N., ve Dinçer, K. "Açık Kaynak Yazılım Seçimi için İki Boyutlu Değerlendirme Metodu" Ulusal Yazılım Mühendisliği Sempozyumu'16, Vol-1721, p. 325-336, Çanakkale, Türkiye (2016)

Tezden Üretilmiş Tebliğ 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: 26.07/2017

Tez Başlığı / Konusu: ACIK KAYNAK YATILIMLARDA BAKIM YAPILABİLİRLİK VE
GÜVENİRLİK ELÇİMLERİ İÇİN İKİ BAYITIM DEĞERLENDİRME METODU

Yukarıda başlığı/konusu gösterilen tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler ve d) Sonuç kısımlarından oluşan toplam sayfalık kısmına ilişkin, 26.07/2017 tarihinde şahsım/tez danışmanım tarafından turnitın adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 6 'tür.

Uygulanan filtrelemeler:

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

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü Tez Çalışması Orjinallik 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: Nebi YILMAZ

26/07/2017

Öğrenci No: 111227742

Anabilim Dalı: Bilgisayar Mühendisliği

Programı: Bilgisayar Mühendisliği

Statüsü: Y.Lisans Doktora Bütünleşik Dr.

DANIŞMAN ONAYI

UYGUNDUR.

Yrd. Doç. Dr. Ayca TARHAN

(Unvan, Ad Soyad, İmza)

(Unvan, Ad Soyad, İmza)