

Bileşen Tabanlı ve Ürün Hattı Yazılım Geliştirme Yaklaşımlarında Yeniden Kullanılabilirlik Metrikleri

Ezgi Cankurtaran¹, Evren Çilden², Ayça Tarhan³

^{1,3} Bilgisayar Mühendisliği Bölümü, Hacettepe Üniversitesi, Ankara, Türkiye
{ezgicankurtaran, atarhan}@hacettepe.edu.tr

^{1,2} Aselsan, REHİS
{ecankurtaran, ecilden}@aselsan.com.tr

Özet. Yeniden kullanım, yazılım kalitesini artırma, geliştirme maliyetini düşürüp süreci kısaltma potansiyeli taşıması nedeniyle yazılım geliştirmenin önemli pratiklerinden biridir. Yeniden kullanım farklı yazılım geliştirme yaklaşımlarında uygulanırsa da Bileşen Tabanlı Yazılım Geliştirme ve Yazılım Ürün Hattı Geliştirme yaklaşımının temel kavramıdır. Literatürde yeniden kullanılabilirliği analiz etmek için farklı yöntem ve metrikler sunulmuştur. Bu bildiri, bileşen tabanlı geliştirmede ve ürün hattı geliştirmede yeniden kullanılabilirliği ölçmek için yapılmış çalışmaları, sistematik yöntemle araştırarak sunmayı hedeflemiştir.

Abstract. Software reuse is an important practice in software development because of its potential to reduce costs by shortening development time and increasing software quality. Although software reuse is applied in different development methodologies, it is the main concept in Component Based Software Development and Software Product Line Development approaches. The academia has proposed different methods and metrics to analyze software reusability. This paper presents studies to measure software reusability in component based development and product line development by applying a systematic analysis of related literature.

Anahtar Kelimeler: Yeniden kullanılabilirlik metrikleri, bileşen tabanlı yazılımlarda yeniden kullanılabilirlik, ürün hattı yaklaşımında yeniden kullanılabilirlik, bileşen tabanlı yazılım, ürün hattı, yeniden kullanım

1 Giriş

Yazılım Mühendisliği araştırma çalışmaları, yazılım geliştirme süreçlerinin verimliliğinin artmasına, kalitenin gelişmesine, maliyetin düşmesine ve zaman yönetiminin desteklenmesine odaklanmaktadır. Var olan yazılımın yeniden kullanımı güvenilirliği artırırken, süreç riskini düşürüp kaynakların verimli kullanılmasını sağlamaktadır. Günümüzde, hızlı ve sürekli teslimatı gerektiren yazılım ihtiyaçları, yazılım bileşenlerinin veya ürün hatlarının yeniden kullanımı kavramını da beraberinde getirmektedir. Kısıtlı iş gücünün olduğu küçük ekiplerde yeniden kullanılabilirlik ayrıca önem taşımaktadır [1].

Kısıtlı zaman ve iş gücünün olduğu yazılım geliştirme projelerinde yeniden kullanılabilirlik değerlerini sayısallaştırıp anlamlandırmak için literatürde çok sayıda metrik ve yöntem önerilmiştir. Metriklerden elde edilen değerlere göre; yeniden kullanılabilir olan bileşenleri belirlemek, yeniden kullanımı söz konusu olan bileşenlerden ne kadar verim elde edildiğini saptamak, yazılım tasarımına karar destek sağlayacak gerçekçi ölçüm değerleriyle geleceğe yönelik tahminleme yapmak ve ortak mimaride yeniden kullanımı arttırmaya yönelik girdi sağlamak mümkündür [2].

Bu çalışmanın amacı; bileşen tabanlı veya ürün hattı yaklaşımı uygulanarak geliştirilmiş yazılımların, yeniden kullanılabilirliğini ölçmek için literatürde sunulan metrikleri araştırmaktır. Uygulanan sistematik araştırma yöntemi sonucuna göre; bileşen tabanlı yazılımların yeniden kullanılabilirliğini ölçmeye yönelik literatürde birçok çalışma bulunurken ürün hattı yaklaşımının yeniden kullanılabilirliğini belirlemeye yönelik olarak metrikleri araştıran çalışmalarda açığın olduğu görülmüştür.

Bildirinin 2. Bölümü'nde yeniden kullanım ve yeniden kullanılabilirlik kavramlarının yazılım geliştirme metodolojileri açısından tanımı yapılmakta, 3. Bölümü'nde daha önce benzer konularda yapılmış literatür çalışmalarından söz edilmekte, 4. Bölümü'nde çalışmada uygulanan araştırma metodolojisi açıklanmakta, 5. Bölümü'nde çalışmanın sonuçlarına yer verilmektedir.

2 Yeniden Kullanım ve Yeniden Kullanılabilirlik

Yazılımda yeniden kullanılabilirlik kavramını anlamak için yeniden kullanım ve yeniden kullanılabilirliği ayırt etmek gerekmektedir. Frakes ve Succi [2] yeniden kullanımı (İng. Reuse), “*var olan yazılım ve yazılım ürünlerinin yeni yazılım geliştirmede direkt kullanımı*” olarak tanımlamaktadır. Aynı araştırmacılar yeniden kullanılabilirliği (İng. Reusability) ise yeniden kullanım olasılığını gösteren yazılım özelliği olarak tanımlamaktadır [2].

Yeniden kullanılabilirliğin literatüre kazandırılması 1968 yılında McIlroy'in yeniden kullanılabilir bileşenleri NATO seminerinde sunmasına dayanmaktadır. Ne var ki pratikte yeniden kullanılabilirliğe ilişkin çalışmalar 1980'lerin sonuna kadar sınırlı kalmıştır. 1990'ların sonunda yeniden kullanılabilirlik ile ilgili yürütülen çalışmalar hız kazanmıştır. Milenyuma giriş, beraberinde birçok bileşenin geniş bir bantta yeniden kullanılmasını beraberinde getirmiştir [3].

Yeniden kullanım tüm yazılım geliştirme yaklaşımlarında esas olsa bile bileşen tabanlı yazılım geliştirmenin (İng. Component Based Software Development - CBSD) ve ürün hattı yaklaşımının (İng. Software Product Line) ana kavramıdır. Bileşen tabanlı yazılım geliştirme, yeniden kullanılabilir bileşenler kullanarak yazılım tasarlanmanın altını çizmektedir [4]. Yazılım ürün hattı ise belirli bir görevin ihtiyacını karşılamak üzere, ortak yetenek kümelerinden oluşan sistemlerin kullanımını esas almaktadır [5]. Bileşenlerin ve ürün hattı yaklaşımlarının yeniden kullanılabilirliğini değerlendirmek için yazılım karakteristikleri, metrikler kullanılarak sayısallaştırılmaktadır.

3 İlişkili Çalışmalar

Sandhu ve Aashima'nın yapmış oldukları çalışmada [6], yazılım bileşenlerinin yeniden kullanılabilirliğini değerlendirmek ve tahmin etmek için literatürde sunulmuş farklı modeller ve ölçümler tartışılmaktadır.

Parmeza ve Fifo'nun yürütmüş olduğu benzer çalışmada [4], literatürde var olan bileşen tabanlı yazılımlar için sunulmuş yeniden kullanılabilirliği attıran ve ölçen metotlar araştırılmıştır. Çalışmada bu metotların avantajları ve dezavantajları tartışılmıştır.

Koteska ve Velinov çalışmalarında [7], bileşenler için sunulmuş yeniden kullanılabilirlik metriklerini araştırmaktadır. Çalışmada, yeniden kullanılabilirliği etkilediği düşünülen yeni özgün özellikler önerilmiştir. Evrensel olarak kabul göre yeniden kullanılabilirlik metriklerindeki açığı gidermek için yeniden kullanılabilirlik metrikleri için birleştirilmiş model önerilmektedir.

Singh ve arkadaşları yapmış oldukları çalışmada [8], bileşen tabanlı yazılım geliştiricileri desteklemek için; literatürde bileşenler için sunulmuş, doğrudan ya da dolaylı yoldan ölçülen yeniden kullanılabilirlik metriklerini araştırmıştır. Çalışmada yeniden kullanılabilirlik *White Box*, *Glass Box* ve *Black Box* bileşenler için ayrı ele alınmıştır.

Kumar ve arkadaşları yapmış oldukları çalışmada [9], bileşen tabanlı yeniden kullanılabilirliğe ilişkin literatürde sunulan yaklaşımları ve metrikleri sistematik olarak araştırmıştır. Belirledikleri araştırma kriterine göre toplam 49 yayın incelenmiş ve elde edilen sonuçlar karşılaştırılmıştır. Yeniden kullanılabilirliği belirlemek için kullanılan teknik, yazılımın hangi karakteristiğinin ele alındığına bağlı olduğu belirtilmiştir.

Mijac ve Stapic yapmışları oldukları çalışmada [10], sistematik araştırma yöntemi uygulayarak yazılım bileşenlerinin yeniden kullanılabilirliğini belirlemeye yönelik sunulan yöntemleri ve metrikleri araştırmıştır.

Bu çalışma, yeniden kullanımı ilke edinen bileşen tabanlı yazılım geliştirmede ve ürün hattı yaklaşımında, bileşenlerin yeniden kullanılabilirliğini ölçmek ve değerlendirmek için literatürde sunulan metrikleri ve yöntemleri araştırmaktadır. Literatürde, bu iki kavramın birlikte sistematik olarak araştırıldığı çalışmaya rastlanmamıştır. Ayrıca, ürün hattı yazılım geliştirme yaklaşımlarında yeniden kullanılabilirlik metriği ya da modeli sunan çalışmaların azlığından dolayı bu alanda literatür taraması bulunmamaktadır. Çalışmamızda yeniden kullanım seviyesini ölçen metrikler, daha sonra araştırılmak üzere hariç tutulmuştur.

4 Araştırma Metodolojisi

Bileşen bazında yeniden kullanılabilirlik metrikleri için literatür taraması yapılırken eksiksiz ve tarafsız bir makale havuzu oluşturmak için Kitchenham'ın sunmuş olduğu Sistematik Literatür Tarama (İng. Systematic Literature Review) kılavuzundan faydalanıldı [11]. Sistematik literatür taraması, spesifik bir araştırma sorusu ile ilgili olan mümkün olan tüm araştırma sonuçlarının değerlendirilmesi ve yorumlanmasına dayalı olan bir metodolojidir. Bu yöntemde, araştırmanın titiz, tarafsız ve tekrarlanabilir olarak yapılması hedeflenmektedir. Kılavuza göre ilgili makaleler seçilirken; araştırma so-

rusu (İng. Research question), seçim kriteri (İng. Selection criteria), veri çıkarma stratejisi (İng. Data extraction strategy) ve çalışmanın kalite değerlendirmesi (İng. Study quality assessment) dikkate alındı.

4.1. Araştırma Hedefi ve Soruları

Çalışmada, yeniden kullanılabilirliği sayısallaştırmak (nicel yöntemlerle değerlendirmek) amaçlı yapılan çalışmaları araştırmak için, Hedef Soru Metrik (İng. Goal-Question-Metric) yaklaşımı [12] uygulanmıştır. Bu yaklaşıma göre araştırma hedefi ve soruları belirlenmiş, türetilen araştırma sorularına cevaben, bileşen tabanlı yazılımlar ve ürün hattı yaklaşımı için yeniden kullanılabilirliği ölçmeye yönelik olarak literatürde sunulmuş metrik setleri elde edilmiştir.

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

AS-1: Bileşen-tabanlı ve ürün-hattı yazılım geliştirme yaklaşımlarında yeniden kullanılabilirliği ölçmek için hangi metrikler önerilmiştir? Bu metrikler hangi yöntemlerle birlikte kullanılmıştır?

AS-2: Bileşen-tabanlı ve ürün-hattı yazılım geliştirme yaklaşımlarında yeniden kullanılabilirliği ölçmek için yapılan çalışmalarda, yeniden kullanılabilirliği etkileyen faktörler nelerdir?

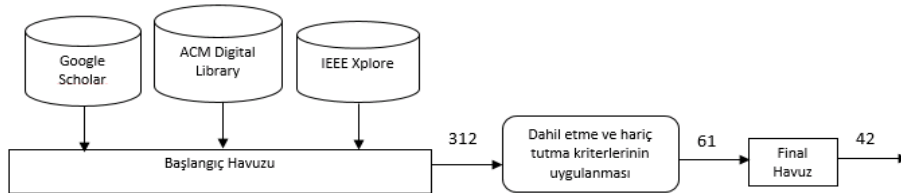
4.2 Araştırma Stratejisi

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

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

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

Araştırma yapılırken Google Scholar diğer akademik veri tabanlarını indekslemesi özelliği ile öncelikli veri tabanı oldu. İlgili makalelerin havuza dâhil edilmesi için, makalelerin özet bölümleri, başlıkları, anahtar kelimeleri ve sonuç bölümleri incelendi. Şekil 1, final makale havuzunun nasıl oluşturulduğunu göstermektedir. Okların üzerindeki sayısal değerler, her adım sonunda havuzda kalan makale sayısını göstermektedir.



Şek. 1. Final Makale Havuzunu Oluşturma Adımları

4.3. Seçim Kriteri

Makalelerin seçimi Tablo 1’de tanımlanan dâhil etme ve hariç tutma kriterlerine göre gerçekleştirilmiştir. İlk seçim, sorgu sonucu veri tabanlarından dönen makalelerin toplanması ile gerçekleştirilmiştir. Çakışan makaleler elimine edilmiştir. Makaleler, özet ve başlık bilgilerine bakılarak değerlendirilmiştir. Kapsamla ilgili olmadığı görülen makaleler başlangıç havuzundan atılmıştır. Geri kalan makalelerde daha derin inceleme yapılmış, Tabloda belirtilen kriterlere göre havuzdan uzaklaştırılmış ya da kalması yönünde karar verilmiştir. Örneğin kapsam ve alan bakımından yeniden kullanılabilirlik ile ilişki olduğu görülen bir makale, yeniden kullanım seviyesini tanımlayan bir çalışma ise (metrikler üzerinden sayısal değerlendirme içermediğinden) havuzdan atılmalıdır.

Tablo 1. Dahil Etme Kriteri ve Hariç Tutma Kriterleri

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

4.4. Yazılım Kalite Değerlendirmesi

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

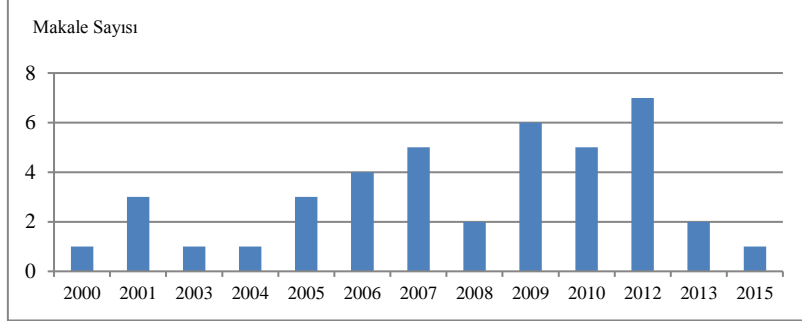
4.5. Veri Çıkarma Stratejisi

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

5 Araştırma Sonuçları

Bileşen tabanlı ve ürün hattı yazılım geliştirme yaklaşımları ile ilgili olarak yapılmış olan çalışmaların yıllara göre dağılımları Şekil 2’de verilmektedir.

Çalışmalar uyguladıkları yöntemler ve odaklandıkları konular açısından çeşitlilik göstermektedir. Yazılımın yeniden kullanılabilirliği birçok faktörden etkilendiği için literatürde farklı yaklaşımlar denenmiştir. Çalışmaların bazıları yazılım içindeki yeniden kullanılabilirliğe uygun parçaların belirlenmesine yönelik iken bazıları bileşenlerin yeniden kullanılabilirliğini değerlendirmeye çalışmaktadır. Tablo 2’de araştırma sorularına yanıt olarak elde edilen metrik, yöntem ve faktörler verilmiştir.



Şek. 2. Yıllara Göre Makale Sayısı

Tablo 2. Hedef-Soru-Metrik Ağacı

Hedef	Bileşen-tabanlı ve ürün-hattı yazılım geliştirme yaklaşımlarında yeniden kullanılabilirliği ölçmek için yapılan çalışmaları anlamak	
	Soru (AS-1)	Bileşen-tabanlı ve ürün-hattı yazılım geliştirme yaklaşımlarında yeniden kullanılabilirliği ölçmek için hangi metrikler önerilmiştir? Bu metrikler hangi yöntemlerle birlikte kullanılmıştır?
	Metrikler (AS-1'in yanıtları)	<ul style="list-style-type: none"> - Bileşenlerin Tam Yeniden Kullanılabilirliği (İng. COR - Component Overall Reusability) - Bulanık temelli yaklaşım, Nöro bulanık temelli yaklaşım, Hibrit yaklaşım kullanarak yeniden kullanılabilirlik seviyesini belirleme (İng. Level of reusability using: Fuzzy-based approach, Neuro-fuzzy based approach, Hybrid fuzzy-GA based approach) - Kalite Sistemi (İng. Quality Framework) f(u) – Yeniden kullanılabilirlik modeli, yeni bir metrik sistemi, Segueno tip bulanık arayüz sistemi (İng. Reusability using Segueno-Type Fuzzy Inference System, Reusability model, new metrics framework {EMI, RCO, RCC, SCC, SCCp}) - Nöro bulanık arayüz sistemi, Bulanık yaklaşım kullanarak yeniden kullanılabilirliğin seviyesi (İng. Level of reusability using: Neuro-fuzzy inference system, Soft Computing Fuzzy approach)
	Soru (AS-2)	Bileşen-tabanlı ve ürün-hattı yazılım geliştirme yaklaşımlarında yeniden kullanılabilirliği ölçmek için yapılan çalışmalarda, yeniden kullanılabilirliği etkileyen faktörler nelerdir?
	Metrikler (AS-2'nin yanıtları)	<p>Adapte olabilirlik (İng. Adaptability), Bağlaşım ve Yapışkanlık (İng. Coupling, Cohesion), Karmaşıklık (İng. Complexity), Uyarlanabilirlik (İng. Customizability), Tamamlanabilirlik (İng. Completeness), Gizlilik (İng. Confidence), Ayarlanabilirlik (İng. Configurability), Uygunluk (İng. Compatibility), Ortaklanabilirlik (İng. Commonality), Birleştirilebilirlik (İng. Composability), Anlaşılabilirlik (İng. Comprehensibility), Dokümantasyon (İng. Documentation), Alan İçeriği (İng. Domain context), Extent of templating, Kullanımda kolaylık (İng. Ease-of-use), Fonksiyonel bağımlılık (İng. Functional dependencies), Fonksiyonellik (İng. Functionality), Kalıtım (İng. Inheritance), Arayüz karmaşıklığı (İng. Interface complexity), Arayüzlerin sağlamlığı (İng. Interface soundness), Bakım kolaylığı (İng. Maintainability), Modülerlik (İng. Modularity), Taşınabilirlik (İng. Portability), Güvenilirlik (İng. Reliability), Yeniden kullanım frekansı (İng. Reuse frequency), Düzenlilik (İng. Regularity), Bağlaşımın ayrılması (İng. Separation of Concerns), Büyüklük (İng. Size), Nakledilebilirlik (İng. Transplanting), Anlaşılabilirlik (İng. Understandability), Kullanılabilirlik (İng. Usability), Yararlılık (İng. Usefulness), Kullanılabilirlik (İng. Utilizability), Değişkenlik (İng. Variability), Hacim (İng. Volume)</p>

İlk araştırma sorusu (AS-1) kapsamında, bileşen tabanlı yazılımların yeniden kullanılabilirliğini değerlendirmeye yönelik olarak sunulmuş çalışmalar EK-A'daki Tablo 4'te sunulmuştur. Tablodaki "Yazılım Özellikleri" sütunu, çalışmalarda yeniden kullanılabilirliği etkilediği düşünülen yazılım özelliklerini; "Yazılım Metrikleri" sütunu, yeniden kullanılabilirliği etkileyen yazılım özelliklerini ölçmek için kullanılan metrikleri; "Yeniden Kullanılabilirliği Ölçme Yöntemleri ve Metrikleri" sütunu ise yeniden kullanılabilirliği ölçmek için önerilen yöntem ve metrikleri ifade etmektedir.

Bileşen tabanlı yazılımların yeniden kullanılabilirliğini araştıran çalışmalarda sunulan metot ve/veya metrikler uygulamalarla doğrulanmıştır. Örneğin Sandhu ve Singh çalışmalarında [20], 6 sınıf içinden 48 örnek seti çıkaran geliştirilmiş sisteme başvurulmuştur. Bulanık arayüz sistemindeki değişkenler, girdi parametrelerin ölçümlerine bağlı olarak, girdi parametrelerine atanmıştır. Bu çalışmada yeniden kullanılabilirlik 6 farklı değışkene atanmıştır (PERFECT, HIGH, MEDIUM, LOW, VERY LOW ve NIL). 200 iterasyon sonuna normalize edilmiş kurallar, yazılımın yeniden kullanılabilirliğini değerlendiren bir sistem sunmuştur.

Torkamani çalışmasında [5], ürün hatlarının yeniden kullanılabilirliğini değerlendirmeye yönelik olarak özgün metrik takımları önermiştir. Yeniden kullanımdan elde edilen kar, bir bileşenin yeniden kullanımının bir ürünün elde edilmesine olan etkisi gibi metrikler sunulmaktadır. Yazılım parçalarının ağırlıklı değerleri dikkate alınarak metrik önerilmektedir.

Zhang ve arkadaşları [47], ürün hattı mimarisinin kalitesini belirlemek için yeni metrik takımı önerilmiştir. Çalışma doğrudan yeniden kullanılabilirliğe odaklanmamıştır. Ürün hattı kalitesini etkileyen faktörler olarak yeniden kullanılabilirlik, değışkenlik, karmaşıklık ve benzerlik gibi özellikler göz önüne alınmıştır. Ürün hattı mimarisi tanımlama dili (İng. Product Line Architecture Description Language), analiz edilerek mimari kalitesinin değerlendirilmesi yaklaşımı benimsenmiştir.

Her ve arkadaşlarının yapmış oldukları çalışmada [48], ürün hattındaki değerlerin yeniden kullanılabilirliğini değerlendirmeye yönelik olarak çerçeve sunulmaktadır. Çalışmada, ürün hattının yeniden kullanılabilirliğini etkileyen kalite faktörleri belirlenmiş ve belirlenen bu faktörleri ölçmek için metrikler sunulmuştur. Yazılım özelliklerinin değışkenlik gösterdiği ya da ortaklandığı noktalar göz önüne alınmıştır.

İkinci araştırma sorusu (AS-2) kapsamında, bileşen tabanlı yazılımlarda yeniden kullanılabilirliği en çok etkileyen ilk üç yazılım özelliğinin çalışmalara göre dağılım frekansları Tablo 3'te sunulmuştur.

Kara Kutu (İng. Black Box) olarak kabul edilen yazılımların bütün olarak yeniden kullanılabilirliği değerlendirilmektedir. Bileşen tabanlı yazılımların Kara Kutu olarak ele alındığı yöntemlerde, yazılımın yeniden kullanılabilirliğinin arayüz üzerinden hesaplandığı yaygın olarak görülmektedir.

Tablo 3. Yeniden Kullanılabilirliği Etkileyen Özelliklerin Frekansı ve İlişkili Çalışmalar

Yazılım Özelliği	Makale Sayısı	Referans eden makaleler
Coupling	17	[14],[15],[17],[18],[20],[21],[23],[24],[25],[26],[29],[32],[34],[35],[36],[37],[39]
Cohesion	11	[20],[21],[23],[24],[26],[34],[35],[36],[37],[38],[44]
Complexity	7	[15],[17],[18],[24],[25],[32],[45]

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

6 Sonuçlar ve Gelecek Çalışmalar

Bu çalışma, yazılım tabanlı bileşenlerin veya ürün hattı yaklaşımı ile geliştirilmiş yazılımların yeniden kullanılabilirliğini ölçmek veya değerlendirmek için sunulmuş olan metrik ve yöntemleri sistematik bir şekilde araştırmayı hedeflemiştir. Yapılan araştırma sonucunda, bileşen tabanlı birçok çalışma olduğu gözlenirken [13-46] ürün hattının yeniden kullanılabilirliğine yönelik olarak sunulmuş çalışmaların az olduğu görülmüştür [5, 46, 47]. Yeniden kullanım ilkesine dayanan yazılım ürün hatlarının yeniden kullanılabilirliğini ölçme çalışmalarında boşluk olup bu alanın çalışmaya ve araştırmaya açık olduğu görülmektedir.

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

Gelecek çalışma olarak bileşen temelli yazılımlarda bileşenlerin yeniden kullanılabilirliğini etkileyen faktörlerin yazılım ürün hatlarının yeniden kullanılabilirliğinin değerlendirilmesinde de kullanılıp kullanılmayacağı araştırılacaktır.

EK-A / Tablo 4. Araştırmaya dâhil edilen çalışmaların analizi

Ref.	Yazılım Özellikleri	Yazılım Metrikleri	Yeniden Kullanılabilirliği Ölçme Yöntemleri ve Metrikleri
[13]	Portability, Understandability, Adaptability	Existence of Meta Information (EMI), Rate of Component Customizability (RCC), Self-Completeness of Component's Return Value (SCCr)	COR - Component Overall Reusability
[14]	Coupling	DD - Component Dependency Metric	-
[15]	Complexity, Regularity, Volume, Reuse Frequency, Coupling	McCabe's Cyclomatic Complexity, Regularity metric, Halstead Software Science Indicator, Reuse frequency metric, LCOM	Level of reusability using: Fuzzy-based approach, Neuro-fuzzy based approach, Hybrid fuzzy-GA based approach
[16]	-	-	Quality Framework

[17]	Coupling, Volume, Complexity, Regularity, Reuse Frequency	McCabe's Cyclomatic Complexity, Halstead Software Science Indicator	f(u) - Reusability using Segueno-Type Fuzzy Inference System
[18]	Adaptability, Completeness, Maintainability, Understandability, Complexity, Volume, Inheritance, Coupling	McCable's Complexity, Halstead's Volume, NPSS (Number of Physical Source Statements, Depth of Nested Statements, WMC (Weighted Methods per Class), Depth of Inheritance, Number of immediate children, RFC (Response for a Class), Coupling between objects (CBO).	-
[19]	Understandability, Adaptability, Portability	-	Reusability model, new metrics framework {EMI, RCO, RCC, SCC, SCCp}
[20]	Coupling, Cohesion, Inheritance	CK metrics (WMC, DIT, NOC, CBO, LCOM)	Level of reusability using: Neuro-fuzzy inference system
[21]	Coupling, Cohesion, Inheritance	CK metrics (WMC, DIT, NOC, CBO, LCOM)	Neuro-fuzzy inference system
[22]	Customizability, Interface complexity, Portability, Documentation	-	Soft Computing Fuzzy approach
[23]	Coupling, Cohesion, Inheritance	CK metrics (WMC, DIT, NOC, CBO, LCOM), Regression algorithms	-
[24]	Modularity, Coupling, Cohesion, Interface complexity, Documentation, Complexity, Size	Modularity = f(Lack of Coupling, Cohesion), Interface size = Adjusted Number of public methods, Documentation = A degree of documented lines of code, methods, attributes, Complexity = f(Size, Complexity of Methods)	Reusability = f(Modularity, Interface Size, Documentation, Complexity)
[25]	Complexity, Regularity, Volume, Reuse Frequency, Coupling	-	Neuro-fuzzy inference system
[26]	Cohesion, Coupling	COHOT - Cohesion Metric by Operation Type, COPOT - Coupling Metric by Operation Type	-
[27]	Reusability in accordance to Reuse level	-	CRL_{LOC} - Component Reuse Level
[28]	Domain Context	-	$R = f(Rf)$, $R = f(Rf, Cu)$, $R = f(Rf, Cu, Rm)$
[29]	Coupling	CC - Coupling between Classes, CPC - Coupling of Component, CBC - Coupling between Component	-
[30]	Customizability, Configurability, Interface complexity, Portability, Compatibility	-	Fuzzy inference system
[31]	Commonality, Modularity, Adaptability, Comprehensibility, Composability, Interface soundness	CoS - Cohesiveness of Services, MoD - Minimality of Dependencies, CoV - Coverage of Variability, CoA - Completeness of Variant Set, ESA - Effectiveness of Service Adaptation, ICC - Inter-CaaS Composability, CAC - CaaS-to-Application Composability	IR - Integrated reusability

[32]	Complexity, Volume, Regularity, Reuse Frequency, Coupling	McCabe's Cyclomatic Complexity metric, Halstead Software Science Indicator, Regularity Metric, ReuseFrequency Metric, LCOM	Neural Network - Resilient Backpropagation algorithm (RB)
[33]	Understandability	APP - Arguments per Procedure, DAC - Distinct Argument Count, DAR - Distinct Argument Ration, ARS - Argument Repetition Scale, MSC - Mean String Commonality, MIL - Mean Identifier Length, RAD - Reference Argument Density	-
[34]	Coupling, Cohesion	WTcoh - Weighted Transitive Cohesion measure, WTcoup - Weighted Transitive Coupling measure	-
[35]	Coupling, Cohesion, Inheritance	CK metrics (WMC, DIT, NOC, CBO, LCOM)	Neural Network - Levenberg & Marguardt algorithm
[36]	Coupling, Cohesion, Inheritance	CK metrics (WMC, DIT, NOC, CBO, LCOM)	K-means clustering algorithm & Decision Approach
[37]	Separation of Concerns, Size, Cohesion, Coupling	-	-
[38]	Cohesion	PCoh - Package Cohesion	-
[39]	Coupling	CC1 - CC8 (Coupling Counts)	-
[40]	Functionality, Reliability, Maintainability, Ease-of-use, Transplanting	-	RV - Reusability Value based on Colony Decision
[41]	Functionality, Reliability, Maintainability, Utilizability, Portability	-	RMV (Reusability Measure Value)
[42]	Adaptability, Composability	Component Adaptability (Ac(kc)), Composability Degree of Component	-
[43]	Reuse Frequency, Understandability, Rate of modification	-	$R = f(Rf)$, $R = f(Rf, Cu)$, $R = f(Rf, Cu, Rm)$
[44]	Cohesion	SBFC - Similarity-Based Functional Cohesion	-
[45]	Adaptability, Availability, Complexity, Documentation, Maintainability, Price, Reuse, Quality	-	R(C) - Reusability of component
[46]	Interface complexity	No. public members in component interface	R(C) - Reusability of component

Referanslar

1. Seungwon Lee and Ho-Jin Choi: Software Component Reusability Measure In Component Grid, 11th International Conference on Advanced Communication Technology, PyeongChang, South Korea, Vol. 1, pp. 576–578, (2009).
2. W.Frakes, G. Succi: An Industrial Study of Reuse, Quality and Productivity, Journal of Systems and Software, Vol:57, pp:99-106, (2001).

3. O.P. Rotaru and M Dobre,: Reusability Metrics for Software Components, the 3rd ACS/IEEE International Conference on Computer Systems and Applications, Cairo, Egypt, (2005).
4. D. Parmeze, M. Fifo: A Survey of Methods for Measuring and Enhancing Component Reusability , J. Converg. Information Technology, Vol. 4, (2009).
5. M. A. Torkamani: Metric Suite to Evaluate Reusability of Software Product Line, International Journal of Electrical and Computer Engineering (IJECE), Vol:4 No:4, pp: 285-294, (2014).
6. P. S. Sandhu and Aashima: A Survey on Software Reusability, 2010 International Conference on Mechanical and Electrical Technology (ICMET 2010), pp: 769- 773, (2010).
7. B. Koteska and G. Velinov: Component-Based Development: A Unified Model of Reusability Metric, ICT Innovation, Springer Berlin Heidelberg, pp. 335–344, (2012).
8. S. Singh, M. Thapa, G. Singh: Software Engineering - Survey of Reusability Based on Software Component, International Journal of Computer Application, Vol:8, No:12, (2010).
9. V.Kumar, A. Sharma, R. Kumar, and P.S. Grover: Quality Aspects for Component- Based Systems: A Metrics Based Approach, Softw. Pract. Exp., Vol: 42, No:12, pp. 1531–1548, (2012).
10. M. Mijac and Z. Stapic,: Reusability Metrics of Software Components: Survey, Central European Conference on Information and Intelligent Systems, (2015).
11. B. Kitchenham: Procedures for Performing Systematic Reviews, Eversleigh, (2004).
12. V. R. Basili, G. Caldiera, H.D. Rombach,: The Goal Question Metric Approach, Encyclopedia of Software Engineering, (1994).
13. H. Washizaki, R. Namiki, T. Fukuoka, Y. Harada, and H. Watanabe: A Framework For Measuring and Evaluating Program Source Code Quality, Product-Focused Software Process Improvement, Vol. 4589, pp. 284–299, (2007).
14. L.G. Yu, S. Ramaswamy: Component Dependency in OO Software, Journal Computer Science and Technologyi Vol:22, pp:379-386, (2007).
15. P. S. Sandhu, S. S. Dalwinder, and H. Singh: A Comparative Analysis of Fuzzy, Neuro-Fuzzy and Fuzzy-GA Based Approaches for Software Reusability Evaluation, Proceedings of World Academy of Science: Engineering & Technology, (2008).
16. H. Washizaki, R. Namiki, T. Fukuoka, Y. Harada, and H. Watanabe: A Framework For Measuring and Evaluating Program Source Code Quality, Product-Focused Software Process Improvement, Vol. 4589, pp. 284–299, (2007).
17. P. S. Sandhu and H. Singh: A Fuzzy-Inference System Based Approach for the Prediction of Quality of Reusable Software Components, International Conference on Advanced Computing and Communications, pp. 349–352, (2006).
18. F. Dandashi,: A Method for Assessing the Reusability of Object-Oriented Code Using a Validated Set of Automated Measurements, (2002).
19. V. P. Venkatesan and M. Krishnamoorthy: A Metrics Suite for Measuring Software Components, J. Converg. Information Technology, Vol. 4, (2009).
20. P. Sandhu and H. Singh: A Neuro-Fuzzy Based Software Reusability Evaluation System with Optimized Rule Selection, International Conference on Emerging Technologies (ICET'06), pp. 664–669, (2006).
21. P. S. Sandhu and H. Singh: A Reusability Evaluation Model for OO-Based Software Components, International Journal Computer Science, Vol. 1, pp. 259–264, (2006).
22. N. W. Nerurkar, A. Sharma, and S. Sagar: A Soft Computing Based Approach To Estimate Reusability of Software Components, ACM SIGSOFT Software Engineering Notes, Vol. 35, (2010).
23. S. I. Zahara, M. Ilias, and T. Zia: A Study of Comparative Analysis of Regression Algorithms for Reusability Evaluation of Object Oriented Based Software Components, International Conference on Open Source Systems and Technologies (ICOSST'13), Pakistan, (2013).
24. L. H. Etzkorn, W. E. Hughes, and C. G. Davis: Automated reusability quality analysis of OO legacy software, Information Software Technology, Vol. 43, No. 5, pp. 295–308, (2001).
25. P. Sandhu and H. Singh: Automatic Reusability Appraisal of Software Components Using Neuro-Fuzzy Approach, International Journal Information Technology, Vol. 1, No. 8, pp. 2407–2413, (2007).
26. B. Ko and J. Park,: Component Architecture Redesigning Approach Using Component Metrics” in Artificial Intelligence and Simulation, Heidelberg: Springer Berlin Heidelberg Vol. 3397, pp. 449–459, (2005).

27. E. S. Cho, M. S. Kim, and S. D. Kim: Component Metrics to Measure Component Quality, Eighth Asia-Pacific Software Engineering Conference, Macau, China, pp. 419–426,(2001).
28. S. Bhattacharya and D.A. Perry: Contextual Reusability Metrics for Event-Based Architectures, International Symposium on Empirical Software Engineering, Australia, (2005).
29. M. Choi and S. Lee: A Coupling Metric Applying the Characteristics of Components, Computational Science and Its Applications, Vol. 3983, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 966–975, (2006).
30. A. P. Singh, P. Tomar: Estimation of Component Reusability through Reusability Metrics, World Academy of Science Engineering and Technology International Journal of Computer, Electrical Automation Control and Information Engineering, Vol:8, No: 11, pp: 2003- 2010, (2014).
31. Hyun Jung La, Jin Sun Her, and Soo Dong Kim: Framework for evaluating reusability of Component-as-a-Service (CaaS), ICSE Workshop on Principles of Engineering Service- Oriented Systems (PESOS), San Francisco, USA, pp. 41–44, (2013).
32. S. Manhas, P.S. Sandhu, V. Chopra, and N. Neeru: Identification of reusable software modules in function oriented software systems using neural network based technique, World Acad. Sci. Eng. Technol., Vol. 43, pp. 823–827, (2010).
33. M.A.S. Boxall and S. Araban, Interface Metrics for Reusability Analysis of Components, Australian SE Conference Proceedings., Melbourne, Australia, pp. 40–51, (2004).
34. G. Gui and P. D. Scott: Measuring Software Component Reusability By Coupling And Cohesion Metrics, J. Comput., Vol: 4, No: 9, (2009).
35. P. S. Sandhu, H. Kaur, and A Singh: Modeling of Reusability of Object Oriented Software System, World Acad. Sci. Eng. Technol., Vol. 56, pp. 162–165, (2009).
36. Li Yingmei, Shao Jingbo, and Xia Weining,:On Reusability Metric Model for Software Component” in Software Engineering and Knowledge Engineering: Theory and Practice, Vol. 114, Springer Berlin Heidelberg, , pp. 865–870, (2012).
37. Claudio Sant’Anna, Alessandro F. Garcia, Christina Chavez, Carlos Lucena, and Arndt Staa,: On the Reuse and Maintenance of Aspect- Oriented Software: An Assessment Framework, in Proceedings of Brazilian Symposium on Software Engineering, (2003).
38. Varun Gupta and Jitender Kumar Chhabra: Package Level Cohesion Measurement in Object- Oriented Software, J. Braz. Comput. Soc., Vol. 18, No. 3, pp. 251–266, Sep. (2012).
39. A. Shri, P . S. Sandhu, V . Gupta, and S. Anand: Prediction of Reusability of Object Oriented Software Systems Using Clustering Approach, World Acad. Sci. Eng. Technol., Vol. 43, pp. 853–856, (2010).
40. M.W. Price, D.M. Needham, and S.A. Demurjian,: Producing Reusable Object-Oriented Components: A Domain-And-Organization- Specific Perspective, Symposium on Software Reusability, Toronto, Canada, pp. 41–50, (2001).
41. Qi Wang: Research on the Quantifying and Calculating Model of the Software Component Reusability, Future Control and Automation Heidelberg: Springer Berlin Heidelberg, pp. 103–109, (2012).
42. O.P. Rotaru and M Dobre,: Reusability Metrics for Software Components, the 3rd ACS/IEEE International Conference on Computer Systems and Applications, Cairo, Egypt, (2005).
43. Seungwon Lee and Ho-Jin Choi: Software Component Reusability Measure In Component Grid, 11th International Conference on Advanced Communication Technology, PyeongChang, South Korea, Vol. 1, pp. 576–578, (2009).
44. J. Al Dallal,: Software Similarity-Based Functional Cohesion Metric, IET Softw., Vol. 3, No. 1, pp. 46–57, (2009).
45. D. Hristov, O. Hummel, M. Huq, W. Janjic,: Structuring Software Reusability Metrics for Component-Based Software Development, ICSEA, pp: 422-429, (2012).
46. Juan Wang and You-An Wang: Teaching software reuse with JavaBeans, 30th Annual Frontiers in Education Conference, Kansas City, USA, Vol. 1, pp. 7–8, (2000).
47. T. Zhang, L. Deng, J. Wu, Q. Zhou, C. Ma: Some Metrics for Assessing Quality of Product Line Architecture, International Conference on Computer Science and Software Engineering, (2008).
48. J. S. Her, J. H. Kim, S. H. Oh, S.Y. Rhew, S.D. Kim: A Framework for Evaluating Reusability of Core Asset in Product Line Engineering, Information and Software Technology, Science Direct, Vol: 49, pp:740-760, (2007).