

OSGi Çerçevesinde ve Ürün Hattı Yaklaşımı ile Geliştirilmiş Bir Radar Kullanıcı Arayüzü Yazılımının Özellik Ağacı İle Yeniden Kullanılabilirliğinin Arttırılması

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. Yazılım kalitesini artırma, süreç riskini düşürme ve kaynakları verimli kullanma ihtiyaçları, yazılım bileşenlerinin veya yazılım ürün hatlarının yeniden kullanımını beraberinde getirmektedir. Yeniden kullanım birçok yazılım geliştirme yaklaşımında uygulansa da Yazılım Ürün Hattı Geliştirme yaklaşımının temelini oluşturmaktadır. Literatürde yeniden kullanılabilirliği değerlendirmek için farklı yöntem ve metrikler sunulmuştur. Daha önce yaptığımız sistematik araştırma sonuçlarına göre, bileşen tabanlı ve ürün hattı yaklaşımları ile geliştirilen yazılımlarda, yeniden kullanılabilirliği ölçmek için en çok kullanılan yazılım faktörünün bağlaşım (İng. Coupling) olduğu görülmüştü. Bu bildiride, modüler yazılım geliştirme için servis platformu sağlayan OSGi çerçevesi kullanılarak geliştirilmiş ve yazılım ürün hattı yaklaşımına dayalı mimarisi olan bir radar kullanıcı arayüzü yazılımının, yeniden kullanılabilirliğinin arttırılması hedeflenmiştir. Yapılan eylem araştırmasında (İng. Action Research) bağlaşım faktörü giriş yelpazesi (İng. Fan-in) ve çıkış yelpazesi (İng. Fan-out) metrikleri ile sayısallaştırılmış ve yazılımın yeniden kullanılabilirliği bileşen seviyesinde (İng. Bundle), bağlaşım faktörüne göre değerlendirilmiştir. Eylem araştırması üç araştırma sorusu ile adreslenmiştir. İlk araştırma sorusu kapsamında, mevcut yazılımın bağlaşımı ölçülmüştür. İkinci araştırma sorusunda ise özellik ağacı (İng. Feature Tree) ile bileşenlerdeki ortak ve projeye özel yetenekler çıkarılarak yazılım yeniden yapılandırıldıktan sonra yazılımın bağlaşımı ölçülmüştür. Araştırma sorusu 1 ve 2'nin sonuçları karşılaştırıldığında, yeniden yapılandırma sonrasında, yazılımın yeniden kullanılabilirliğinin arttığı görülmüştür.

Anahtar Kelimeler: Yeniden Kullanılabilirlik, Yazılım Metrikleri, Bağlaşım, Bileşen Tabanlı Yazılımlar, Ürün Hattı, Özellik Ağacı, OSGi, Giriş Yelpazesi, Çıkış Yelpazesi

Improving the Reusability of an OSGi and Product Line Approach Based Radar User Interface Software by Feature Tree Method

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. The reuse of software components and software product lines is required for improving the quality of software, lowering the risks of process, and for the efficient use of resources. Even though many software development approaches have been applying reuse, software product line approach is based on reusability. Different types of methods and metrics have been proposed in the literature for evaluating the level of reusability. The results of our previous systematic research study showed that coupling is the mostly used software factor for measuring the reusability of the software that has been developed by using component oriented and software product line approaches. This study aims at improving the reusability of a radar user interface software, which has an architecture based on a software product line approach, and which has been developed by using a service based platform for modular software, namely the OSGi Framework. We have performed action research to evaluate the reusability of software at component level and according to the coupling measures based on Fan-in and Fan-out metrics. Action research has mapped three research questions. In the first research question, we have measured the coupling of the original software. In the second research question, we have measured the coupling of the software after the refactoring based on the common and project specific features in the Feature Tree. The comparison of the results of research question 1 and 2, showed an improvement in the reusability of the software after the refactoring.

Keywords: Reusability, Software Metrics, Coupling, Component Based Software, Software Product Line, Feature Tree, OSGi, Fan-In, Fan-Out

1 Giriş

Yazılım Mühendisliği araştırma çalışmaları, yazılım kalitesinin gelişmesine, geliştirme sürecinin riskinin azalıp verimliliğinin artmasına ve zaman yönetiminin desteklenmesine odaklanmaktadır. Var olan yazılımın yeniden kullanımı geliştirme maliyetini düşürürken güvenilir yazılım üretilmesini sağlamaktadır [1].

Literatürde yazılımın yeniden kullanımı ve yeniden kullanılabilirliği farklı kavramlara karşılık gelmektedir. Yeniden kullanım, Frakes ve Succi'ye göre "var olan yazılım ve yazılım ürünlerinin yeni yazılım geliştirmede direkt kullanımı" olarak tanımlanırken aynı araştırmacılara göre yeniden kullanılabilirlik yeniden kullanım olasılığını gösteren yazılım özelliği olarak tanımlanmaktadır [2].

Yazılımların yeniden kullanılabilirliğini sağlayabilmek ve karmaşıklığını yönetebilmek için genel olarak uygulanan yöntem, yazılımı işlevsel olarak sorumlulukları belirlenmiş birimlere ayırmaktadır [3]. Hızlı ve sürekli teslimatı gerektiren yazılım ihtiyaçları, yazılım bileşenlerinin ve ürün hattı yaklaşımlarının kullanımlarını beraberinde getirmiştir. 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 tasarlamının altını çizmektedir. Yazılım ürün hattı ise belirli bir görevin ihtiyacını karşılamak üzere, ortak yetenek kümelerinden oluşan sistemlerin kullanımını esas almaktadır [4].

Bu çalışmanın amacı, bileşen tabanlı ve ürün hattı yaklaşımına dayalı olan, radar uygulama alanında geliştirilen kullanıcı arayüzü yazılımındaki bileşenlerin yeniden kullanılabilirliğini arttırmaktır. Ürün hattımızdaki bileşenlerin yeniden kullanılabilirliğini değerlendirip arttırmaya yönelik çalışma yapılmıştır. 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 hedeflenmiştir.

Yazılımın geliştirilmesinde, modülerlik için servis platformu sağlayan OSGi çerçevesi ve radar alanına özel yetenek setlerini bir araya getiren yazılım ürün hattı yaklaşımı kullanılmıştır. Yapılan eylem araştırmasında bağlaşım faktörü giriş yelpazesi (İng. Fan-in) ve çıkış yelpazesi (İng. Fan-out) metrikleri ile sayısallaştırılmış ve yazılımın yeni-den kullanılabilirliği bileşen seviyesinde bağlaşım faktörüne göre değerlendirilmiştir. Eylem araştırmasını adresleyen ilk araştırma sorusu kapsamında, mevcut yazılımın bağlaşımı ölçülmüştür. İkinci araştırma sorusu kapsamında ise özellik ağacı (İng. Feature Tree) ile bileşenlerdeki ortak ve projeye özel yetenekler çıkarılarak yazılım yeni-den yapılandırıldıktan sonra yazılımın bağlaşımı ölçülmüştür.

Bildiri'nin 2. Bölümü'nde; OSGi, yazılım ürün hattı gibi temel kavramlar ve yeniden kullanılabilirliği arttırmaya ilişkin literatür çalışmaları aktarılmakta, 3. Bölümü'nde çalışmada uygulanan metod ve adımları açıklanmakta, 4. Bölümü'nde yapılan eylem araştırması ve eylem araştırmasını adresleyen 3 araştırma sorusunun sonucunun karşılaştırılması anlatılmakta, 5. Bölümü'nde ise çalışmanın sonuçlarına yer verilmektedir.

2 İlişkili Kavramlar ve Çalışmalar

2.1. OSGi Çerçevesi

Yazılımda Java için bir bileşen sistemi olan Equinox tabanlı OSGi teknolojisinden faydalanılmıştır. OSGi çerçeve yaklaşımı ile Java'daki iş bölümü, soyutlama, bakım, onarım ve yeniden kullanılabilirlik gibi problemlere çözüm getirilmiştir. Modülün mantıksal amaca bağlı kalması ve bileşenin etkileşimde olduğu diğer bileşenlerin gerçekleştirim detaylarına bağlı olmaması sağlanmaktadır. OSGi çerçevesi kullanılarak geliştirilen yazılımlar, bağımlılıkları gösteren meta-data (MANIFEST-MF) bilgisine, bileşenler arası bilgi saklama mekanizmasına doğrudan sahip olmaktadır. OSGi çerçevesi, bileşen yapısının oluşturulması ve bu bileşenlerin çalışma zamanında birbiri ile etkileşimde bulunabilmesi için yöntem sunmaktadır. Java'da

birçok sorunun kaynağı global ve tek olan classpath'e çözüm olarak OSGi her bileşen için ayrı classpath sunmaktadır. Paylaşılan sınıflar için ise OSGi, paketlerin bileşenler arasında nasıl paylaşılacağına dair dışardan alma (İng. import) ve dışarı aktarma (İng. export) mekanizması kullanmaktadır [5].

OSGi'de bileşen, demet (İng. Bundle) olarak da adlandırılmaktadır. Yığın bir jar dosyası olup bu jar dosyasına dışardan alınan ve dışarıya aktarılan paketlere dair bilgileri içeren meta-data dosyası da eklenerek bir yığın haline getirilmektedir. Jar dosyaları olarak paketlenmiş yığınlar, OSGi'nin sağladığı servis arayüzleri aracılığı ile birbirleri ile haberleşmektedir. OSGi çerçevesi kullanarak geliştirilen yığınlar, çalışma zamanı (İng. Runtime) sırasında dinamik olarak keşfettikleri diğer yığınlar tarafından sunulan servisleri kullanabilmektedir [5].

OSGi'de her yığın ve paket versiyonlanabilmekte, yığınlar arasındaki kullanımlara versiyonlara göre kısıtlamalar tanımlanabilmektedir. Sonuç olarak, OSGi teknolojisi ile dinamik bir modül sistemi oluşturulmakta, Java'daki classpath, versiyonlama ve Jar problemine çözüm getirilmektedir. OSGi teknolojisi, modülerliği arttırdığı için bağlantıyı düşük ve yeniden kullanılabilirliği yüksek bileşenler geliştirilmesine yönelik bir altyapı sağlamaktadır [5].

2.2. Ürün Hattı Yaklaşımı

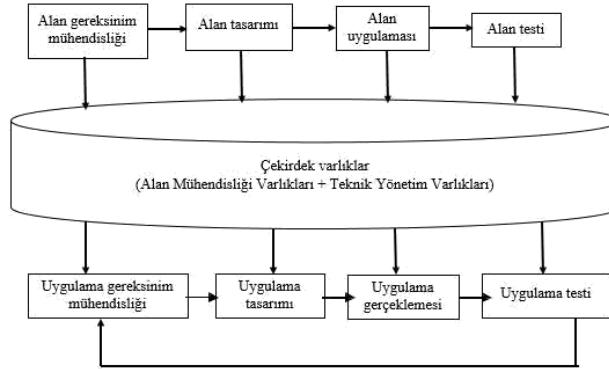
Eylem araştırmasının yapıldığı radar kullanıcı arayüzü yazılımının geliştiriminde faydalanılan bir diğer yaklaşım Yazılım Ürün Hattı'dır. Yazılım Ürün Hattı Mühendisliği (İng. Software Product Line Engineering) gerçekleşirken tüm ürün hattı için değişken ve ortak olan yetenekler bir araya getirilerek yeniden kullanımın artırılması amaçlanmaktadır.

Yazılım Ürün Hattı, "özgün bir pazarın özel ihtiyaçlarını karşılamak üzere, ortak ve değişken yönetilebilir özellik gruplarını destekleyen yazılım yoğun sistemler" olarak tanımlanmaktadır [6]. Şekil 1'de, yazılım ürün hattının döngüsel süreci sunulmaktadır [6].

Yazılım Ürün Hattı yaklaşımı, 1970'lerde Ürün Ailesi kavramı ile ortaya çıkmaya başlamış 1990'larda ise tamamen tanımlanıp kullanıma geçirilmiş bir yeniden kullanım yaklaşımıdır. Maliyeti ve proje geliştirim süresini düşürürken yazılım kalitesini arttırmayı hedeflemektedir. Olgunlaşmış ve doğrulanmış yetenekler ile daha kaliteli yazılımlar ortaya çıkarılmaktadır. Yaklaşım, yazılım geliştirme süreci boyunca büyük ölçekli yeniden kullanıma dayanmaktadır [6].

Yazılım ürün hattı yaklaşımı, yeniden kullanım için geliştirme ve yeniden kullanımla geliştirme kavramları arasındaki farklılığı temel almaktadır. Yeniden kullanım için yazılım geliştirme, Şekil 1'de üst kısımda akışı verilen Alan Mühendisliği paydaşına karşılık gelmektedir. Alan mühendisliği ortak yeteneklerin farklı sistemlerde yeniden kullanımı sürecini kapsamaktadır. Uygulama mühendisliği ise ürün hattında yeniden kullanım ile yazılım geliştirme paydaşına karşılık gelmektedir. Değişken yeteneklerin yönetimleri ve geliştirilmesi bu aşamada gerçekleşmektedir [6]. Geliştirilen yazılımlarda ortak varlıkların doğrudan ya da değiştirilerek kullanılması söz konusu olduğu için yeniden kullanılabilirlik üst seviyeye çıkarılmaktadır [7].

Özellik Ağacı, Yazılım Ürün Hattı yaklaşımının dayandırıldığı özellik modeli olup nihai ürünün karakteristik özelliklerini göstermektedir. Model, hiyerarşik olarak sunulmaktadır. Özellikler; zorunlu, opsiyonel, veyalanmış (İng. ORed) ya da harici veyalanmış (İng. XOR) olarak çeşitlenmektedirler. Çoğu özellik modeli, farklı özellik tiplerinin kombinasyonları sonucunda oluşturulmaktadır [8].



Şekil 1. Yazılım Ürün Hattı Süreci [6]

Radar ihtiyacına yönelik özel ihtiyaçları karşılamak üzere geliştirdiğimiz radar kullanıcı arayüzü yazılımlarımızda, ortak ve yönetilen bir özellik grubunu destekleyen çekirdek bir varlık kümesinin kullanımını esas alan ürün hattı yaklaşımı kullanılmıştır. Radar uygulama alanında geliştirilen, operatörün radar ile etkileşimini ve kontrol etmesini sağlayan radar kullanıcı arayüzü yazılımları, radar sisteminden elde edilen tespit ve takip verilerinin sunulmasını ve radarın kontrol edilmesini sağlamaktadır. Gerçek zamanlı çalışan radar sisteminde, aynı anda farklı arayüzlere sahip olunması bu yazılımların karmaşıklığını arttırmaktadır. ASELSAN Radar ürün ailesinde yer alan farklı Radar sistemlerinin kullanıcı arayüzü yazılımlarının yeniden kullanılabilirliği, yazılım geliştirme sürecinin kısaltılarak eforun düşürülmesi ve riskin minimize edilmesi için önemli bir kalite gereksinimi olarak görülmektedir [3]. Şekil 1’de sunulan alan gereksinim mühendisliği sonucu Yapıtışı, uygulama gereksinim mühendisliği sonucu Projeye Özel olarak tanımlanan bileşenlerimiz geliştirilmiştir.

2.3. Yeniden Kullanılabilirlik Faktör ve Metrikleri

Yazılımların karmaşıklığı ve boyutu sebebi ile yazılımı işlevsel sorumluklara göre bileşenlere ayırmak önem taşımaktadır. Önemli bir tasarım prensibi, bileşenlerin kendi içinde yüksek uyumlu (İng. High Cohesion) kendi aralarında düşük bağlaşımlı (İng. Low Coupling) olarak geliştirilmesi görülmektedir. Bileşenler arasındaki bağlaşımın yüksek olması o birimin tek başına kullanımını zorlaştırmakta, yeniden kullanılabilirlik faktörünü düşürmektedir [9,10].

Yeniden kullanılabilirlik faktörleri için literatür taraması yapılırken eksiksiz ve tarafsız bir makale havuzu oluşturmak için Sistematik Haritalama metodundan yararlanılmıştır [11]. Araştırma, sırası ile hedef, soru ve faktörlerin belirlenmesine dayalı olarak yapılandırılmıştır [12]. Literatürde modüler seviyede yeniden kullanılabilirlik ile yapılan çalışmalar seçilirken; araştırma sorusu, seçim kriterleri, veri çıkarma stratejisi ve çalışmanın kalite değerlendirme kriterleri belirlenmiştir. Sistematik Haritalama çalışmasında frekansı en yüksek olan faktörü ölçmek için metrik seçilmiştir. Seçilen metrikler kullanılarak mevcut yazılımın ölçümü alınmıştır.

“Bileşen-tabanlı ve ürün-hattı yazılım geliştirme yaklaşımlarında yeniden kullanılabilirliği ölçmek için yapılan çalışmalarda, yeniden kullanılabilirliği etkileyen faktörler nelerdir?” sorusundan yola çıkarak gerçekleştirilen haritalama çalışmasının sonucuna göre, bileşen tabanlı yazılımlarda yeniden kullanılabilirliği en çok etkileyen ilk üç faktör ile bunların çalışmalara göre dağılım frekansları Tablo 1’de sunulmuştur:

Tablo 1. Faktörlerin Frekansı ve İlişkili Çalışmalar

Faktör	Makale Sayısı
<i>Bağlaşım (İng. Coupling)</i>	17
<i>Uyum (İng. Cohesion)</i>	11
<i>Karmaşıklık (İng. Complexity)</i>	7

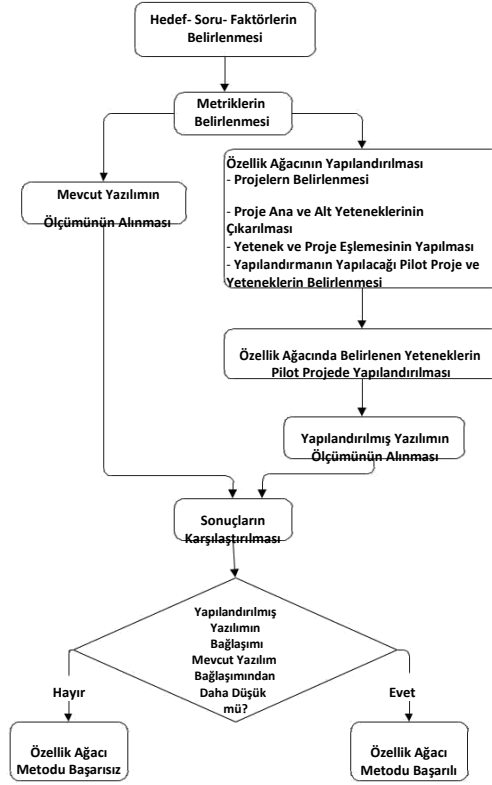
Yüksek seviyede (İng. High level) OSGi çerçevesindeki bileşenlerin yeniden kullanılabilirliğini ölçmek için Bağlaşım (İng. Coupling) faktöründen faydalanılmıştır. Bağlaşım faktörü literatürdeki genel geçer giriş yelpazesi (İng. Fan-in) ve çıkış yelpazesi (İng. Fan-out) metrikleri ile sayısallaştırılmıştır [13, 14, 15].

3 Araştırmanın Amacı ve Yöntemi

Çalışmada uygulanan metodun süreç şeması Şekil 2’de verilmektedir. Yeniden kullanılabilirliğin artırılması motivasyonu ile eylem araştırması yapılmıştır. Eylem araştırmasını adresleyen 3 araştırma sorusu geliştirilmiştir. Geliştirilen araştırma sorularından ilki mevcut durumun yeniden kullanılabilirliğini, ikincisi yeniden yapılandırılan kodun yeniden kullanılabilirliğini değerlendirmeyi amaçlamaktadır. Üçüncü araştırma sorusunda ise ilk iki araştırma sorusunun sonuçlarını karşılaştırarak metodun doğrulanmasını hedeflemektedir.

OSGi alt yapısını kullanan projelerden ve bu projelerin ana-alt yetenek setlerinden Özellik Ağacı oluşturulmuştur. Bu çalışmada, Yazılım Gereksinim Özelliği (İng. Software Requirement Specification) dokümanından özelliklerin tanımlanıp çıkarılması süreci ile model oluşturulmuştur. Çıkarılan özellik ağacı, iki geliştirici tarafından karşılıklı olarak gözden geçirilmiştir.

Yetenek ve projeler eşleştirilmiştir. Özellik Ağacı modelinden, yapılandırılacak yetenek ve pilot proje seçimi yapılmıştır. Kodu yapılandırmak için Parçala-Yönet (İng. Divide and Conquer) metodundan faydalanılmıştır. Mevcut yazılım yapılandırıldıktan sonra belirlenen metrik ölçüm değerleri alınmıştır. Yapılandırılmış ve yapılandırılmamış yazılımların ölçüm sonuçları karşılaştırılarak Özellik Ağacı ile yeniden kullanılabilirliğin artıp artmadığına karar verilip metodun doğrulanmıştır.



Şekil 2. Uygulanan Metodun Süreç Şeması

Eylem araştırmasını adreslemek için geliştirilen 3 araştırma sorusu Tablo 2 'de sunulmaktadır. 4.1., 4.2. ve 4.3. bu araştırma sorularına karşılık gelen yanıtların bulunması hedeflenmiştir.

Tablo 2. Eylem Araştırmasını Adresleyen Araştırma Soruları

	Araştırma Soruları
Araştırma Sorusu 1	AS1: Mevcut yazılımın bağlaşım analizi sonucundaki giriş - çıkış yelpaze metrik değeri nedir?
Araştırma Sorusu 2	AS2: Özellik ağacına göre yeniden yapılandırılan yazılımın bağlaşım analizi sonucundaki giriş - çıkış yelpaze metrik değeri nedir?
Araştırma Sorusu 3	AS3: Karşılaştırılan araştırma sorusu 1 ve 2'nin sonucuna göre yeniden kullanılabilirlik özelliği nasıl değişim gösterir?

Ölçüm Java programlama dilini destekleyen statik kod analiz aracı olan SonarGraph Explorer aracı ile çevrim dışı (İng. offline) olarak alınmıştır. Lisanssız olan bu araç, bileşenler arasındaki bağımlılıkların görsel sunumunu sağlamaktadır.

4 Eylem Araştırması

4.1. Araştırma Sorusu 1 – Mevcut Yazılım Bağlaşım Analizi

Eylem araştırmasını adresleyen araştırma sorusu 1 kapsamında , Tablo 2’deki araştırma sorularından ASI’in yanıtlanması için mevcut yazılımın bağlaşım metrikleri ölçülüp mevcut durumun analizinin yapılması hedeflenmiştir.

SonarGraph aracı kullanılarak yazılımı oluşturan bileşenlerin bağlaşım değerlerine bakılarak mevcut durum değerlendirilmesi yapılmıştır. 5 Radar Kullanıcı Arayüzü Yazılım projemiz için özellik ağacı çıkarılmıştır. 5 Radar Kullanıcı Arayüzü Yazılım projemizden biri pilot proje olarak seçilmiştir. SonarGraph aracının sağladığı yazılımın genel metrik değerleri, proje büyüklüğünün anlaşılması açısından Tablo 3’te sunulmaktadır:

Tablo 3. Mevcut Projenin Genel Metrik Değerleri

	Büyüklik
Toplam Satır Sayısı (İng. Total Lines of Code):	234.052
Sınıf Sayısı (İng.Number of Class):	1.468
Java Paket Sayısı (İng. Number of Java Package):	240

Ürün hattı yaklaşımı uygulanarak geliştirilen yazılımlarımızda, ortak kullanılan yeneleklerin oluşturduğu bileşenler YAPITAŞI olarak adlandırılmaktadır. YAPITAŞI’ları ürün hattında, alan mühendisliğinin çıktısı olarak düşünülmüştür. Projeden projeye değişkenlik gösteren bileşenler ise uygulama mühendisliği adı altında geliştirilmiştir.

Tablo 4’te PROJE1’deki bileşenlerin giriş ve çıkış yelpaze metrik değerlerini sunmaktadır.

Tablo 4. PROJE1’deki Bileşenlerin Tipi ve Giriş Çıkış Yelpaze Metrik Değerleri

Bileşen	Alan (A) / Uygulama(U) Mühendisliği	Bileşenin Bağımlılık Kurduğu Bileşen Sayısı (Çıkış Yelpazesi)	Bileşene Bağımlılık Kuran Bileşen Sayısı (Giriş Yelpazesi)
ALARMVERICI	U	7	1
ANAPENCERE	A	8	5
ARAYUZDESTEK	A	2	7
CITSUNUM	A	8	1
CRAMKKM	U	6	2
CRAMKKM Sim	U	4	1
DESTEK	A+U	0	25
DESTEK2	U	3	2
DURUMMODEL	A	3	4
ERISIMKONTROLCU	A	2	2
EUA	U	1	6

EUG	U	7	1
HARITASUNUM	A	6	1
KKSARAYUZ	U	5	2
KKSARAYUZ Sim	U	4	1
KONTROLMODEL	A	3	13
OLAYVERIKUTUPHANESI	A+U	2	19
OLAYYONETICI	A+U	2	1
OLAYYONETICI ARAYUZ	A+U	1	19
RADARKABUK	U	1	1
RAPORMODEL	A	3	9
RAPORSUNUM	U	8	9
RSYARAYUZ	U	7	2
RSYARAYUZ Sim	U	4	1
SEMBOLOJI	A	4	1
SKOPSUNUM	A	2	3
SKOPSUNUM ARAYUZ	A	2	3
YARDIM	A	3	2

Tablo 4’te verilen “Bileşenin Bağımlılık Kurduğu Bileşen Sayısı” çıkış yelpazesi, “Bileşene Bağımlılık Kuran Bileşen Sayısı” ise giriş yelpazesi metrik değerlerine karşılık gelmektedir. Tabloda “Alan Uygulama Mühendisliği” sütununda “A+U” ile gösterilen bileşenler, yapılandırma yapılacak bileşen karşılaştırılmasında hariç tutulacaktır. Bu bileşenler, radar alanı dışında Elektronik Harp gibi başka alanlarda da kullanılmakta ve kesişim alanı olarak tanımlanmaktadır. “Alan Uygulama Mühendisliği” sütununda sadece “A” veya “U” olan bileşenler ise radar alanına özgü olarak geliştirilmiş olup, “A” ürün ailesi içinde yer alan 5 radar projesinde de ortak kullanımı olan “U” tek bir projeye özgü olan bileşen olma özelliği taşımaktadır.

Kesişim alanındaki bileşenler hariç tutulduğunda Tablo 4’deki sonuçlara göre giriş ve çıkış yelpaze metrik değerleri en yüksek olan bileşen KONTROLMODEL bileşeni olarak görülmektedir. Eylem araştırmasını adresleyen ilk araştırma sorusunun sonucunda mevcut durum analizi yapılarak en yüksek bağlaşım değerli bileşen saptanmıştır.

4.2 Araştırma Sorusu 2- Yeniden Yapılandırılan Yazılımın Bağlaşım Analizi

Eylem araştırmasını adresleyen ikinci araştırma sorusu kapsamında ise özellik ağacı (İng. Feature Tree) ile bileşenlerdeki ortak ve projeye özel yetenekler çıkarılarak yazılım yeniden yapılandırıldıktan sonra yazılımın bağlaşım ölçülmüştür. Yeniden yapılandırma ile yazılımın mimarisi değiştirilmiştir. Tablo 2’deki araştırma sorularından AS2’nin yanıtına karşılık gelmektedir.

Özellik ağacı, OSGi çerçevesinde geliştirilmiş ortak yapı taşlarından ve projeye özel bileşenlerden ürün hattı yaklaşımı ile oluşturulmuş PROJE1, PROJE2, PROJE3, PROJE4 ve PROJE5 projeleri için çıkarılmıştır. Tablo 5’te çıkarılan özellik ağacının örnek bir kesiti sunulmaktadır. Yetenekler sütunu YGÖ (Yazılım Gereksinim Özellikleri - İng. SRS) dokümanı esas alınarak çıkarılmıştır. Özellik ağacı, 46 ana yetenekten,

188 alt yetenekten oluşturulmuştur. Projelerin altındaki yetenek hücreleri her bir radar kullanıcı ara yüzü yazılımı için YGÖ dokümanı temel alınarak doldurulmuştur. ‘RADAR DURUMU SUNUM YETENEKLERİ’, yazılımlardaki bağlantı durumu sunum ve sıcaklık sunum yeteneklerini içeren kök yetenek olarak tabloya eklenmiştir. Bağlantı ve sıcaklık sunum alt yetenekleri için projelerin altındaki sütun hücrelerinin işaretlenmiş olması her iki yeteneğinde 5 proje tarafından kullanıldığını belirtmektedir. Metro Raporu Yeteneği ise sadece PROJE1’de mevcut bir yetenek olarak tanımlanmaktadır.

Tablo 5. Yetenek Ağacından Örnek Bir Kesit

Yetenekler	Açıklama	PROJE 1	PROJE 2	PROJE 3	PROJE 4	PROJE 5
<i>RADAR DURUMU YETENEK LERİ</i>						
Bağlantı Durum Sunum Yeteneği	Bağlantılar listesi, devam eden ve henüz başlamamış olan bağlantılar	x	x	x	x	x
Sıcaklık Sunum Yeteneği	Cihaz birim sıcaklıklarının derece olarak sunumu	x	x	x	x	x
<i>MERMİ RAPORU YETENEĞİ</i>						
Mermi Raporunun Skoplarda Sunulması		x				
Mermi Raporunun Haritada Sunulması		x				
<i>METRO RAPORU YETENEĞİ</i>						
Metro Raporu Sunum		x				
Metro Raporu’nun KKS’den Alınması		x				

Tablo 6’da sunulan “MERMİ RAPORU YETENEĞİ” ve “METRO RAPORU YETENEĞİ” sadece pilot proje olarak seçilen PROJE1 KGÜ YKB’de (Kontrol Görüntü Ünitesi- Yazılım Konfigürasyon Birimi) kullanıp diğer projelerde kullanılmayan

ancak alan mühendisliği yaklaşımı ile geliştirilen KONTROLMODEL bileşeninde bulunan yeteneklerdir. Bu yetenekler, yapılandırma ile sadece PROJE1 projesine özel yetenekler olacak şekilde ayrı bileşenler haline getirilmiştir. Yani “METRORAPORU” ve “MERMIRAPORU” bileşenlerine ayrılmışlardır.

Tablo 6. Yapılandırma Sonrasında PROJE1’deki Bileşenlerin Tipi ve Giriş Çıkış Yelpaze Metrik Değerleri

Bileşen	Alan (A) / Uygulama(U) Mühendisliği	Bileşenin Bağımlılık Kur- duğuBileşen Sayısı(Çıkış Yelpazesi)	Bileşene Bağımlılık Ku- ran Bileşen Sa- yısı (Giriş Yel- pazesi)
KONTROLMODEL	A	3	11
MERMIRAPORU	U	1	1
METRORAPORU	U	1	1

4.3. Araştırma Sorusu 3 – Yeniden Kullanılabilirlik Bağlaşım Analizi Sonuçlarının Karşılaştırılması

Eylem araştırmasını adresleyen üçüncü araştırma sorusu Tablo 2’deki AS3’ün yanıtına karşılık gelmektedir. Araştırma sorusu 1 ve 2’nin sonuçlarındaki giriş ve çıkış yelpaze metrik değerlerine bakılarak yapılandırma öncesi ve sonrası yazılımın yeniden kullanılabilirliği bağlaşım analizi ile karşılaştırılmıştır. Karşılaştırma sonucunda, yapılandırma sonrasındaki bağlaşım faktörünün düştüğü buna bağlı olarak yeniden kullanılabilirlik özelliğinin arttığı görülmüştür.

5 Sonuçlar

Bu çalışma, modüler yazılım geliştirme için servis platformu sağlayan OSGi çerçevesi kullanılarak geliştirilmiş ve yazılım ürün hattı yaklaşımına dayalı mimarisi olan bir radar kullanıcı arayüzü yazılımının, yeniden kullanılabilirliğinin artırılması hedeflenmiştir. Yeniden kullanılabilirliği arttırmaya yönelik olarak Özellik Ağacı modelinden faydalanılmıştır.

Yapılan eylem araştırma çalışmasında bağlaşım faktörü giriş yelpazesi ve çıkış yelpazesi metrikleri ile sayısallaştırılmış ve yazılımın yeniden kullanılabilirliği bileşen seviyesinde, bağlaşım faktörüne göre değerlendirilmiştir. Eylem araştırmasını adresleyen ilk araştırma sorusu kapsamında, mevcut yazılımın bağlaşımı ölçülmüştür. İkinci araştırma sorusu kapsamında ise özellik ağacı ile bileşenlerdeki ortak ve projeye özel yetenekler çıkarılarak yazılım yeniden yapılandırıldıktan sonra yazılımın bağlaşımı ölçülmüştür.

Analiz edilen özellik ağacına göre yapılan yapılandırma sonucunda, yazılımın çıkış yelpazesinde azalma olmuştur. PROJE1’in KONTROLMODEL YAZIT bileşeninden dışarıya çıkarılıp ayrı bir bileşen haline getirilen MERMIRAPORU ve METRORAPORU bileşenleri PROJE1’e yani projeye özel hale getirilmiştir. Alan mühendisliği uygulanarak geliştirilen KONTROLMODEL’de bulunan MERMIRAPORU

ve METRORAPORU bileşenleri yapılandırma sonrası değişen mimaride uygulama mühendisliğinin ihtiyacını karşılayacak şekilde projeye özel hale getirilmiştir. Böylece bu yetenekler, PROJE2, PROJE3, PROJE4 ve PROJE5 gibi bu yeteneği kullanmayan yazılımların kaynak kodlarından çıkarılmıştır. KONTROLMODEL'e olan bağımlılık düşmüştür. Eylem araştırmaları için türetilen araştırma 1 ve 2'nin sonuçları karşılaştırıldığında, Özellik Ağacı'na göre gerçekleştirilen yeniden yapılandırma sonrasında, yazılımın yeniden kullanılabilirliğinin arttığı görülmüştür.

Gelecek çalışma olarak yapılandırmanın tüm yetenekler için tamamlanıp bağlaşım analizinin yapılması planlanmaktadır.

Teşekkür

Değerli yorumları ile çalışmaya katkıda bulunan Sn. Baki Demirel'e teşekkürlerimizi sunarız.

Referanslar

1. Martin R.C.: Clean Code – A Handbook of Agile Software Craftmanship, Pearson Education, (2009).
2. Frakes W., Succi G.: An Industrial Study of Reuse, Quality and Productivity, Journal of Systems and Software, Vol:57, pp:99-106, (2001).
3. Cilden E., Bayraktar G., Işık G., Demirel B.: OSGİ Çerçevesinde Bir Olay Güdümlü Mimari Uygulaması, UYMS'13, (2013).
4. Cankurtaran E., Cilden E., Tarhan A.: Bileşen Tabanlı ve Ürün Hattı Yazılım Geliştirme
5. Bartlett N., OSGi In Practice, Njbartlett, (2009).
6. Liden F.J., Schmid K., Rommes E.: Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering, Springer –Verlag, (2007).
7. Jones L., Northrop L.: Clearing the Way of for Software Product Line Success, IEEE Software, (2010).
8. Hamza M., Walker s: Recommending Features and Feature Relationships from Requirement Documents for Software Product Lines, IEEE/ACM 4th International Workshop on Realizing AI Synergies in Software Engineering, (2015).
9. Ko B., Park J.: Component Architecture Redesigning Approach Using Component Metrics” in Artificial Intelligence and Simulation, Heidelberg: Springer Berlin Heidelberg Vol. 3397, (2005).
- 10 Choi M., Lee S.: A Coupling Metric Applying the Characteristics of Components, Computational Science and Its Applications, Vol. 3983, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 966–975, (2006).
11. Kitchenham B.: Procedures for Performing Systematic Reviews, Eversleigh, (2004).
12. Basili V. R., Caldiera G., Rombach H.D.: The Goal Question Metric Approach, Encyclopedia of Software Engineering, (1994).
13. Mugia A., Marchesi M., Concas G.: Parameter-Based Refactoring and Relationship with Fan in Fan out Coupling, 2011 Fourth International Conference on Software Testing, Verification and Validation Workshop, (2011).
14. Sarala S., Jabbar A.: Information Flow Metrics and Complexity Measurement, Conference Computer Science and Information Technology, (2010).
15. S. Hamza, S. Sandou, R. Fleurquin: Measuring Qualities for OSGi Component Based Application, 13th International Conference On Quality Software, (2013).