

Android Uygulamalar İin Yazılım Kalite Modeli

Merve Vildan ŐimŐek¹, Aysu Betin-Can² ve Barbaros Can¹

¹ BiliŐim Teknolojileri Grubu (BİLTEG), Teknoloji ve Yenilik Destek Programları BaŐkanlıđı (TEYDEB), TÜBİTAK, Ankara, Trkiye

{vildan.simsek, barboros.can} @tubitak.gov.tr

² Enformatik Enstits, Orta Dođu Teknik niversitesi, Ankara, Trkiye

betincan@metu.edu.tr

Özet. Gnmzde akıllı telefon ve tablet gibi mobil cihazlar giderek yaygınlaŐmaktadır. Bu durum mobil uygulama sayısının nemli lde artıŐını da beraberinde getirmektedir. Bu sistemlerin poplerliđini yakın gelecekte de arttırmaya devam edeceđi tahmin edildiđinden, mobil uygulamaların kalitesi olduka fazla nem kazanmaktadır. Bu alıŐmanın amacı Android uygulamaları iin yazılım kalite modeli oluŐturulmasıdır. Mobil pazardaki yaygınlıklarından dolayı Android İŐletim Sistemi iin geliŐtirilmiŐ uygulamalar hedef olarak seilmiŐtir. alıŐmada ISO/IEC SQuaRE Yazılım Kalite Modeli Standardında tanımlanmıŐ olan geleneksel yazılım kalite zellikleri adapte edilmiŐtir. Standartta tanımlanmıŐ olan yazılım kalite zelliklerinden Android uygulamalara uygulanabilir olanlar seilmiŐ ve Android uygulamalarının kalitesini etkileyen yeni bir zellik eklenmiŐtir. Sonrasında Android uygulamalara zg yeni kaynak kod metrikleri belirlenmiŐtir. Nihai zelliklerin yer aldıđı bir kalite modeli oluŐturulmuŐ ve bu model bir rnek olay incelemesinde Android uygulamalarına uygulanarak, modelin uygulanabilirliđi gsterilmiŐtir.

Anahtar Szckler: Yazılım Kalitesi, Mobil Yazılım, Kalite Modeli, Android Uygulamalar, ISO/IEC SQuaRE Standardı

1 GİRİŞ

Mobil iletişim hayatımızda önemli bir yer kaplamaktadır. Akıllı telefonların ve tabletlerin sayısındaki artış mobil uygulamaların sayısının çarpıcı bir oranda artışını beraberinde getirmektedir. Gartner Grubunun raporuna göre “2017 tarihine kadar 268 milyardan fazla mobil uygulama indirilmiş olacak ve indirilen bu uygulamalar 77 milyar dolardan fazla hasılat oluşturacaktır” [1]. Mobil uygulama sayısı çok yüksek bir hızla arttığından uygulamaların kalitesi önemli bir konu haline gelmektedir. Kalite artık sadece bir BT sorunu değildir, geliştiricilerin ve firmaların itibarını direkt olarak etkilemektedir. Kullanıcılar uygulamalara yorumlar yazmakta ve puanlar vermektedir. Uygulamaların almış olduğu ortalama puan değeri ve kullanıcı yorumları herkesin erişimine açık olduğundan bu değerler potansiyel kullanıcıları da etkilemektedir. Ayrıca uygulama mağazalarında rakip ürün sayısı fazla olduğundan, bir uygulamadan vazgeçmek ve ona alternatif bir uygulama bulmak oldukça kolay hale gelmiştir.

Bu çalışmanın amacı Android uygulamaları için bir kalite modeli oluşturulmaktır. Mobil uygulama pazarında en büyük paya sahip olan işletim sistemi Android İşletim Sistemi olduğundan Android uygulamalar hedef olarak seçilmiştir. İstatistiklere göre Google Play Mağazası’nda indirilmeye hazır uygulama sayısı Haziran 2015 itibarıyla 1,6 milyondur. Apple’ın Uygulama Mağazası 1,5 milyon uygulama barındırdığından ikinci en büyük uygulama mağazasıdır. [2]

Android uygulamaları için kalite modeli geliştirilmesi sürecinde, ISO/IEC SQuaRE Yazılım Kalite Modeli Standardında tanımlanmış olan kalite özellikleri analiz edilmiştir. Çalışma kapsamında ürün kalite modelinde (product quality model) yer alan sekiz özellik göz önüne alınmıştır: işlevsel uygunluk, performans verimliliği, güvenilirlik, güvenlik, kullanılabilirlik, bağdaşırılık, taşınabilirlik ve bakım kolaylığı. Bu sekiz özelliğin alt özellikleri ve kullanımdaki kalite (quality in use) modelindeki özellikler çalışmaya dâhil edilmemiştir. Kalite modelimiz mobil yazılım alanı ile ilişkili olan kalite özellikleri ile ilgilendiğinden, ürün kalite modelinden uygulanabilir kalite özellikleri seçilmiş ve bunlara ek olarak veri bütünlüğü adlı bir özellik eklenmiştir.

Android uygulama geliştiricilerine anket yapılarak, kaynak kod metrikleri ve kalite özellikleri arasındaki ilişki tanımlanmıştır. SQuaRE kalite modeli Android uygulamalarına uygun olması adına adapte edilmiş, oluşturulan kalite modelinin geçerlilik sınaması yapılmış ve model bir örnek olay incelemesinde iki adet açık kaynak kodlu Android uygulamasının farklı dağıtımlarına uygulanarak modelin uygulanabilirliği gösterilmiştir.

Mobil yazılım kalite modellerinin geliştirilmesi üzerine az sayıda çalışma yapılmıştır. İlişkili çalışmaların birçoğu uygulamaların kalitesini ölçme konusunda yetersizdir, çünkü bu çalışmalarda metrik tanımlaması yapılmamıştır. Bizim çalışmamızda ise mobil uygulamaların kalitesinin ölçülebilmesi adına Android’e özgü kaynak kod metrikleri tanımlanmıştır.

Bu makalenin katkıları Android uygulamaları için geçerlilik sınaması yapılmış bir kalite modeli ve bu model için belirlenmiş olan metriklerdir. Mobil platformların kalite değerlendirmesi alanında bir başlangıç çalışması gerçekleştirilmiştir. Çalışma Android’e özgü metrik tanımlamaları içermesi ve güncel kalite standardının (ISO/IEC SQuaRE) kullanılması sebebiyle bu alanda yapılan diğer çalışmalardan ayrılmaktadır.

Makalenin devamı şu şekilde yapılandırılmıştır: Bölüm 2’de Android uygulamaları için oluşturulan kalite modelinin geliştirme aşamaları açıklanmıştır. Bölüm 3’te sunulan modelin değerlendirilmesi ve örnek olay incelemesi sonuçları yer almaktadır. Son bölümde ise sunulan modele ilişkin çalışmanın sonucu ve bu konuda gelecekteki olası çalışmalar açıklanmıştır.

2 ANDROID KALİTE MODELİ GELİŞTİRİLMESİ

2.1 Kalite Özelliklerinin Tanımlanması

Kalite özelliklerinin tanımlanması sürecinde ISO/IEC SQuaRE Kalite Standardı, nesne tabanlı modeller ve Android uygulamalara özgü komponentler göz önünde bulundurulmuştur. Kalite özelliklerinin başlangıç seti olarak ISO/IEC 25010 Kalite Standardından “Performans Verimliliği”, “Taşınabilirlik”, “Bakım Kolaylığı”, “İşlevsel Uygunluk”, “Güvenilirlik”, “Kullanılabilirlik”, “Güvenlik” ve “Bağdaşılık” özellikleri seçilmiştir. Bu özelliklere ek olarak, bu alanda yapılmış olan çalışmaların [3], [4], [5], [6], [7] incelenmesiyle “Veri Bütünlüğü” özelliği seçilmiştir. Oluşturulan Android kalite modeli Şekil 1 ile gösterilmiştir.



Şekil 1. Android Kalite Modeli

2.2 Metrik Tanımlanması

Yazılım sistemlerinin kalitesini değerlendirirken genellikle kaynak kod metrikleri kullanılmaktadır. Bu metrikler kaynak kod hakkında somut bilgi edilmek için objektif bir yol sunmaktadırlar. Kalite standartları ve bu alanda yapılmış ilgili çalışmalar analiz edilerek Android uygulamalarına uygulanabilir olan metrikler elde edilmiştir [9] [32] [33]: Sınıf Sayısı (Number of Classes), Kalıtım Ağacının Derinliği (Depth of Inheritance Tree), Kararsızlık (Instability), Metotlardaki Uyum Eksikliği (Lack of Cohesion of Methods), Metot sayısı (Number of Methods), Nitelik Sayısı (Number of Attributes) ve McCabe Çevrimsel Karmaşıklığı (McCabe Cyclomatic Complexity). Android uygulama geliştirimi java tabanlı ve nesne tabanlı olmasına rağmen bazı kilit farklılıklar bulunmaktadır. Kaynak kod yapılarındaki farklılıklar nedeniyle Android uygulamalara özel metrik tanımlaması yapmaya ihtiyaç duyulmuştur. Android Programlama kitaplarını, öğreticileri ve yayınları kapsamlı şekilde irdeleyerek [8] [9] [10] [11] [12] [13] [14] [15] [16], Android’e özgü sekiz yeni metrik tanımlaması yapılmış-

tır: Aktivite Sayısı (Number of Activities), Servis Sayısı (Number of Services), Yayın Alıcıları Sayısı (Number of Broadcast Receivers), İçerik Sağlayıcıların Sayısı (Number of Content Providers), Minimum SDK Sürümü (Minimum SDK Version), Hedeflenen SDK Sürümü (Target SDK Version), Intent Sayısı (Number of Intents) ve İzin Sayısı (Number of Permission).

Android'e özgü metriklere ek olarak, Android uygulamalarına uygulanabilir olduğunu düşündüğümüz nesne tabanlı üç metrik daha tanımlanmıştır: Dialog Sayısı (Number of Dialogs), İş Parçacığı Sayısı (Number of Threads) ve Veritabanındaki Tabloların Sayısı (Number of Tables in Database). Ortaya çıkan metrik listesi ve bu metriklerin açıklamaları Tablo 1'de yer almaktadır. Tablodaki ilk sütun metriklerin isimlerini, ikinci sütun metriklerin türlerini örneğin nesne tabanlı (tabloda OO olarak ifade edilmiştir) veya Android, son sütun ise metrikler ile ilgili kısa açıklamaları içermektedir. Bu metriklerin tümü doğal sayı değerleri almaktadırlar. Bu metriklerin her biri için yüksek veya düşük olmasının kalite açısından anlamı, diğer bir deyişle metrik-kalite ilişkisi bir sonraki kısımda açıklandığı üzere Android yazılım geliştiricilerinin görüşleri ile elde edilmiştir. Bu nedenle, örneğin Sınıf Sayısının yüksek olması daha iyi bir özelliktir gibi tanımlar bu kısımda verilmemiştir.

Tablo 1. Kalite Modelinde Kullanılan Metriklerin Listesi

Metrik	Tür	Açıklama
<i>Sınıf Sayısı</i>	OO	Kaynak koddaki toplam sınıf sayısı.
<i>Kalıtım Ağacının Derinliği</i>	OO	Kalıtım hiyerarşisinde bir sınıftan kök sınıfa giden en uzun yolun uzunluğu.
<i>Kararsızlık</i>	OO	Götürgen bağımlılığın toplam bağımlılığa oranı (Efferent / Efferent + Afferent). Kararsızlık paketin değişime karşı olan dayanıklılığını göstermektedir. [18].
<i>Metotlardaki Uyum Eksikliği</i>	OO	Metotlardaki benzerlik derecesinin ölçümü.
<i>Nitelik Sayısı</i>	OO	Kaynak koddaki toplam nitelik sayısı.
<i>Metot Sayısı</i>	OO	Kaynak koddaki toplam metot sayısı.
<i>McCabe Çevrimsel Karmaşıklık</i>	OO	Programdaki lineer bağımsız yolların sayısı.
<i>Aktivite Sayısı</i>	Android	Aktivite (Activity) sınıfından doğrudan veya dolaylı olarak türetilmiş olan toplam sınıf sayısı.
<i>Servis Sayısı</i>	Android	Servis (Service) sınıfından doğrudan veya dolaylı olarak türetilmiş olan toplam sınıf sayısı.
<i>Yayın Alıcıları Sayısı</i>	Android	Yayın Alıcıları (BroadcastReceiver) sınıfından doğrudan veya dolaylı olarak türetilmiş olan toplam sınıf sayısı.
<i>İçerik Sağlayıcıların Sayısı</i>	Android	İçerik Sağlayıcı (Intent) sınıfından doğrudan veya dolaylı olarak türetilmiş olan toplam sınıf sayısı.
<i>Dialog Sayısı</i>	OO, Android	Kaynak koddaki toplam dialog sayısı.
<i>İş Parçacığı Sayısı</i>	OO, Android	Runnable interface sınıfını doğrudan veya dolaylı olarak implemente eden toplam sınıf sayısı.
<i>Minimum SDK Sürümü</i>	Android	Uygulamanın çalışması için gerekli olan minimum uygulama programlama arayüzü (API) düzeyini belirten tam sayı. En eski API düzeyi 1, en son Android API sürümü ise 23'tür. Belirtilmediği takdirde sistem API düzeyini "1" olarak kabul etmektedir.
<i>Hedeflenen SDK Sürümü</i>	Android	Uygulamanın çalışması için hedeflenen uygulama programlama arayüzü (API) düzeyini belirten tam sayı. Veri belirtilmediği takdirde sistem API düzeyini Minimum SDK Sürümü olarak kabul etmektedir.
<i>Intent Sayısı</i>	Android	Intent sınıfından doğrudan veya dolaylı olarak türetilmiş olan toplam sınıf sayısı.
<i>Veritabanındaki Tabloların Sayısı</i>	OO, Android	Uygulamanın veritabanındaki tabloların toplam sayısı.
<i>İzin Sayısı</i>	Android	Uygulamanın düzgün çalışmak için talep ettiği izin sayısı.

2.3 Metrik-Kalite Özelliği İlişkisi

Kaynak kod metrikleri ile kalite özellikleri arasındaki ilişkiyi saptamak için Android uygulama geliştiricilerine anket yapılmıştır. Ankette katılımcılardan her metriğin kalite özellikleri üzerindeki etkisinin derecesini ([-1, 1] aralığında Likert Ölçeği) seçmelerini talep eden 18 soru mevcuttur. Seçenekler Kesinlikle Olumsuz (-1), Kısmen Olumsuz (-0,5), Ne Olumsuz Ne Olumlu (0), Kısmen Olumlu (+0,5) and Kesinlikle Olumlu (+1) şeklinde belirlenmiştir. Anket çevrimiçi gönderilecek şekilde planlanmış ve hazırlanması aşamasında Google Forms [19] kullanılmıştır.

Anket sosyal medyadaki (Facebook, LinkedIn) Android uygulama geliştirici gruplarına, Türkiye'deki muhtemel bilişim firmalarına ve kişisel bağlantılarımıza gönderilmiştir. Ankete 33 Android geliştirici katılmıştır. Katılımcıların aktif olarak yer aldığı Adroid yazılımı projeleri sayılarına göre dağılımı şu şekildedir: %15,2si 10'dan fazla projede, %27,3ü ise 5-10 projede, %57,6sı 1-4 projede yer almıştır. Katılımcıların %27,3ü şirketleri için, %48,5 kişisel olarak, %24,2si ise hem kişisel hem de şirketleri için Android yazılımı geliştirmiştir. Anket sonuçlarının iç tutarlılığı kontrol edilmiştir. Cronbach Alfa değerinin ortalaması 0.82 olduğundan anketin güvenilirliği kabul edilebilir bir aralık içindedir.

İlişki Ağırlıklandırma. Anket sonuçlarını kullanarak metrikler ve kalite özellikleri arasındaki ilişkiyi gösteren bir matris oluşturulmuştur. Bu matris Tablo 2'de gösterilmektedir. Tabloda sütunlar kalite özelliklerini, satırlar ise metriklerin kalite özellikleri üzerindeki etkilerini içermektedir. Matris bir metriktaki artışın kalite özelliklerini nasıl etkilediğini göstermektedir. Tabloda 0 ne olumlu ne olumsuz etki, -1 kesinlikle olumsuz etki, -0,5 kısmen olumsuz etki, +0,5 kısmen olumlu etki, +1 ise kesinlikle olumlu etkiyi ifade etmektedir.

Tablo 2. Metrik - Kalite Özelliği İlişkisi

	İşlevsel Uygunluk	Güvenilirlik	Performans Verimliliği	Taahhütlilik	Bakım Kolaylığı	Veri Bütünlüğü	Kullanılabilirlik	Güvenlik	Bağdaşlılık
Sınıf Sayısı (NOC)	+0.5	0	+1.0	+1.0	+0.5	0	+0.5	+0.5	0
Kalıtım Ağacının Derinliği (DIT)	+0.5	-0.5	-1.0	+0.5	+0.5	+0.5	0	0	+0.5
Kararsızlık (COP)	-0.5	-0.5	0	0	-0.5	+0.5	0	+0.5	+0.5
1 / Metotlardaki Uyum Eksikliği (COH)	+0.5	+0.5	0	0	0	+0.5	0	+0.5	+0.5
Nitelik Sayısı (NOAT)	0	+0.5	-0.5	0	0	+0.5	+0.5	+0.5	+0.5
Metot Sayısı (NOM)	+0.5	0	0	+0.5	-0.5	+0.5	+0.5	+0.5	0
McCabe Çevrimsel Karmaşıklık (CYC)	0	0	-1.0	0	0	0	0	0	0
Aktivite Sayısı (NOAC)	+0.5	0	0	0	+0.5	+1.0	+0.5	0	0
Servis Sayısı (NOS)	+0.5	+0.5	-0.5	+0.5	+0.5	+0.5	+0.5	-0.5	0
Yayın Alıcıları Sayısı (NOBR)	+0.5	+0.5	0	0	0	0	0	0	0
İçerik Sağlayıcıların Sayısı (NOCP)	-0.5	+0.5	+0.5	+0.5	+0.5	0	+0.5	+0.5	0
Dialog Sayısı (NOD)	0	0	-1.0	0	0	0	+0.5	0	0
İş Parçacığı Sayısı (NOT)	+0.5	+0.5	+1.0	0	+0.5	+0.5	+0.5	0	0

Minimum SDK Sürümü (MSDK)	+0.5	0	0	-1.0	0	0	0	0	0
Hedeflenen SDK Sürümü (TSDK)	+0.5	0	0	0	0	0	0	0	0
Intent Sayısı (NOI)	+0.5	0	0	+0.5	0	+0.5	+0.5	+0.5	+1.0
Veritabanındaki Tabloların Sayısı (NOTD)	+0.5	0	-1.0	+0.5	+0.5	+0.5	0	-1.0	+0.5
İzin Sayısı (NOP)	0	0	+0.5	0	0	0	-0.5	+0.5	0

Tablo 2 ile gösterilen veriler için katılımcıların yanıtlarının medyan değerleri kullanılmıştır. Bu tabloda iş parçacığı sayısı ve katılım ağacı derinliği metrikleri için beklenenden farklı sonuç alınmıştır. Katılım ağacı derinliği taşınabilirlik ve bakım için ters orantılı olarak beklenirken +0,5 etki sonucu alınmıştır. İş parçacığı sayısı için de bakım kolaylığını ve güvenilirliği ters etkilemesi beklenirken +0,5 etki yanıtı alınmıştır. Android geliştirme ortamının sağladığı özellikler ve özellikle işçi (worker thread) parçacıklarının kullanımın teşviki bu etki değerini çıkarmış olabilir. Bu durum gelecekteki çalışmalarda daha detaylı incelenecektir.

Tablo 2'deki matris kullanılarak her kalite özelliği için hesaplama formülleri elde edilmiştir. Bütün kalite özellikleri için hesaplanan değerlerin aynı aralıkta olması için metriklerin kalite özellikleri üzerindeki etkileri ağırlıklandırılmıştır. Metrik etkileri için aralık olarak ± 1 aralığı seçildiğinden, (1) de belirtilmiş olan formül kullanılarak her metriğin ağırlığı normalize edilmiştir.

$$\text{Yeni Değer} = \text{Metrik Değeri} / |\text{Metrik Değerlerinin Toplamı}| \quad (1)$$

Hesaplamaya bir örnek vermek gerekirse: Kalıtım Ağacının Derinliği (DIT) metriğinin Bağdaşırılık özelliğini +0,5 etkilediği görülmektedir. Bağdaşırılık özelliğini etkileyen bütün metriklerin değerleri Tablo 2 son sütunda görülmektedir. Bu değerler toplandığında $(+0,5+0,5+0,5+0,5+1,0+0,5)= 3,5$ değeri elde edilmektedir ve Kalıtım Ağacının Derinliği (DIT) metriğinin yeni değeri $0,5/3,5 = 0,14$ olarak bulunur.

Kalite özellikleri için elde edilmiş olan nihai hesaplama formülleri Tablo 3'te gösterilmiştir. Bu formüllere göre örneğin sınıf sayısının (NOC) fazla oluşu performans verimliliğini arttırırken, kalıtım ağacının derinliğinin (DIT) fazla oluşu performans verimliliğini azaltmaktadır.

Tablo 3. Kalite Özellikleri için Hesaplama Formülleri

Kalite Özelliği	Hesaplama Formülleri
Performans Verimliliği	$+0.5*(NOC) -0.5*(DIT) -0.25*(NOAT) -0.5*(CYC) -0.25*(NOS) +0.25*(NOCP) -0.5*(NOD) +0.5*(NOT) -0.5*(NOTD) +0.25*(NOP)$
Taşınabilirlik	$+0.33*(NOC) +0.165*(DIT) +0.165*(NOM) +0.165*(NOS) +0.165*(NOCP) -0.33*(MSDK) +0.165*(NOI) +0.165*(NOTD)$
Bakım Kolaylığı	$+0.2*(NOC) +0.2*(DIT) -0.2*(COP) -0.2*(NOM) +0.2*(NOAC) +0.2*(NOS) +0.2*(NOCP) +0.2*(NOT) +0.2*(NOTD)$
İşlevsel Uygunluk	$+0.1*(NOC) +0.1*(DIT) -0.1*(COP) +0.1*(COH) +0.1*(NOM) +0.1*(NOAC) +0.1*(NOS) +0.1*(NOBR) -0.1*(NOCP) +0.1*(NOT) +0.1*(MSDK) +0.1*(TSDK) +0.1*(NOI) +0.1*(NOTD)$
Güvenilirlik	$-0.25*(DIT) -0.25*(COP) +0.25*(COH) +0.25*(NOAT) +0.25*(NOS) +0.25*(NOBR) +0.25*(NOCP) +0.25*(NOT)$
Veri Bütünlüğü	$+0.09*(DIT) +0.09*(COP) +0.09*(COH) +0.09*(NOAT) +0.09*(NOM) +0.18*(NOAC) +0.09*(NOS) +0.09*(NOT) +0.09*(NOI) +0.09*(NOTD)$

Kullanılabilirlik	+0.125*(NOC) +0.125*(NOAT) +0.125*(NOM) +0.125*(NOAC) +0.125*(NOS) +0.125*(NOCP) +0.125*(NOD) +0.125*(NOT) +0.125*(NOI) -0.125*(NOP)
Güvenlik	+0.2*(NOC) +0.2*(COP) +0.2*(COH) +0.2*(NOAT) +0.2*(NOM) - 0.2*(NOS) +0.2*(NOCP) +0.2*(NOI) -0.4*(NOTD) +0.2*(NOP)
Bağdaşlılık	+0.14*(DIT) +0.14*(COP) +0.14*(COH) +0.14*(NOAT) +0.28*(NOI) +0.14*(NOTD)

3 MODEL DEĞERLENDİRMESİ VE SONUÇLAR

3.1 Model Doğrulama

Android kalite modelinin doğrulaması için iki uygulama üzerinde çalışan üç bağımsız değerlendirici kullanılmıştır. Tüm değerlendiricilerin yazılım değerlendirme ve Android uygulama programlama konularında dört ya da beş yıllık tecrübeleri bulunmaktadır.

Açık kaynak kodlu doğrulama suiti seçmek amacıyla Android platform uygulamaları için ücretsiz ve açık kaynak kodlu bir katalog olan F-Droid [22] kullanılmıştır. F-Droid'in listesinden popüler, açık kaynaklı, küçük boyutlu (en fazla 5000 KLOC), en az iki dağıtım yapılmış olan bir uygulama araştırılmıştır. Bu kriterlere uygun olan 2048-android uygulamasının ilk ve son dağıtımları seçilmiştir. 4275 KLOC boyutundaki 2048-android uygulaması [20] basit bir bulmaca oyunudur.

Üç bağımsız değerlendirici, 2048-android v1 ve 2048-android v2.06 yazılımlarının kaynak kodlarını analiz etmiş ve kalite özelliklerini puanlamışlardır. Katılımcılar her özelliği [0, 10] aralığında puanlamıştır. Tablo 4, üç değerlendiricinin (D1, D2 ve D3) puanlarını ve kalite özelliklerinin ortalama değerlerini (ORT) göstermektedir.

Tablo 4. 2048-android Uygulamaları İçin Değerlendirici Skorları

Değerlendiriciler	2048-android v1.0				2048-android v2.06			
	D1	D2	D3	ORT	D1	D2	D3	ORT
Performans Verimliliği	4	2	5	3,66	6	5	8	6,33
Taşınabilirlik	2	5	7	4,66	8	10	8	8,66
Bakım Kolaylığı	7	6	8	7	4	6	7	5,66
İşlevsel Uygunluk	5	4	6	5	8	7	10	8,33
Güvenilirlik	4	3	6	4,33	7	9	10	8,66
Veri Bütünlüğü	5	3	6	4,66	8	7	9	8
Kullanılabilirlik	4	5	8	5,66	7	9	10	8,66
Güvenlik	6	4	7	5,66	8	7	8	7,66
Bağdaşlılık	2	3	5	3,33	6	5	5	5,33

Değerlendiricilerin puanları alındıktan sonra, seçilen uygulamaların kaynak kodları analiz edilerek iki dağıtıma ait kalite puanları hesaplanmıştır. Kalite puanlarını hesaplayabilmek için öncelikle, Bölüm 3.3 ile anlatılan yöntemle, iki dağıtımın metrikleri toplanmıştır. Kalite özelliklerinin hesaplanması sürecinde farklı aralık değerlerindeki metriklerin toplanmış olması sebebiyle bu değerlerin normalize edilmesi gerekmiştir. Metrikler [0, 1] aralığında Min-Max normalizasyonu [28] ile normalize edilmişlerdir.

Tanımlanan dokuz kalite özelliği Tablo 3 de açıklanan formüller kullanılarak hesaplanmıştır. Her bir formül için normalize edilmiş olan metrik değerleri kullanılmıştır. Tablo 5, 2048-android uygulamasının iki dağıtım için hesaplanmış dokuz kalite özelliğine ait değerleri göstermektedir.

Değerlendiricilerin puanları ile kalite modelinin sonuçları arasında korelasyon bulunup bulunmadığını kontrol etmek için Pearson'ın korelasyon katsayısı [24] kullanılmıştır. 9 çiftin katsayı değerleri IBM SPSS Statistics [21] kullanılarak elde edilmiştir. 0.697 lık katsayı değeri ile iki veri seti arasında istatistiksel olarak anlamlı bir korelasyon bulunduğu ortaya çıkmıştır. Bu korelasyon çalışmamızda oluşturulan Android kalite modelini doğrulamakta ve modelin güvenilirliğini desteklemektedir.

Tablo 5. 2048-android Uygulamaları İçin Hesaplanan Kalite Özellikleri

Kalite Özelliği	2048-android v1.0	2048-android v2.06
Performans Verimliliği	3,125	5,625
Taşınabilirlik	0	5
Bakım Kolaylığı	5,7	1,11
İşlevsel Uygunluk	2,14	3,57
Güvenilirlik	2,5	3,75
Veri Bütünlüğü	0	3,63
Kullanılabilirlik	1	4
Güvenlik	2,72	7,27
Bağdaşılık	0	4,28

Değerleme Ölçeğinin Belirlenmesi. Sonuçlar hakkında bilgi çıkarımı yapabilmek amacıyla değerlendirme için Android uygulamaların kalite özelliklerinin kalite puanları belirlenmiştir. Değerleme ölçeği altı kategoriden oluşmaktadır: 0-0,99, Çok Zayıf, 1-2,99 Zayıf, 3-4,99 Orta, 5-6,99 İyi, 7-8,99 Çok İyi ve 9-10 Mükemmel.

Değerleme ölçeği kalite özelliklerinin sonuçlarına uygulanmıştır. Kalite özellikleri 2048-android uygulamaları için Android Kalite Modeli ile hesaplanmış ve derece değerleri Tablo 6'da verilmiştir.

Tablo 6. 2048-android Uygulamalarının Kalite Özelliklerinin Kategorileri

Kalite Özelliği	2048-android v1.0	2048-android v2.06
Performans Verimliliği	3,125 ORTA	5,625 İYİ
Taşınabilirlik	0 ÇOK ZAYIF	5 İYİ
Bakım Kolaylığı	5,7 İYİ	1,11 ZAYIF
İşlevsel Uygunluk	2,14 ZAYIF	3,57 ORTA
Güvenilirlik	2,5 ZAYIF	3,75 ORTA
Veri Bütünlüğü	0 ÇOK ZAYIF	3,63 ORTA
Kullanılabilirlik	1 ZAYIF	4 ORTA
Güvenlik	2,72 ZAYIF	7,27 ÇOK İYİ
Bağdaşılık	0 ÇOK ZAYIF	4,28 ORTA

3.2 Örnek Olay İncelemesi İçin Test Verisinin Toplanması

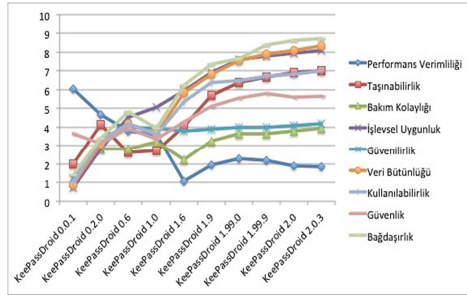
F-Droid’de açık kaynaklı, büyük boyutlu (en az 10000 KLOC), birçok dağıtımı olan uygulamalar araştırılmıştır ve örnek olay incelemesi için iki Android uygulaması seçilmiştir: KeePassDroid [25] ve CosyDVR [26]. İlk uygulama KeePassDroid, 29.328 KLOC büyüklüğünde ve 115 dağıtımı yapılmış olan ücretsiz açık kaynak şifre yönetim uygulamasıdır. Diğer uygulama CosyDVR ise 24.087 KLOC büyüklüğünde ve 21 dağıtımı yapılmış araç içi kullanım için tasarlanmış ücretsiz bir açık kaynak Dijital Video Kaydedici (DVK) yazılım kitidir.

KeePassDroid uygulamasının, 115 dağıtımından 10 tanesinin, CosyDVR uygulamasının ise yine 21 dağıtımından 10 tanesinin kaynak kodları indirilmiştir. KeePassDroid ve CosyDVR uygulamalarının 10’ar dağıtımı seçilirken uygulamaların havuzlarında en çok kayıt (commit) sayısına sahip 10 dağıtım göz önünde bulundurulmuştur. Uygulamalar havuzda ne kadar çok kayıt sayısına sahipse, uygulamada o kadar çok değişiklik yapılmış anlamına gelmektedir.

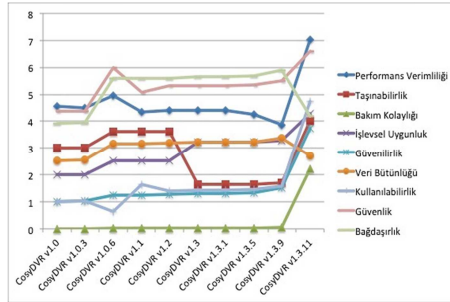
3.3 Metrik Verisinin Toplanması ve Normalize Edilmesi

Tablo 1’de tanımladığımız 18 metrik KeePassDroid ve CosyDVR uygulamalarının 10 dağıtımı için hesaplanmıştır. Bu çalışmada Tablo 2’de gösterilen nesne tabanlı metrikleri hesaplamak için Eclipse Metrics plug-in 1.3.8 [27] uygulaması kullanılmıştır. Bizim tanımladığımız ek metrikleri hesaplayabilmek için özel herhangi bir araç ya da plug-in bulunmadığından, bir Java programı yazılmıştır.

Kalite özelliklerinin hesaplanması esnasında asıl metrik değerlerinin farklı aralıklardan toplanması sebebiyle metrik değerlerinin normalize edilmesine ihtiyaç duyulmuş ve Min-Max normalizasyonu ile bu işlem gerçekleştirilmiştir.



Şekil 2. KeePassDroid Uygulamasının Kalite Özelliklerinin Grafikleri



Şekil 3. CosyDVR Uygulamasının Kalite Özelliklerinin Grafikleri

3.4 Sonuçların Analiz Edilmesi

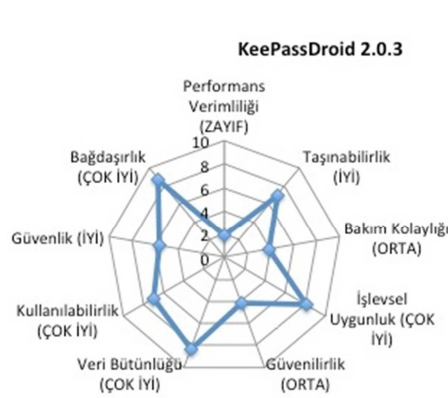
Tanımlanan dokuz kalite özelliğinin değerleri, Tablo 3’de açıklanan formüllerle hesaplanmıştır. Kalite özellik grafikleri, hesaplanan bu değerleri baz alarak çizilmiştir.

Şekil 2 ve Şekil 3’deki KeePassDroid ve CosyDVR uygulamalarına ait kalite özellik grafikleri elde edilmiştir.

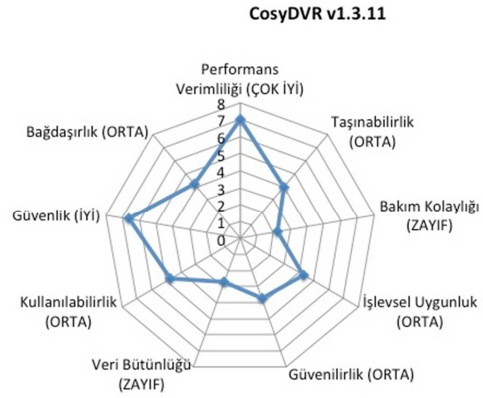
Kalite özelliklerine ait kalite puanlarındaki beklenen artış, kalite özelliklerinin yeni Android uygulamalarının dağıtımlarında artacağı öngörüsünü doğrulamıştır. KeePassDroid uygulamasının son dağıtımına ait kalite puanlarının çoğu “Orta” ya da daha üstteki kategoriye girmiştir. Diğer taraftan CosyDVR’in son dağıtımına ait puanlar ise genellikle “Orta” ya da daha düşük kategorilerdedir. KeePassDroid’in son dağıtımına ait kalite puanları Şekil 4’te, CosyDVR’e ait olanlar ise Şekil 5’de gösterilmiştir.

KeePassDroid ve CosyDVR uygulamalarının indirilme sayıları ve bu uygulamalara yapılmış olan yorumlarla, Android kalite modelimiz kullanılarak hesaplanmış olan kalite puanları birbiri ile tutarlı bir profil çizmektedir. Kullanıcı yorumlarına göre, Google Play Store’da KeePassDroid değerlemeleri yüksekken, CosyDVR değerlemeleri düşüktür. Google Play Store’daki uygulamaların indirilme sayıları ve yıldız değerleri aşağıda yer almaktadır:

- KeePassDroid [29]: İndirilme sayısı 1,000,000 - 5,000,000 aralığındadır. 31,305 kullanıcı yorumuna göre uygulama 5.0 üzerinden 4.6 puan değerine sahiptir.
- CosyDVR [30]: İndirilme sayısı 100 – 500 aralığındadır. 6 kullanıcı yorumuna göre uygulama, 5 üzerinden 2.5 puan değerine sahiptir.



Şekil 4. KeePassDroid v.2.0.3 Kalite Skorları



Şekil 5. CosyDVR v1.3.11 Kalite Skorları

4 SONUÇ VE GELECEKTEKİ ÇALIŞMALAR

Bu çalışmada Android uygulamaları için kalite modeli önerilmiştir. Önerimiz genel bir model olup, Android uygulama geliştiren yazılımcılar kendi uygulamalarına göre kalite modelini uyarlayabileceklerdir.

Yeni bir disiplini olan mobil yazılım kalite değerlendirmesi üzerine yaptığımız bu çalışmada bazı kısıtlar vardır. İlk kısıt, araştırmada yer alacak katılımcıların seçimidir. Android 2007 yılında doğup 2010 yılında yaygınlaştığı için [31], bu konuda yetkin yazılımcılar bulmak zordur. Bu sorunu gidermek adına Sosyal Medyadaki (Facebook,

LinkedIn) Android uygulama geliştirici gruplarına e-posta atılmıştır. Farklı ülkelerden katılımcı sağlamak amacıyla sosyal medyadaki gruplara mesaj atılmasına rağmen araştırmaya katılan 33 geliştiricinin tamamı Türkiye’den olmuştur. Türk yazılımcıların karakteristikleri diğer ülkelerdekilerden farklı olabileceğinden, başka ülkelerden geliştiricilerin katılacağı bir araştırmada farklı sonuçlar olabilecektir. Bu sınırlamanın ilerideki çalışmalarımızda aşılması planlanmaktadır.

Bir sonraki kısıt ise, model doğrulamada kullanılan mobil uygulamaların boyutu ile ilgilidir. Yazılım değerlendirme emek yoğun bir iş olduğundan, orta büyüklükte uygulamalar kullanılmıştır. Çalışmanın daha büyük ölçekli uygulamalar üzerinde de doğrulanması amacıyla genişletilmesi hedeflenmektedir.

Bu çalışma kapsamına ürün kalite modelinde yer alan sekiz özellik dahil edilmiş, bu sekiz özelliğin alt özellikleri dahil edilmemiştir. Bu alt özelliklerin de mobil uygulama kalitesinin değerlendirilmesi açısından incelenmesi planlanmaktadır.

Gelecekteki çalışmalarda kalite modelinde bazı geliştirmeler yapılacaktır. Mobil uygulamaların kaynak kod metriklerinin istatistiksel olarak daha büyük kapsamda analiz edilmesi ve farklı mobil platformlarının (örn. IOS, Windows Mobile) kalite modeline dâhil edilmesi planlanmaktadır. Sistemin sonuçlarını daha etkin hale getirmek için farklı ülkelerdeki mobil uygulama geliştiricilerini de kapsayan daha fazla yazılımcıdan veri toplanacak bir model geliştirilecektir. Ayrıca kalite modelini doğrulamak için daha fazla değerlendirici çalışmaya dâhil edilecektir. Değerlendirici sayısındaki çoğalma, modelin güvenilirliğini de arttıracaktır.

5 KAYNAKÇA

1. “Predicts 2014: Apps, Personal Cloud and Data Analytics Will Drive New Consumer Interactions”, <https://www.gartner.com/doc/2628016/predicts--apps-personal-cloud>, Ekim 21, 2015.
2. “The Statistics Portal: Number of Apps Available in Leading App Stores”, <http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>, Ekim 24, 2015.
3. Franke, D., Kowalewski, S., “A Mobile Software Quality Model”, in 12th International Conference on Quality Software, Xi’an, Shaanxi: IEEE, 2012.
4. Franke, D., Weise, C., “Providing a Software Quality Framework for Testing of Mobile Applications”, Fourth IEEE International Conference on Software Testing, Verification and Validation, Berlin: IEEE, 2011.
5. Idri, A., Moumane, K., Abran, A., “On the Use of Software Quality Standard ISO / IEC 9126 in Mobile Environments”, 20th Asia-Pacific Software Engineering Conference, Aveiro: IEEE, 2013.
6. Zahra, S., Khalid, A., Javed, A., “An Efficient and Effective New Generation Objective Quality Model for Mobile Applications.” I.J.Modern Education and Computer Science (MECS), 4, 36-42, 2013.
7. Hecht, G., Benomar, O., Rouvoy, R., Moha, N., Duchien, L., “Tracking the Software Quality of Android Applications along their Evolution”, 30th IEEE/ACM International Conference on Automated Software Engineering, United States: IEEE, 2015.
8. “Android Developer Page”, <http://developer.android.com/>, Kasım 12, 2015.
9. Jetter, A., “Assessing Software Quality Attributes with Source Code Metrics. (Diploma Thesis)”, http://www.ifi.uzh.ch/seal/research/tools/archive/swQuality/da_jetter.pdf, 2006.

10. Meier, R., "Professional Android 4 Application Development", https://books.google.com.tr/books?hl=en&lr=&id=bmJlI_wPgQsC, 2012.
11. Smyth, N., "Android 4 App Development Essentials" http://www.techotopia.com/index.php/Android_4_App_Development_Essentials, 2014
12. Murphy, M., "The Busy Coder's Guide to Android Development" https://commonsware.com/Android/Android_3-6-CC.pdf, 2012.
13. Gramlich, N., "Android Programming with Tutorials from the anddev.org-Community", <http://andbook.anddev.org/files/andbook.pdf>, 2012.
14. Önder, M., Mermerkaya, A., "Merhaba Android", İstanbul, Turkey: Deniz Ofset, 2013.
15. Vogel, L., "Introduction to Android development with Android Studio", <http://www.vogella.com/tutorials/Android/article.html>, 2015.
16. "Android Development", <https://teamtreehouse.com/tracks/android-development>, Kasım 20, 2015.
17. "Difference between Android and Java", <http://www.differencebetween.com/difference-between-android-and-vs-java/#ixzz2TikUX7cI>, Aralık 17, 2015.
18. Almugrin, S., Albattah, W., Alaql, O., Alzahrani, M., Melton, A., "Instability and Abstractness Metrics Based on Responsibility", 38th Annual International Computers, Software and Applications Conference, Sweden: IEEE, 2014.
19. "Google Forms", <https://www.google.com/forms/about/>, Aralık 20, 2015.
20. "2048-android game in Google Play Store" <https://play.google.com/store/apps/details?id=com.gabrielecirulli.app2048>, Aralık 25, 2015.
21. "IBM SPSS Statistics", <http://www-01.ibm.com/software/analytics/spss/>, Aralık 20, 2015.
22. "F-Droid Limited", <https://f-droid.org/wiki/index.php?title=Category: Apps>, Aralık 22, 2015.
23. "Pearson product-moment correlation coefficient", <https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php>, Mayıs 12, 2016.
24. "KeePassDroid", <http://www.keepassdroid.com/>, Aralık 22, 2015
25. "CosyDVR", <https://github.com/sergstetsuk/CosyDVR>, Şubat 10, 2015.
26. "Eclipse Metrics Plugin 1.3.8.", <http://metrics2.sourceforge.net/>, Aralık 25, 2015.
27. "Data Normalization and Standardization", <http://www.benetz Korn.com/2011/11/data-normalization-and-standardization/>, Aralık 25, 2015.
28. "KeePassDroid in Google Play Store", <https://play.google.com/store/apps/details?id=com.android.keepass&hl=en>, Aralık 25, 2015.
29. "CosyDVR in Google Play Store", <https://play.google.com/store/apps/details?id=es.esy.CosyDVR>, Şubat 10, 2015.
30. "The Android Story", <http://www.xcubelabs.com/infographic-android-story/>, Şubat 10, 2015.
31. "ISO/IEC 25010:2011", <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>, Aralık 2, 2015.
32. G. Jost, J. Huber, and M. Hericko. "Using Object Oriented Software Metrics for Mobile Application Development", in *2nd Workshop of Software Quality Analysis, Monitoring, Improvement and Applications (SQAMIA)*, Serbia, 2013.