

YAZILIM GELİŞTİRME KALİTESİNİN KARŞILAŞTIRILMASINDA TOPSIS - VIKOR YÖNTEMİNİN UYGULANMASI

Özge Tuğrul

İstanbul Teknik Üniversitesi İşletme Fakültesi, Endüstri Mühendisliği Bölümü
Maçka, 34367, İstanbul, Türkiye
tugrulo@itu.edu.tr

Özet. Günümüzde bilgi teknolojileriyle ilgilenen şirketlerin temel amaçlarından biri müşteri taleplerini karşılayan, hatasız yazılımlar üretmektir. Yazılım kalitesinin belirlenmesinde birçok değerlendirme kriteri bulunmaktadır. Bu kriterler arasında yazılımın geçerliliği, performansı, fonksiyonelliği, sürdürülebilirliği, yeniden kullanılabilirliği, test edilebilirliği, güvenilirliği, dokümantasyonu gibi kriterler yer almaktadır (Rehman&Sarrab, 2014). Bu çalışmada, bir yazılım şirketinde bulunan 10 farklı yazılım geliştirme bölümünün 3 aylık bir sürede ürettikleri uygulama yazılımlarının kalitelerinin karşılaştırılabilmesi için sırası ile Technique For Order Preference By Similarity To An Ideal Solution (TOPSIS) ve Vise Kriterijumska Optimizacija I Kompromisno Resenje (VIKOR) yöntemleri kullanılmıştır. TOPSIS yöntemi kullanılarak bölümlerin uygulama yaygınlaştırmaları öncesinde yapılan kod ve ekran gözden geçirmelerinde bulunan bulgu tiplerinin oranlarına ve veri tablosu hata oranlarına göre aldıkları puanlar, kriterleri üretim ortamına aktarım ve kontrol tipleri olan VIKOR yöntemi ile değerlendirilip, bölümlerin geliştirdikleri yazılımların genel performanslarının karşılaştırmaları elde edilmiştir. Bu çalışmada farklı bir yaklaşım olarak TOPSIS ve VIKOR yöntemleri entegre bir şekilde kullanılmış ve toplam 96 adet olan kriter sayısı her bir kontrol kriteri (8 adet) için 12 adet bulgu kriteri kullanılarak iki aşamalı bir değerlendirme biçiminde uygulanmıştır.

Anahtar Kelimeler: Çok Kriterli Karar Verme, Yazılım Kalite Değerlendirme, TOPSIS, VIKOR, Yazılım Kalite Karşılaştırma

1 Giriş

Yazılım performanslarının nesnel ve farklı kriterlere dayandırılarak, doğru bir şekilde ölçümlenebilmesi oldukça önem taşımaktadır. Yazılım kalitesinin doğrudan ölçülebilmesi için bazı ön koşulların sağlanması gerekmektedir. Bu sebeple kalite özellikleri için ortak bir anlayış oluşturulmalıdır. Herhangi bir yazılımın daha iyi veya aynı kalite düzeyinde olduğuna dair ilişki kurulabilmelidir ve bu doğrultuda bir ölçüm metodu geliştirilmelidir (Jørgensen, 1999). Birimlerin yazılımlarının kaliteleri arasında ilişkisel bir ölçüm kriteri oluşturabilmek amacıyla, bu birimlerin kodlarına ait

karşılaştırma yapılabilmesi için kod gözden geçirme, kullanıcı ara yüzü denetimi, veri yapısı ve mimarisinin onaylanması ile ilgili faaliyet alanları değerlendirilmektedir. Kod gözden geçirme faaliyetleriyle ilgili kalite ölçümü kod kontrol listelerine göre oluşturulan bulgu tipleri; kullanıcı ara yüz denetimi ile ilgili kalite ölçümü ekran kontrol listelerine göre oluşturulan bulgu tipleridir. Veri yapısı ve mimarisine ilgili kalite ölçümü ise onaylanan veri tablolarının oranıdır.

Daha önce gerçekleştirilen yazılım kalite değerlendirme çalışmalarında yazılım kaliteleri değerlendirilirken birden çok alternatifin kalite kriterleri için karşılaştırılması yapılmamıştır. Ustasüleyman (2009), bankacılık sektörü için hizmet kalitesinin değerlendirilmesi amacıyla AHS (Analitik Hiyerarşi Süreci) ve TOPSIS yöntemlerini uygulamıştır. Benzer şekilde Demireli (2010), üç adet kamu bankası (Halk Bankası, Türkiye Vakıflar Bankası ve Ziraat Bankası) için Türkiye Bankalar Birliği'nin <http://www.tbb.org.tr/tr> web sitesinden bankaların finansal tablolarını inceleyerek finansal kriterlere göre yıllar itibarıyla bankaların performans kıyaslamalarını TOPSIS yöntemiyle elde etmiştir. Ertuğrul ve Karakaşoğlu (2009) bir bankaya ait şube performanslarının finansal kriterlere göre sıralamasını VIKOR yöntemi ile gerçekleştirmiştir. Bu çalışmada da diğer sektörler için uygulanmış Çok Kriterli Karar Verme (ÇKKV) yöntemlerinden TOPSIS ve VIKOR yöntemleri farklı bir bakış açısıyla TOPSIS yönteminden elde edilen puanların VIKOR yöntemi kriter değerleri olarak kullanılarak yazılım kalitelerinin değerlendirilmesi amacıyla kullanılmıştır.

Bu çalışma üç bölümden oluşmaktadır. Birinci bölümde kullanılan verilerin yapısı ve uygulanacak metodoloji hakkında bilgi verilmektedir. İkinci bölümde, verilere dayalı yöntemin uygulaması yer almaktadır. Üçüncü bölümde ise sonuçlar ve değerlendirmeler bulunmaktadır.

2 Kullanılan Veri ve Metodoloji

Bu uygulamada yazılım kalitesinin ölçülmesi amacıyla üç temel kriter göz önünde bulundurulacaktır. Bu kriterler kod gözden geçirme, kullanıcı ara yüz denetimi ve veri yapısı ve mimarisinin onaylanmasıdır. Kod gözden geçirmede standartlara uygunluk, kod tekrarı ve performans değerlendirilmektedir. Kullanıcı ara yüz denetiminde standartlara uygunluk, tasarım ve kullanılabilirlik değerlendirilmektedir. Bu amaçla kod kontrol listesi ve ekran kontrol listesine göre bulgu ve uymazlıklar tespit edilmektedir. Veri yapısı ve mimarisinin onaylanmasında standartlara uygunluk ve ER diyagram kontrolü yapılmaktadır. Geliştirilen yazılım kod parçaları (bundan sonra "nesne" olarak kullanılacaktır) halinde yaygınlaştırılmakta olup, nesneleredeki bulgu sayıları kalite göstergesi olarak alınmaktadır. Yaygınlaştırmalar (yazılımın uygulamaya alınması) dört farklı şekilde gerçekleştirilmektedir. Bunlar, "Acil Yaygınlaştırmalar", "Aylık Yaygınlaştırmalar", "Haftalık Yaygınlaştırmalar" ve "Pilot Yaygınlaştırmalar" 'dır. Yaygınlaştırmalar için kod gözden geçirme ve kullanıcı ara yüz denetimi gerçekleştirilmektedir. Veri yapıları değerlendirilirken "İsmlendirme Problemleri", "Tablo Tasarım Hataları", "Index Hataları", "Normalizasyon Hataları", "Açıklama Hataları" gibi hatalar dikkate alınmaktadır.

Kod ve Ekran gözden geçirmeler sonucunda elde edilen nesne başına düşen bulgu sayıları bölümlerin yazılım kalitelerini ölçmede kullanılır. Bulgular kendi içerisinde kritiklik derecesine ve bulgu çözüm durumuna göre sınıflandırılarak kriterler oluşturulmuştur. Yazılım geliştiren bölümler ise alternatif olarak alınmıştır. Bölümlerin sıralamaları, yaygınlaştırdıkları kod parçalarının bulgu sayıları ve bu bulguların kritiklik derecesine göre gerçekleştirilecektir. Bölümlerin yazılım geliştirme kalitelerini değerlendirmede hem bulgu kriterleri hem de kontrol yöntemi büyük önem taşıdığından değerlendirme yapmak amacıyla iki farklı çok ölçütlü karar verme yöntemi art arda kullanılacaktır. Bu amaçla her bir kontrol kriteri içerisinde yer alan bulgu kriterlerine (12 adet) göre bölümlerin sıralamasını yapmak için TOPSIS yöntemi kullanılacaktır. TOPSIS yöntemi ile elde edilen bölümlere ait C_i^* değerleri i bölümünün değerlendirilen kontrol kriterindeki başarısını gösterir. 1'e yakın değerler yüksek kalite göstergesidir. Bölümlerin her bir kontrol kriteri için kalite puanları elde edildikten sonra farklı kontrol kriterleri (8 adet) için elde edilen bu puanlar VIKOR yöntemi ile değerlendirilerek genel bir yazılım kalite puanı elde edilecektir. Genel yazılım kalitesinin değerlendirilmesinde VIKOR yönteminin kullanılmasının nedeni, bu yöntem tüm alternatiflerin ideal alternatife “yakınlık” ölçüsünde dayanan bir sıralama dizini oluşturması ve VIKOR yönteminin TOPSIS yönteminden farklı olarak karar vericilere daha çok alternatif sunabilmesidir (Ertuğrul & Karakaşoğlu, 2009). Öncelikli olarak aşağıdaki tabloda yer alan kriterler için bölümlerin TOPSIS yöntemine göre alacağı puanlar belirlenecektir.

Bulgu Kriterleri	
KD: Kritik ve Düzeltilmiş	UD: Uyarı verilmiş ve Düzeltilmiş
KE: Kritik ve Ertelenmiş	UE: Uyarı verilmiş ve Ertelenmiş
KA: Kritik ve Aktif	UA: Uyarı verilmiş ve Aktif
RD: Reddedilmiş ve Düzeltilmiş	ÖD: Öneri verilmiş ve Düzeltilmiş
RE: Reddedilmiş ve Ertelenmiş	ÖE: Öneri verilmiş ve Ertelenmiş
RA: Reddedilmiş ve Aktif	ÖA: Öneri verilmiş ve Aktif

Tablo 1. Yazılım Bulgu Kriterleri

Daha sonra bölümlerin her bir yaygınlaştırma türünde TOPSIS yöntemine göre alacağı puanlar ile onaylanan tablo oranı değerleri VIKOR yönteminde kullanılarak genel kalite puanları elde edilecektir.

Kontrol Kriterleri
KAC: Kod Gözden Geçirme Acil Yaygınlaştırma
KAY: Kod Gözden Geçirme Aylık Yaygınlaştırma
KHA: Kod Gözden Geçirme Haftalık Yaygınlaştırma
KPI: Kod Gözden Geçirme Pilot Yaygınlaştırma
EAC: Ekran Gözden Geçirme Acil Yaygınlaştırma
EAY: Ekran Gözden Geçirme Aylık Yaygınlaştırma
EHA: Ekran Gözden Geçirme Haftalık Yaygınlaştırma

EPI: Ekran Gözden Geçirme Pilot Yaygınlaştırma

OTO: Onaylanan Tablo Oranı

Tablo 2. Yazılım Bulgularının Değerlendirildiği Kontrol Kriterleri

2.1 TOPSIS Yöntemi

TOPSIS yönteminin temeli pozitif ideal çözüme en kısa mesafe ve negatif ideal çözüme en uzak mesafedeki alternatifi seçmeye dayanır. TOPSIS yönteminin aşamaları aşağıda verilmiştir (Tzeng ve diğerleri, 2005):

1.Adım: Normalize edilmiş karar matrisinin hesaplanması. Normalize edilmiş r_{ij} değerleri matristeki değerler olan f_{ij} 'lere aşağıdaki işlem uygulanarak hesaplanır:

$$r_{ij} = f_{ij} / \sqrt{\sum_{j=1}^J f_{ij}^2} \quad j = 1, \dots, J; i = 1, \dots, n.$$

2.Adım: Ağırlıklı normalize edilmiş karar matrisinin hesaplanması. Ağırlıklı normalize edilmiş v_{ij} değerleri aşağıdaki gibi hesaplanır:

$$v_{ij} = w_i r_{ij}, j = 1, \dots, J; i = 1, \dots, n.$$

3.Adım: İdeal ve negatif ideal çözüm belirlenir:

$$A^* = \{v_1^*, \dots, v_n^*\} = \{(j \max v_{ij} | i \in I^+), (j \max v_{ij} | i \in I^-)\},$$

$$A^- = \{v_1^-, \dots, v_n^-\} = \{(j \min v_{ij} | i \in I^+), (j \min v_{ij} | i \in I^-)\},$$

Burada, I^+ fayda kriteri ve I^- maliyet kriteriyle ilişkilidir.

4. Adım: n boyutlu Öklid uzaklıklarını kullanarak pozitif ideal noktadan ve negatif ideal noktadan ayırım ölçümleri hesaplanır. Pozitif ideal noktadan ayırım aşağıdaki şekilde hesaplanır:

$$D_j^* = \sqrt{\sum_{i=1}^n (v_{ij} - v_i^*)^2}, j = 1, \dots, J.$$

Negatif ideal çözüm noktasından ayırım aşağıdaki şekilde hesaplanır:

$$D_j^- = \sqrt{\sum_{i=1}^n (v_{ij} - v_i^-)^2}, j = 1, \dots, J.$$

5.Adım: İdeal çözüm noktasına göreli yakınlık hesaplanır ve elde edilen alternatiflerin C_j^* değerleri büyükten küçüğe sıralanır.

$$C_j^* = D_j^- / (D_j^* + D_j^-), j = 1, \dots, J.$$

2.2 VIKOR Yöntemi

Vikor yönteminin adımları şu şekilde özetlenebilir (Ertuğrul & Karakaşoğlu, 2008):

1.Adım: Her bir kriter için en iyi (f_i^*) ve en kötü (f_i^-) değerleri belirlenir. Eğer kriter “i” fayda kriteri ise;

$$f_i^* = \max_j f_{ij} \quad f_i^- = \min_j f_{ij} \quad i = 1, 2, \dots, n$$

2.Adım: S_j ve R_j değerleri $j = 1, 2, \dots, J$ için hesaplanır. S_j ve R_j değerleri j . alternatif için ortalama ve en kötü grup skorlarını gösterir.

$$S_j = \sum_{i=1}^n w_i (f_i^* - f_{ij}) / (f_i^* - f_i^-) \quad R_j = \max [w_i (f_i^* - f_{ij}) / (f_i^* - f_i^-)]$$

Burada w_i görelî önemleri gösteren kriter ağırlıklarını ifade etmektedir. Ağırlıklar toplamı 1'e eşit olacaktır.

3.Adım: Q_j değerleri tüm $j = 1, 2, \dots, J$ için belirlenir.

$$Q_j = v(S_j - S^*) / (S^- - S^*) + (1-v)(R_j - R^*) / (R^- - R^*)$$

Burada $S^* = \min_j S_j$, $S^- = \max_j S_j$, $R^* = \min_j R_j$, $R^- = \max_j R_j$ ve “v” değeri maksimum grup faydasını göstermektedir. Diğer bir ifade ile “(1-v)”karşıt görüş-tükilerin minimum pişmanlığının ağırlığını ifade etmektedir. Uzlaşma “çoğunluk oyu” ($v > 0,5$) ile, “konsensüs” ($v = 0,5$) ile veya “veto” ($v < 0,5$) ile sağlanabilir.

4.Adım: S , R ve Q değerleri küçükten büyüğe doğru sıralanarak alternatiflerin sıralaması belirlenir. Sonuç olarak üç sıralama listesi oluşturulur.

5.Adım: Eğer aşağıdaki iki koşul sağlanırsa en iyiyi Q (Minimum) değerine göre sıralayan alternatif a' uzlaştırıcı çözüm olarak önerilir.

C_1 : Kabul edilebilir avantaj

$$Q(a'') - Q(a') \geq DQ$$

Burada a'' değeri Q değerine göre sıralamada ikinci sırayı alan alternatiftir.

$$DQ = 1 / (J - 1); \quad J \text{ alternatif sayısını gösterir.}$$

C_2 : Karar vermede kabul edilebilir istikrar

Ayrıca alternatif a' , S ve / veya R değerlerine göre sıralanan en iyi alternatiftir. Bu uzlaşık çözüm karar verme sürecinde istikrarlıdır. Eğer bu iki durumdan bir tanesi sağlanmazsa uzlaşık çözüm şu şekilde önerilir:

Eğer C_2 durumu sağlanmıyorsa a' ve a'' alternatifleri; eğer C_1 durumu sağlanmıyorsa $a', a'', \dots, a^{(M)}$ alternatifleri ve değeri maksimum M için $Q(a^{(M)}) - Q(a') < DQ$ belirlenir. Q değerlerine göre sıralanan en iyi alternatif minimum Q değerine sahip alternatiflerden biridir (Ertuğrul & Karakaşoğlu, 2008; Opricovic ve Tzeng, 2004).

3 Uygulama

Uygulamanın birinci aşamasında kullanılan TOPSIS yönteminde, yazılım şirketindeki yazılım geliştirme bölümleri alternatifleri oluşturmaktadır. Yazılım kalite değerlendirmesi yapılırken 3 aylık süre içerisinde bu bölümlerin ürettikleri yazılımlar baz alınmıştır. Tablo 1.'deki yazılım bulgu çeşitleri kriterleri meydana getirmektedir. Tablo 2.'de bulunan toplam 8 adet kontrol kriteri için uygulama geliştirme bölümlerinin TOPSIS puanları hesaplanmıştır. Tablo 3.'te kalite karşılaştırması için 10 adet yazılım bölümü ele alınmıştır ve 12 adet bulgu kriteri yer almaktadır. Bulgu kriterlerinin ağırlıkları bulguların karşılaştırmalı önem düzeylerine göre basit ağırlıklı ortalamaya ile elde edilmiştir. TOPSIS yönteminin gösterimi amacıyla "Kod Gözden Geçirme Acil Yaygınlaştırma" ve "Ekran Gözden Geçirme Aylık yaygınlaştırma" kontrol kriterleri için TOPSIS puanları elde edilmiştir. Diğer 6 adet kontrol kriteri için uygulama geliştirme bölümlerinin puanları aynı şekilde hesaplanmış olup, tüm değerler Tablo 11.'de verilmektedir. Bölümlerin bulgu kriterlerine göre TOPSIS yöntemi kullanılarak değerlendirilmesi aşamasında bulgulara verilen ağırlıklar ile bölümlerin 3 aylık süre içerisinde kod gözden geçirme faaliyetleri için acil olarak yaygınlaştırdıkları nesne başına düşen bulgu sayıları aşağıdaki tabloda verilmiştir. "0" olan değerler belirtilen bulgu türüne ait bulgunun oluşmadığı anlamına gelmektedir.

Uygulamanın ikinci aşamasında alternatiflere (bölümlere) ait TOPSIS puanları VIKOR yöntemindeki kontrol kriterleri alanına yerleştirilmiş ve nihai sıralama oluşturulmuştur. Aşağıdaki Tablo 3., Tablo 4. ve Tablo 5.'te adım adım uygulama sonuçları gösterilmektedir.

Ağırlıklar	0,05	0,15	0,20	0,05	0,10	0,15	0,05	0,05	0,05	0,05	0,05	0,05
	KD	KE	KA	RD	RE	RA	UD	UE	UA	ÖD	ÖE	ÖA
A1	0	0	0	0	0,500	0	0,500	0	0	0	0	0
A2	0,125	0	0	0,250	0,250	0	0,250	0	0	0	0	0
A3	0,118	0	0	0,412	0,118	0	0,059	0,059	0	0	0	0
A4	0	0	0	0,429	0	0	0,286	0	0	0	0	0
A5	0	0	0	0,250	0	0,250	0	0	0,500	0	0	0
A6	0	0	0	0	0,333	0	0	0,333	0	0	0,333	0
A7	0	0	0	0	1	0	0	0	0	0	0	0
A8	0	0	0	0	0	0	0	0	0	0	0	0
A9	0	0	0	0	0	0	0	1	0	0	0	0
A10	0	0	0	0	0	0	0	0	0	0	0	0

Tablo 3. Kod Gözden Geçirme Acil Yaygınlaştırmalar için bölümlerin yaygınlaştırılan nesne başına bulgu oranları

TOPSIS yönteminin 1. ve 2. adımlarında belirtilen işlemlerin gerçekleştirilmesi ile acil yaygınlaştırmalar için nesne başına bulgu oranlarının ağırlıklı normalize edilmiş değerleri elde edilir.

	KD	KE	KA	RD	RE	RA	UD	UE	UA	ÖD	ÖE	ÖA
A1	0	0	0	0	0,042	0	0,040	0	0	0	0	0
A2	0,036	0	0	0,018	0,021	0	0,020	0	0	0	0	0
A3	0,034	0	0	0,030	0,010	0	0,005	0,003	0	0	0	0
A4	0	0	0	0,031	0	0	0,023	0	0	0	0	0
A5	0	0	0	0,018	0	0,150	0	0	0,050	0	0	0
A6	0	0	0	0	0,028	0	0	0,016	0	0	0	0
A7	0	0	0	0	0,083	0	0	0	0	0	0	0
A8	0	0	0	0	0	0	0	0	0	0	0	0
A9	0	0	0	0	0	0	0	0,047	0	0	0	0
A10	0	0	0	0	0	0	0	0	0	0	0	0

Tablo 4. Kod Gözden Geçirme Acil Yaygınlaştırmalar için nesne başına bulgu oranlarının ağırlıklı normalizasyon değerleri

TOSIS yönteminin 3. adımı uygulanarak tüm kriterlere göre aşağıdaki pozitif ve negatif ideal çözüm noktaları elde edilir.

Pozitif İdeal Çözüm	0	0	0	0	0	0	0	0	0	0	0	0
Negatif İdeal Çözüm	0,036	0	0	0,031	0,083	0,150	0,040	0,047	0,050	0	0	0

Tablo5. Kod Gözden Geçirme Acil Yaygınlaştırmalar için Pozitif ve Negatif İdeal Çözüm Değerleri

TOPSIS yönteminin 4. adımı uygulanarak S^* pozitif ideal çözüm noktasına uzaklık ve S^- negatif ideal çözüm noktasına uzaklık değerleri ile C^* değerleri hesaplanır. Daha sonra, C^* değerleri büyükten küçüğe doğru sıralanır. C^* değerlerinden 1 değerini alan veya 1'e en yakın değeri alan alternatifler en iyi alternatifleri oluşturmaktadır. Kod gözden geçirme faaliyetinde acil yaygınlaştırmalar alanında en iyi yazılım kalitesi gösteren bölümlerin A8 ve A10 bölümleri olduğu görülmektedir.

	S^*	S^-	C^*	Sıralama
A1	0,0575	0,1768	0,7545	8
A2	0,0498	0,1781	0,7815	7
A3	0,0468	0,1834	0,7968	6
A4	0,0384	0,1892	0,8314	4
A5	0,1591	0,1107	0,4103	10
A6	0,0320	0,1815	0,8502	3
A7	0,0834	0,1764	0,6789	9
A8	0	0,1951	1	1
A9	0,0474	0,1892	0,7998	5
A10	0	0,1951	1	1

Tablo 6. Kod Gözden Geçirme Acil Yaygınlaştırmalar için bölümlerin C^* değerleri ile sıralamaları

Yukarıda yer alan işlemler VIKOR yönteminde kullanılacak her bir kriter için uygulanarak, bölümlerin TOPSIS puanları hesaplanır. VIKOR yönteminde kullanılacak bir diğer kriter olan “Ekran Gözden Geçirme Aylık Yaygınlaştırmalar” için kullanılan nesne başına bulgu oranları, ağırlıklı normalize değerler, pozitif ve negatif ideal çözüm noktalarına uzaklık değerleri ile C^* değerleri Tablo7-10 ile gösterilmiştir.

Ağırlıklar	0,05	0,15	0,20	0,05	0,10	0,15	0,05	0,05	0,05	0,05	0,05	0,05
	KD	KE	KA	RD	RE	RA	UD	UE	UA	ÖD	ÖE	ÖA
A1	0,200	0	0	0,650	0	0	0	0	0	0	0	0
A2	0,115	0	0	0,770	0,008	0	0,049	0	0	0	0	0
A3	0,200	0	0	0,600	0	0	0	0	0	0	0	0
A4	0,111	0	0	0,533	0,133	0	0,133	0	0	0,067	0	0
A5	0,118	0	0	0,294	0,235	0	0,059	0,118	0	0	0	0
A6	0,136	0	0	0,682	0	0	0,091	0	0	0	0	0
A7	0,077	0	0	0,577	0,154	0	0,038	0	0	0	0	0
A8	0	0	0	0,286	0	0	0,286	0	0	0,286	0	0
A9	0,167	0	0	0,556	0,111	0	0	0	0	0	0	0
A10	0,200	0	0	0,600	0	0	0,200	0	0	0	0	0

Tablo 7. Ekran Gözden Geçirme Aylık Yaygınlaştırmalar için Bölümlerin Yaygınlaştırılan Nesne Başına Bulgu Oranları

	KD	KE	KA	RD	RE	RA	UD	UE	UA	ÖD	ÖE	ÖA
A1	0,022	0	0	0,018	0	0	0	0	0	0	0	0
A2	0,012	0	0	0,021	0,002	0	0,006	0	0	0	0	0

A3	0,022	0	0	0,017	0	0	0	0	0	0	0	0
A4	0,012	0	0	0,015	0,040	0	0,017	0	0	0,011	0	0
A5	0,013	0	0	0,008	0,071	0	0,007	0,050	0	0	0	0
A6	0,015	0	0	0,019	0	0	0,012	0	0	0	0	0
A7	0,008	0	0	0,016	0,047	0	0,005	0	0	0	0	0
A8	0	0	0	0,008	0	0	0,036	0	0	0,049	0	0
A9	0,018	0	0	0,015	0,034	0	0	0	0	0	0	0
A10	0,022	0	0	0,017	0	0	0,025	0	0	0	0	0

Tablo 8. Ekran Gözden Geçirme Aylık Yaygınlaştırmalar için nesne başına bulgu oranlarının ağırlıklı normalizasyon değerleri

Pozitif İdeal Çözüm	0	0	0	0,008	0	0	0	0	0	0	0	0
Negatif İdeal Çözüm	0,022	0	0	0,021	0,071	0	0,036	0,050	0	0,049	0	0

Tablo 9. Ekran Gözden Geçirme Aylık Yaygınlaştırmalar için Pozitif ve Negatif İdeal Çözüm Değerleri

	S^*	S^-	C^*	Sıralama
A1	0,024	0,106	0,816	4
A2	0,019	0,103	0,841	1
A3	0,023	0,106	0,819	3
A4	0,047	0,073	0,607	8
A5	0,088	0,059	0,400	10
A6	0,022	0,103	0,826	2
A7	0,048	0,082	0,629	7
A8	0,061	0,091	0,599	9
A9	0,039	0,087	0,692	6
A10	0,035	0,1	0,744	5

Tablo 10. Ekran Gözden Geçirme Aylık Yaygınlaştırmalar için bölümlerin C^* değerleri ile sıralamaları

Tablo 2.'de belirtilen VIKOR kriterleri için bölümlerin elde ettikleri puanlar Tablo 11. ile verilmiştir.

Alternatifler	Kriterler								
	KAC	KAY	KHA	KPI	EAC	EAY	EHA	EPI	OTO
A1	0,754	0,449	0,422	0,468	1	0,816	0,583	0,373	0,709

A2	0,781	0,766	0,634	1	1	0,841	0,685	1	0,885
A3	0,797	0,949	0,955	0,742	0,500	0,819	0,855	0,738	0,832
A4	0,831	0,802	0,819	0,615	1	0,607	0,857	0,634	0,875
A5	0,410	0,808	0,852	1	1	0,400	0,668	1	0,847
A6	0,850	0,959	0,964	1	1	0,826	0,555	1	0,928
A7	0,679	0,549	0,590	0,739	1	0,629	0,707	1	0,871
A8	1	0,916	0,838	1	1	0,599	0,415	1	0,883
A9	0,800	0,622	0,727	1	0,500	0,692	0,516	1	0,842
A10	1	0,910	0,919	0,700	1	0,744	0,751	1	0,833

Tablo 11. VIKOR Yöntemi için kullanılan C^* değerleri

	KAC	KAY	KHA	KPI	EAC	EAY	EHA	EPI	OTO
Maksimum Değer (Maksimum ideal nokta)	1	0,96	0,96	1	1	0,84	0,86	1	0,93
Minimum Değer (Minimum ideal nokta)	0,41	0,45	0,42	0,47	0,50	0,40	0,42	0,37	0,71
Ağırlıklar	0,2	0,2	0,1	0,05	0,1	0,05	0,05	0,05	0,2

Tablo 12. VIKOR Yönteminde Kriterler için Minimum ve Maksimum İdeal Noktalar

Aşağıda verilen Tablo 13 ile alternatiflerin sonuç olarak aldıkları S_j , R_j Q_j değerleri gösterilmektedir.

	S_j	R_j	Q_j	v
A1	0,72	0,20	1	0,7
A2	0,27	0,08	0,25	0,7
A3	0,31	0,10	0,35	0,7
A4	0,29	0,06	0,24	0,7
A5	0,42	0,20	0,67	0,7
A6	0,09	0,05	0,00	0,7
A7	0,46	0,16	0,63	0,7
A8	0,16	0,05	0,08	0,7
A9	0,48	0,13	0,60	0,7
A10	0,16	0,09	0,16	0,7

Tablo 13. Bölümlerin S_j , R_j ve Q_j değerleri

4 Sonuç ve Öneriler

Yazılım kalitesinin değerlendirilmesi için pek çok yazılım kalite karakteristiği belirlenmiştir. Jørgensen'e (1999) göre kalite faktörlerinin bir kümesi olarak etkinlik, esneklik, entegre edilebilirlik, birlikte çalışabilirlik, sürdürülebilirlik değerlendirilebilir. Bir diğer yazılım kalite değerlendirme yöntemi kullanıcı memnuniyeti yaklaşımıdır. Bu iki yaklaşıma göre yazılım kalite değerlendirmesinin yapılabilmesi, ölçümü nesnel kriterlere dayandırma bakımından güçlükler yaratmaktadır. Ayrıca, yazılım kalite değerlendirmesinin anlamlı olabilmesi için yazılım geliştiren bölümler arasında karşılaştırma yapma gereksinimi doğmaktadır. Bu sebeple, bu çalışmada Jørgensen'in çalışmasından tamamen farklı olarak nesnel kalite ölçüm kriterleri kullanılmıştır.

Bu sebeple, yazılan kod satırına, yazılım büyüklüğüne, nesne başına göre hata sayılarının, çeşitlerinin ve önem düzeylerinin tespit edilerek yazılım kalitesinin değerlendirilmesi farklı alternatifleri kıyaslamamızı sağlarken, nesnel bir ölçüm de sağlamaktadır. Yazılım hata tipleri üzerinden gidilerek, alternatifler farklı iş seçenekleri ve değerlendirme yöntemlerine (yaygınlaştırma türleri ve kod/ekran gözden geçirme) göre karşılaştırılmıştır. TOPSIS ve VIKOR yöntemleri entegre bir şekilde kullanılmıştır. Bunun sonucunda Tablo 14.'te belirtildiği üzere **A6** ve **A8** bölümleri en iyi yazılım kalitesine sahip bölümler olarak bulunmuş olup; **A3**, **A7** ve **A1** bölümlerinin sonuçlarının istikrarlı olduğu tespit edilmiştir. **A1** alternatifinin en düşük yazılım kalitesine sahip olduğu görülmektedir. VIKOR yönteminde kullanılan ν değeri 0,7 olarak alınmıştır. Bu değer [0-1] aralığında değiştirilerek, uzlaşmanın çoğunluk oyu, veto veya uzlaşma olarak gerçekleştirilmesi sağlanabilir.

	S_j	R_j	Q_j	ν	Sıralama S_j	Sıralama R_j	Sıralama Q_j
A1	0,72	0,20	1	0,7	A6	A8	A6
A2	0,27	0,08	0,25	0,7	A8	A6	A8
A3	0,31	0,10	0,35	0,7	A10	A4	A10
A4	0,29	0,06	0,24	0,7	A2	A2	A4
A5	0,42	0,20	0,67	0,7	A4	A10	A2
A6	0,09	0,05	0,00	0,7	A3	A3	A3
A7	0,46	0,16	0,63	0,7	A5	A9	A9
A8	0,16	0,05	0,08	0,7	A7	A7	A7
A9	0,48	0,13	0,60	0,7	A9	A5	A5
A10	0,16	0,09	0,16	0,7	A1	A1	A1

Tablo 14. Bölümlerin VIKOR Yöntemi ile Sıralama Sonuçları

*** Yazılım kalite kontrol faaliyetlerini anlamama ve verileri temin etmeme katkılarından dolayı Sn. Nihan Akkuş'a teşekkürlerimi sunuyorum.

Referanslar

1. Demireli, E., 2010, "Topsis Çok Kriterli Karar Verme Sistemi: Türkiye'deki Kamu Bankaları Üzerine Bir Uygulama", *Journal of Entrepreneurship and Development*, (5:1).
2. Jørgensen, M., 1999, "Software Quality Measurement", *Advances in Engineering Software*, Vol. 30, pp.907-912.
3. Karakaşoğlu, N., Ertuğrul, İ., 2009, "Banka Şube Performanslarının VIKOR Yöntemi ile Değerlendirilmesi", *Endüstri Mühendisliği Dergisi*, YA-EM 2008 Özel Sayısı, Cilt:20, Sayı:1, Sayfa: (19-28).
4. Sarrab, M. & Rehman, O.M.H., 2014, "Empirical Study of Open Source Software Selection for Adoption Based on Software Quality Characteristics", *Advances in Engineering Software*, Vol. 69, pp. 1-11.
5. Tzeng, G.H., Lin, C.W.& Opricovic, S., 2005, "Multicriteria Analysis of Alternative Fuel Buses for Public Transportation", *Energy Policy*, Vol.33, pp. 1373-1383.
6. Ustasüleyman, T., 2009, "Bankacılık Sektöründe Hizmet Kalitesinin Değerlendirilmesi: Ahs-Topsis Yöntemi", *Bankacılar Dergisi*, Sayı 69.