

Bir Yazılımın Çevre ve Kalite Açısından Değerlendirilmesi için Önerilmiş Karar Alma Modeli

Sedef Akınlı Koçak¹, Gülfem Işıklar Alptekin², Ayşe Başar Bener³

¹ Data Science Laboratory, Environmental Applied Science, Ryerson University, Canada, Sedef.akinlikocak@ryerson.ca

² Bilgisayar Mühendisliği, Galatasaray Üniversitesi, İstanbul, Türkiye, gisiklar@gsu.edu.tr

³ Data Science Laboratory, Mechanical and Industrial Engineering, Ryerson University, Canada, Ayse.bener@ryerson.ca

Özet. Yeşil yazılım geliştirme, yeşil bilgi teknolojileri alanında yeni sayılabilecek bir çalışma alanıdır. Yazılım geliştirme endüstrisi artık, çevresel etmenleri daha fazla göz önünde bulundurarak yazılım ürünleri ortaya koymaktadır. Yeşil yazılıma duyulan bu ilgi, bir yazılımın sahip olması gereken çevresel özellikleri de, yazılımın ana kalite özelliklerinin içine alınması fikrini doğurmuştur. Bu çalışmada, akademik yazında sıklıkla kullanılan çok ölçütlü karar alma yöntemlerinden biri olan Analitik Ağ Süreci kullanılarak, yazılımın çevresel sürdürülebilirlik ve kalite kriterlerinin bir arada değerlendirilmesi yapılmıştır. Çalışmanın amacı, bilinen kalite kriterleri ile yeni ortaya çıkan çevresel kriterler arasındaki ilişkiyi göstermek ve kriterlerin öncelik sırasını belirlemektir. Elde edilen sonuçlar, bir yazılımı ‘yeşil’ olarak geliştirmede kullanılabilmesi gibi, kriterlerin hesaplanan ağırlıkları kullanılarak birbiri ile çelişen kalite ve çevresel kriterler arasındaki seçimde de yol gösterici olacaktır. Sonuçlar ayrıca, enerji harcamak zorunda olan bir çeşit ürünün, mümkün olan en üst seviyelerde enerji tasarrufu sağlayacak şekilde geliştirilmesinde yardımcı olabilecektir.

Anahtar Kelimeler: Yeşil yazılım; enerji verimliliği; yazılım geliştirme

1 Giriş

Yazılımların hayatımızdaki önemi ile birlikte, yazılım ürünlerinin sürdürülebilirliği ve çevresel etkilerinin önemi de gittikçe artmaktadır. Kullanıcıların gereksinimlerini karşılayacak kalitede yeşil yazılım ürünleri geliştirmek veya mevcut ürünleri çevresel etkilerini (örneğin enerji harcama) azaltacak şekilde yeniden düzenlemek karmaşık bir problemdir. Çevresel sürdürülebilirliğe ulaşmayı hedefleyen yeşil bilgi ve iletişim teknolojileri ve yeşil yazılım kalitesinin değerlendirilmesi son zamanlarda önemli araştırma konuları arasına girmiştir. Yeşil ve sürdürülebilir yazılım kavramı akademik yazında ilk defa 2011 yılında Neumann et al. (2011) tarafından şu şekilde tanımlanmıştır: “Yeşil ve sürdürülebilir yazılım, yazılımı geliştirme, piyasaya sürme ve kullanma aşamalarında ekonomiye, sosyal hayata, insanoğluna ve çevreye doğru-

dan ve dolaylı olarak en az şekilde olumsuz etkisi olan ve/veya sürdürülebilir kalkınmaya olumlu katkıda bulunan yazılımdır.” (Neumann et al., 2011, pp:296). Bu bağlamda, geleneksel ürün kalite modelleriyle belirlenen gereksinimlerin yanında çevresel ve sürdürülebilir gelişmeye etkileri gözetilen yeşil gereksinimlerin de ürün geliştirme aşamasında değerlendirilme gerekliliği ortaya çıkmıştır.

Yazılım geliştiren firmalar rakiplerinden daha kaliteli, daha fazla ve daha iyi işlevsellik içeren ürünler geliştirmek için yoğun bir baskı altındadırlar. Ancak yazılımın kalite gereksinimleri (örneğin performans, kullanılabilirlik, güvenlik, vb.) ve fonksiyonel özellikleri çoğu zaman birbirleriyle çelişmektedirler. Yazılım geliştiriciler, bu çelişkiler ışığında kalite ve fonksiyonellik arasında seçim yapmak zorunda kalabilecek ve/veya bunları bir öncelik sırasına koyarak çözüm bulma yoluna gidecektir. Pek çok açıdan bakıldığında, fonksiyonel gereksinimler ve kalite gereksinimleri birbirlerini tamamlayıcı ve bağlayıcı nitelik taşımaktadırlar. Aralarındaki bu ilişkiden dolayı seçim kararı yazılım geliştiriciler açısından karmaşık olmaktadır (Tiwari et al., 1996).

Buna ek olarak, değişen çevre şartları sürdürülebilirlik kaygılarını arttırmış ve ürün geliştirme ve ürün kalitesine çevresel (yeşil) gereksinimlerin eklenmesini zorunlu bırakmıştır. Sözkonusu yeşil gereksinimler, çoğunlukla, fonksiyonel gereklilikler ve kalite gereklilikleri arasında uyumsuzluk yaratmaktadır. Bunlar arasındaki ilişkiler daha gizli ve birbirlerine etkileri daha az anlaşılabilir. Bu nedenle, çevresel sürdürülebilirliğe katkısı olacak yeşil bir yazılım geliştirme sırasındaki gerekliliklerin belirlenmesinin önemli olduğuna inanıyoruz. Akademik yazında yazılım gereksinimlerinin önceliklendirilmesine (Karlsson et al., 1998) ve yazılım kalite gereksinimlerinin arasında seçim yapılmasına (tradeoff-ödüneşim) (Kazman et al., 2000, Chang et al., 2008) ilişkin yaklaşım ve modeller vardır. Bu çalışmalarda çok kriterli karar verme tekniklerinden biri olan Analitik Hiyerarşi Prosesi (AHP), Analitik Ağ Süreci (ANP) ve ANP/AHP birleştirilmiş yöntemler, hedef modelleme (goal modeling), ikili arama ağacı yaratma (binary search tree creation) ve fırsatçı (greedy) algoritmalar en çok kullanılanlardır. Bunlardan başka, yazılım mimarisi seçiminde karar verme süreci için bazı ayrıntılı ödüneşim teknikleri de kullanılmıştır (Zhu et al., 2005). Yazılım mühendisliği karar verme süreçlerini destekleyen, en önemli yöntemlerin başında Ruhe et al. tarafından geliştirilen teknik yer almaktadır (Ruhe et al., 2003). Bu yöntemde, gereksinim önceliklendirilmesi müşteri memnuniyeti, kaynak dağılımı ve iş değeri/maliyeti gibi etmenlere göre yapılmaktadır (Ruhe et al 2003, Ruhe/Ngo-The 2009). Önceliklendirmeyi yaparken iteratif ve genel bir algoritmaya sahip olan bir optimizasyon metodu kullanılmaktadır.

Akademik yazında şimdiye kadar yapılan çalışmalar incelendiğinde, çevresel gereksinimler ile kalite gereksinimlerinin arasındaki ilişkiye odaklanan ve yeşil yazılım geliştirme sürecinde karar vermenin incelendiği bir çalışmaya rastlanmamıştır. Bu çalışmada, kriterler arasındaki ilişkiyi/ bağıntıyı oldukça iyi gösteren çok kriterli karar verme tekniklerinden biri olan Analitik Ağ Prosesi (ANP) kullanılmış ve gereksinimlerin öncelik sırası ortaya konmuştur. Çok ölçütlü karar verme yöntemleri, kriterler arasındaki ilişkiyi belirlemekle kalmayıp, ağırlıklarını da ortaya koymakta; böylece karar kalitesini arttırmaktadırlar (Campanella, Ribeiro, 2011).

Çalışmanın geri kalan kısmı şu şekilde düzenlenmiştir. İkinci bölümde akademik yazın araştırması kapsamında yeşil ve sürdürülebilir yazılım kavramı, çevresel kriterler beraberinde yeşil ölçütler, yazılım kalite modelleri ve ANP yöntemi anlatılmıştır. Üçüncü bölümde ANP yöntemi hakkında bilgi verilmiştir. Dördüncü bölümde, önerilen yöntem ve ANP uygulama sonuçları verilmiştir. Beşinci ve son bölümde ise sonuçlar ve tartışma bölümüne yer verilmiş ve çalışmanın teorik ve pratik katkıları anlatılmıştır.

2 İlgili Akademik Yazın

2.1 Yeşil ve Sürdürülebilir Yazılım

Akademik yazında yazılımın sürdürülebilirliği hakkındaki çalışmalar öncelikle donanımın enerji etkinliğinin ölçülmesi, enerji etkin yazılım mimarileri ve/veya optimizasyon algoritmaları ile başlamıştır (Kothiyal et al., 2009) (Bhattacharjee et al., 2009). Sürdürülebilirlik ve yazılım mühendisliği kavramlarının kullanıldığı ve yeşil yazılım geliştirmede referans teşkil eden ilk çalışma “GreenSoft” referans modeliyle Neumann et al. (2011) tarafından yapılmıştır. Bu model yazılım ürününün yaşam döngüsü, sürdürülebilirlik kriterleri ve ölçütleri, prosedürleri ve bir yazılım ürününün çevreye ve sürdürülebilirliğe olumlu katkı sağlayacak şekilde kullanılması gerektiği hakkında öneriler ve kullanılacak araçları ele almıştır. Bir diğer çalışmada Alberto et al. (2010), çok kullanılan yazılım kalite kriterlerini (etkinlik, performans, uygunluk vb.) sürdürülebilirliğin boyutları (ekonomik, sosyal ve çevresel) ile ilişkilendirmiştir. Ancak bu çalışmada her bir kalite kriterinin sürdürülebilirlik boyutlarına nasıl bir bireysel etki yapacağı irdelenmemiştir. Bir diğer çalışma, sürdürülebilirliği yazılım ürün kalitesini etkileyen yepyeni bir etmen olarak kabul etmiş ve sürdürülebilirliğin yazılım kalite standartlarının içerisine dahil edilmesi gerekliliğini ortaya koymuştur (Calero ve Barteo, 2013). Diğer bir yaklaşımda, yeşil ve sürdürülebilir yazılım için bir kalite modeli geliştirilmiştir (Kern et al., 2013). Bu model, doğrudan ve dolaylı olarak yazılımın sürdürülebilirliği ile ilişkili tüm kriterleri içermektedir. Çalışmada ayrıca etkin bir karar verme süreci için kalite kriterlerinin özelleştirilmesi gerekliliğinin de üzerinde durulmuştur. Bu gereklilik bizim çalışmamızın da ana motivasyonunu oluşturmaktadır. Çalışmamızda yazılım kalite ve çevresel sürdürülebilirlik kriterlerinin aralarındaki bağlantıyı göstererek, bu ilişkilerin karar verme sürecinde kullanılması amaçlanmıştır.

2.2 Yeşil Ölçütler

Yeşil ölçütler BT sistemlerinin yeşil performanslarının ölçülmesinde kullanılan ölçütler olarak akademik yazındaki yerini almıştır. Kipp et al., (2011)’e göre, yeşil ölçütler, sistemlerin enerji harcamalarıyla ilişkili ölçütlerdir. Yazarlara göre bu ölçütler BT sistemlerinin ‘yeşillik’ oranının belirlenmesinde kilit etmen olarak da düşünülmektedir. Diğer yandan, enerji harcama, enerji etkinliği ve enerji tasarrufu olasılıklarını belirtmektedir. Çalışmada yeşil performans belirleyicileri (green performance indicators) göz önüne alınarak, ölçütler dört sınıfa ayrılmıştır (Tablo 1):

- **BT kaynak kullanım ölçütleri:** BT sisteminin enerji kullanımı ile yakından ilgilidir.
- **Yaşam döngüsü ölçütleri:** Ürünün yaşam döngüsünde enerji kullanımını takip etmek için geliştirilmiştir.
- **Enerji etki ölçütleri:** Ürünün yaşam döngüsündeki elektrik harcama, güç, kullanılan materyal ve karbondioksit salınımı gibi çevreye olan etkisini görmek için geliştirilmiştir.
- **Organizasyonel ölçütler:** Enerji bağlantılı maliyetlerin hesaplanması ve takibi için geliştirilmiştir.

Tablo 1. Yeşil ölçüt sınıfları (Kipp et al., 2011, Mahmoud ve Ahmad, 2012)

BT kaynak kullanım	Yaşam döngüsü	Enerji etkisi	Organizasyonel
- BT ekipman kullanımı/etkinliği	- Yaşam döngü maliyeti	- Güç enerjisi (yeşil enerji kullanımı)	- Kanun ve regülasyonlar
- BT ekipman saklama alanı etkinliği	- Süreç mühendisliği	- Isı ve nem	- Kaynak çabaları
	- Servis kalitesi	- Karbondioksit salınımı	- Sera gazları kredisi
			- BT'nin toplam maliyeti
			- Yeşil çözümler (Yeşil yatırım geri dönüş maliyeti)

2.3 Yazılım Kalite Modelleri

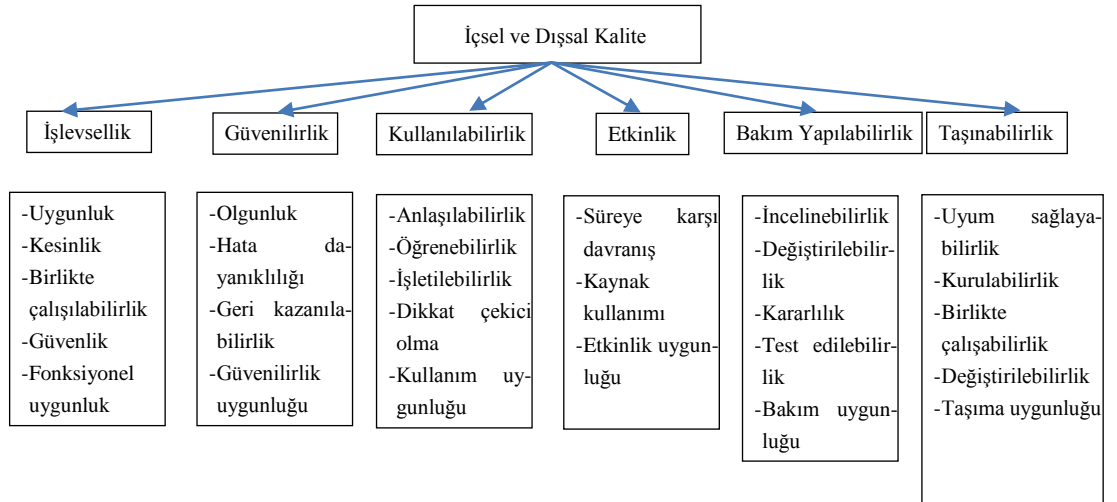
Yazılım kalitesi, IEEE Std. 1061, (1998) standardına göre “yazılımın istenilen kalite kriterlerinin kombinasyonlarına sahip olması” olarak tanımlanmıştır. Yazılım kalitesinin belirlenmesi için bu tanımlamada geçen, istenilen kalite kriterleri kombinasyonunun açık bir şekilde belirlenmesi gerekmektedir. Yazılım geliştirme projelerindeki, belirli özelliklerin belirlenen zaman içinde ve belirlenen maliyetin altında olması istendiğinden dolayı yöneticiler kalite kriterleri arasında seçim yapmak zorunda kalmaktadırlar. Bu nedenle, kalite modelleri hem bu seçimin yapılmasında yardımcı olmakta, hem de kullanım kolaylığı sağlamaktadır.

Akademik yazında pek çok kalite modeli geliştirilmiştir. En çok kullanılanları Tablo 2’de gösterilmiştir (Marciniak 2003, Berander et al., 2005).

McCall (1977) ve Boehm et al. (1978) modelini esas alarak geliştirilen ISO/IEC 9126 (ISO/IEC 9126-1, 2001) en yeni modellerden biridir. Bu kalite standart modeli, içsel ölçütler, dışsal ölçütler ve ürün kullanım kalitesi olmak üzere dört bölüm içermektedir. ISO 9126-1, son kullanıcı kullanım kalitesine yönelik dört etmen (fonksiyonellik, kullanılabilirlik, etkinlik ve güvenilirlik) ve geliştiriciye yönelik iki etmen (bakım ve taşınabilirlik) içermektedir (Şekil 1).

Tablo 2. Akademik yazındaki kalite modelleri ve kriterleri

(McCall, 1977)	(Boehm et al., 1978)	(Bowen et al., 1985)	(Murine, 1983)	Diğerleri
Doğruluk	Güvenilirlik	Doğruluk	Doğruluk	Doğruluk
Güvenilirlik	Etkinlik	Güvenilirlik	Güvenilirlik	Güvenilirlik
Etkinlik	Anlaşılabilirlik	Etkinlik	Etkinlik	Etkinlik
Kullanılabilirlik	Değiştirilebilirlik	Kullanılabilirlik	Kullanılabilirlik	Kullanılabilirlik
Bütünlük	Test edilebilirlik	Bütünlük	Bütünlük	Bütünlük
Değiştirilebilirlik	Taşınabilirlik	Değiştirilebilirlik	Değiştirilebilirlik	Değiştirilebilirlik
Esneklik		Esneklik	Esneklik	Esneklik
Test edilebilirlik		Çeşitlilik	Test edilebilirlik	Test edilebilirlik
Taşınabilirlik		Taşınabilirlik	Taşınabilirlik	Taşınabilirlik
Yeniden kullanılabilirlik		Yeniden kullanılabilirlik	Yeniden kullanılabilirlik	Yeniden kullanılabilirlik
Birlikte çalışabilirlik		Birlikte çalışabilirlik	Birlikte çalışabilirlik	Birlikte çalışabilirlik
		Kalımlılık		Kalımlılık
		Genişletilebilirlik		Güvenlilik
				Yönetilebilirlik
				İşlevsellik
				Desteklenebilirlik



Şek. 1. ISO 9126-1 yazılım kalite etmenleri

3 Analitik Ağ Süreci (ANP)

ANP yaklaşımı, 1980 yılında Saaty tarafından önerilen Analitik Hiyerarşi Süreci'nin genelleştirilmiş halidir (Saaty, 1980) (Saaty, 2004). ANP problemleri, kriterlerin aralarındaki ilişkileri ve bu ilişkilerin yönlerini tanımlayarak, problemi bir ağ şeklinde tasarlamayı sağlar. Oysa AHP yaklaşımında kriterler birbirlerine tek yönden bağlı varsayılır. ANP ve AHP arasındaki en önemli farklardan biri de hiyerarşik yapılarıdır. AHP'de yukarıdan aşağı doğru bir hiyerarşik yapı söz konusuysen, ANP'de etkileşimli bir hiyerarşik yapı vardır. Kriterler diğer kriterlere bağımlı olabildikleri gibi, kendi içlerinde de bağımlı olabilirler.

ANP altyapısında üç önemli adım vardır: (1) Amacı, kriterleri ve alt kriterleri belirlemek (2) Aradaki bağıntıları ve ağı belirlemek, ve (3) Süpermatrisi oluşturmak (Gürbüz et al., 2012). AHP'de de yapılan ikili karşılaştırmalar, bir elemanın bir diğer eleman üzerindeki göreceli önemini belirtmeye yarayan Saaty'nin 1-9 ölçeği esas alınarak yapılmıştır (Tablo 3) (Saaty, 2004). ANP yaklaşımı, bu karşılaştırma matrislerinin tutarlılığının kontrolünü şart kılar (Saaty 1980). Ardından, kriterler ve alt kriterler arasındaki ilişkiler uyarında bağıl (composite) ağırlıklar hesaplanır. Bağıl ağırlıkları içeren süpermatris, her sütunu 1'e eşit olacak hale getirilerek ağırlıklandırılmış süpermatris haline getirilir. Son adımda ise, her seçeneğin (alternatifin) genel puanını hesaplamak için, her bir seçenek verilen kriterler ve alt kriterler uyarınca değerlendirilir. Bu çalışmada, iki alternatif olduğu varsayılmıştır: Yeşil yazılım ve yeşil olmayan yazılım. Çalışmada, ANP yaklaşımı yeşil ölçütlerin önem derecelerini (ağırlıklarını) hesaplamak için kullanılmıştır.

Tablo 3. Değerlendirme yapılırken kullanılan ölçek (Saaty, 2004)

Önem derecesi	Tanım
1	Eşit önemde
3	Biraz daha önemli
5	Daha önemli
7	Çok daha önemli
9	Son derece önemli

4 ANP Yaklaşımının Uygulanması

4.1 Model Tasarımı

Öznel kriterlerin önceliklendirme prosedürü dört adımdan meydana gelir:

1. Kriter ve alt kriterlerin belirlenmesi.
2. ANP modelinin oluşturulması.
3. Bir anket yardımıyla ikili karşılaştırmaların yaptırılması.
4. Kriter ve alt kriterler arasındaki bağıntıların incelenmesi.

Kriter ve Alt Kriterlerin Belirlenmesi.

Bir yazılım ürününün sürdürülebilirliği ile ilgili ana kriterler, ilgili akademik yazında verilene uygun olarak belirlenmiştir (Bölüm 2.3 ve Bölüm 2.4). Ardından, ana kriterlerle doğrudan ilgisi olan kriterler, alt kriterler olarak belirlenmiştir. ANP modelini kurmakta kullandığımız kriterler ve alt kriterlerin tamamı, Tablo 4’te özetlenmiştir. Çevresel sürdürülebilirlik kriterleri arasında yer alan enerji tüketimi ve CO₂ salınımı, aslında sayısal olarak ölçülebilen değerlerdir. Bunların anket yoluyla değerlendirilmesinin sebebi, bu kriterlerin yeşil bir yazılım geliştirilirkenki önceliğini/önemini sayısılaştırmaktır.

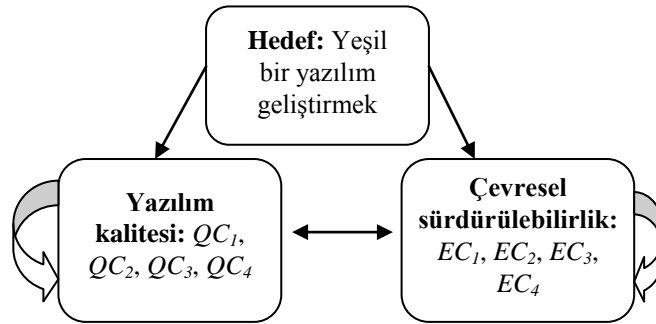
ANP Altyapısı.

Önerilen ANP modelinde, hedef, kriterler ve alt kriterler arasındaki ilişkiler Şekil 2’de gösterilmiştir. Önerilen karar modelinde iki seviye vardır. Hedef (yeşil ve sürdürülebilir bir yazılım geliştirmek) ilk seviyede yer alır. İkinci seviyede kriterler bulunmaktadır. Önerilen hiyerarşide, kriterler ve alt kriterler arasında içsel ve dışsal bağlantılar olduğu varsayılmış ve bunlar oklarla gösterilmiştir.

Tablo 4. Yazılım kalite ve sürdürülebilirlik kriterleri

Yazılım Kalite Kriterleri (QC) (ISO / 9126-1, 2001)	
<i>QC₁</i>	İşlevsellik
<i>QC₂</i>	Güvenilirlik
<i>QC₃</i>	Kullanılabilirlik
<i>QC₄</i>	Etkinlik

Çevresel Sürdürülebilirlik Kriterleri (EC) (Mahmoud ve Ahmad, 2012)	
<i>EC₁</i>	Enerji Tüketimi
<i>EC₂</i>	CO ₂ Salınımı
<i>EC₃</i>	Yeşil Enerji Kullanımı
<i>EC₄</i>	Yeşil Yatırımın Geri Dönüşü



Şek. 2. ANP’de kullanılan kriterler ve hedef

Anket ve İkili Karşılaştırmalar.

Önerilen modelde sekiz kriter bulunmaktadır. Her bir elemanın hedef üzerindeki etkisini belirleyebilmek için, bir anketten faydalanılmıştır. Her bir kriter ve alt kriter grubu için, ayrı ayrı matrisler oluşturulmuştur. Anketler, on farklı Türk yazılım şirketine çalışan, on karar verici tarafından doldurulmuş ve verdikleri puanların geometrik ortalaması alınmıştır (Saaty ve Vargas 2006).

Kriter ve Alt Kriterler Arasındaki İlişkilerin Analizi.

Son adım olarak, kriterlerin son ağırlıklarını hesaplayabilmek için, ANP'nin gerektirdiği matris işlemleri yapılmıştır. Bu işlemleri yapmak için Super Decisions adlı yazılım ürünü kullanılmıştır (Creative Decisions Foundation, 2013). Super Decision yazılımı, ANP hiyerarşik modelini oluşturup, matris işlemlerini yapabildiğimiz, kullanımı oldukça kolay bir araçtır. İkili karşılaştırma matrislerinin tutarlılık indislerinin hepsi 0.1'den düşük çıkmıştır; yani karşılaştırmaların tutarlı olduğu varsayılabilir.

Sonuçlar.

Tablo 5'teki gibi bir ikili karşılaştırma matrisinde, karar vericinin, yeşil ve sürdürülebilir bir yazılım geliştirirken, hangi kriter kümesinin (yazılım kalitesi ve sürdürülebilirlik kriterleri) daha önemli olduğunu değerlendirmesi beklenmektedir. Tablo 3'te verilen Saaty'nin ölçekleri kullanılmış ve karar vericilerin cevaplarının geometrik ortalaması alınarak tek bir değer oluşturulmuştur. Tablo 6(a, b) ise, sırasıyla, yazılım kalitesi ile ilgili alt kriterlerin geometrik ortalaması ve çevreyle ilgili alt kriterlerin geometrik ortalamasını vermektedir.

Tablo 5. İki ana kriter grubuna ait ikili karşılaştırma matrisi

	EC	QC
EC	1	6
QC	1/6	1

Tablo 6. (a) Kalite ile ilgili alt kriterlere ait ikili karşılaştırma matrisi (b) Çevreyle ilgili alt kriterlere ait ikili karşılaştırma matrisi

(a)	QC_1	QC_2	QC_3	QC_4
QC_1	1	1/3	1/2	1/2
QC_2	3	1	4	3
QC_3	2	1/4	1	1/3
QC_4	2	1/3	3	1
(b)	EC_1	EC_2	EC_3	EC_4
EC_1	1	2	2	1
EC_2	1/2	1	2	1/2
EC_3	1/2	1/2	1	1
EC_4	1	2	1	1

ANP yöntemi her bir karşılaştırma matrisine uygulanmış ve kriter ve alt kriterlerin öncelik sıraları bulunmuştur. Tablo 7 ve Şekil 3’de sonuçlar ve Tablo 8’de ise bu ağırlıklardan meydana gelen süpermatris yer almaktadır. Süpermatris, görelî öncelik vektörlerinden oluşmaktadır. Süpermatrisin bir sütunundaki ağırlık değerlerinin toplamı bire eşitlenecek şekilde normalize edilerek oluşturulan matrise, normalize edilmiş süpermatris denmektedir.

Tablo 7. Kriter ve alt kriterlerin ağırlıkları

Kriter	Ağırlık	Alt Kriter	Ağırlık
Kalite	0.14	İşlevsellik (F)	0.23
		Güvenilirlik (R)	0.24
		Kullanılabilirlik (U)	0.19
		Etkinlik (E)	0.34
Çevre	0.86	Enerji Tüketimi (EC)	0.35
		CO ₂ Salınımı (CO ₂)	0.24
		Yeşil Enerji Kullanımı (GEU)	0.18
		Yeşil Yatırımın Geri Dönüşü (ROGI)	0.23

Sonuçlara bakıldığında, çevresel sürdürülebilirlik kriteri, %86’lık ağırlığı ile, yeşil bir yazılım geliştirmek için en çok ihtiyaç duyulan kriter olmuştur. Bu kriterin dört alt kriterinden, enerji tüketimi %35 ile en yüksek ağırlığa sahip çıkmıştır. Bunu %24 ağırlık ile CO₂ salınımı ve %23 ağırlık ile yeşil yatırım geri dönüşümü izlemiştir. Kalitenin dört alt kriterinden en yüksek ağırlığa sahip olan %34 ile kaynakların etkin kullanılması olarak belirlenmiştir.

Tablo 8. Normalize edilmiş süpermatris

	EC	CO ₂	GEU	ROGI	E	F	R	U
EC	0	0.274	0.129	0.194	0.247	0.217	0.137	0.247
CO₂	0.156	0	0.164	0.085	0.116	0.098	0.213	0.117
GEU	0.098	0.105	0	0.222	0.068	0.088	0.085	0.068
ROGI	0.245	0.123	0.206	0	0.068	0.097	0.065	0.068
E	0.254	0.260	0.247	0.250	0	0.050	0.206	0.130
F	0.134	0.115	0.10	0.083	0.130	0	0.130	0.206
R	0.058	0.056	0.070	0.083	0.206	0.3	0	0.163
U	0.053	0.069	0.082	0.083	0.164	0.150	0.164	0

5 Sonuç ve İlerideki Çalışmalar

Yazılım şirketlerinin çoğu, geliştirme süreçlerini müşterilerin iş standartları ve taleplerine göre şekillendirmektedirler. Çevreyi önemseyen ürünler, günümüzde bu talepler arasındaki yerini almıştır. Çevreye duyarlı yazılım üretmek için, bazı kaynakların kullanımını azaltırken, bazılarıninkini arttırmak gerekebilmektedir. İşte bu noktada,

şirketler kullanıcıların talepleriyle, sürdürülebilirlik gereksinimleri arasında ödünleşim problemleriyle karşı karşıya kalmaktadırlar. Bu çalışmada, bu gereksinimleri önceliklendirmek için bir çok ölçütlü karar verme yaklaşımı olan ANP'den faydalanılmıştır. ANP uygularken, kalite ve çevresel kriterler arasında bağıntılar olduğu varsayılmıştır. Bunlar ışığında kriter ve alt kriterlerin ağırlıkları hesaplanmıştır. Geleneksel ödünleşimler, birbiriyle çelişen kalite kriterleri arasında olur. Çalışmada bu kriterlerin ağırlıklandırıldığından, ödünleşim problemlerinde bu değerler kullanılabilir ve bu sayede daha doğru kararlar alınabilir. Kullanılan bu yaklaşım, herhangi bir sektör veya faaliyet alanı için uygulanabilir.

Bundan sonraki adımda, önerilen bu altyapı gerçek bir yazılım şirketine ait gerçek verilerle çalıştırılacaktır. Şirkete ait bütçe, kaynak ve süre kısıtları göz önünde bulundurularak, bir hedef programlama tasarlanabilir .

Icon	Name	Normalized by Cluster	Limiting
No Icon	CO2 emissions	0.23857	0.119287
No Icon	Energy Consumption	0.34815	0.174076
No Icon	Green energy usage	0.18618	0.093092
No Icon	Return on green investment	0.22709	0.113545
No Icon	Develop Sustainable Software	0.00000	0.000000
No Icon	Efficiency	0.33905	0.169524
No Icon	Functionality	0.22735	0.113675
No Icon	Reliability	0.23619	0.118097
No Icon	Usability	0.19741	0.098703

Şek. 3. Kriterler ve alt kriterlerin ağırlıkları

Kaynaklar

1. Albertao, F., Xiao, J., Tian, C., Lu, Y., Zhang, K. Q., & Liu, C. (2010): Measuring the sustainability performance of software projects. In e-Business Engineering (ICEBE), 2010 IEEE 7th International Conference on (pp. 369-373). IEEE, pp. 369–373.
2. Berander, P., Damm, L., Eriksson, J., Gorschek, T., Henningson, K., Jonsson, P., Kagstrom, S., Milicic, D., Martensson, F., Renkko, K., and Tomaszewski, P. (2005): Software quality attributes and trade-offs. Blekinge Institute of Technology.
3. Bhattacharjee, B., Lim, L., Malkemus, T., Mihaila, G., Ross, K., Lau, S., McArthur, C., Toth, Z., Sherkat, R. (2009): Efficient index compression in DB2 LUW, In Proceedings of the VLDB Endow. 2(2), pp.1462-1473.
4. Boehm, B.W, Brown, J.R., Kaspar, J.R. (1978): Characteristics of Software Quality, TRW Series of Software Technology, Amsterdam, North Holland.
5. Bowen, T.P., Wigle, G.B., Tsai, and J.T. (1985): Specification of software quality attributes, Technical Report RADC-TR-85-37, Rome Air Development Center.

6. Calero C. and Bertoa, M.F. (2013): 25010+S: A software quality model with sustainable characteristics. Sustainability as an element of software quality. GIBSE 2013, Fukuoka, Japan.
7. Campanella, G., Ribeiro, R.A. (2011): A framework for dynamic multiple-criteria decision making. *Decision Support Systems*, 52, pp. 52–60.
8. Creative Decisions Foundation. (2013): Super Decisions (ver. 2.2.6) [Open source Software]. Retrieve February 2, 2013. Available from: <http://www.superdecisions.com>.
9. Gürbüz, T, Alptekin, S.E. and Işıklar Alptekin, G. (2012): A hybrid MCDM methodology for ERP selection problem with interacting criteria, *Decision Support Systems*, 54, pp. 206–214.
10. IEEE Std.061-1998 (1998): IEEE Standard for a SW Quality Metrics Meth., IEEE Computer Society.
11. ISO/IEC 9126-1 (2001): Information technology - Software quality characteristics and metrics - Part 1: Quality characteristics and subcharacteristics, ISO, Int. Electrotechnical Commission, Geneva.
12. Kazman, R., Klein, M., and Clements, P. (2000): ATAM: Method for architecture evaluation (No. CMU/SEI-2000-TR-004). Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst.
13. Karlsson, J., Wohlin, C., and Regnell, B. (1998): An evaluation of methods for prioritizing software requirements, *Information and Software Technology*, 39, pp. 939–947.
14. Kern, E. Dick, M., Naumann, S, Guldner, A, and Johann, T. (2013): Green Software and Green Software Engineering – Definitions, Measurements, and Quality Aspects, In *Proceeding of the First International Conference on ICR for Sustainability (Zurich)*, pp. 87-94.
15. Kipp, A., Jiang, T., Fugini, M. (2011): Green metrics for energy-aware IT systems, *IEEE Computer Society*, 42, pp. 241-248.
16. Kothiyal, R., Tarasov, V., Sehgal, P., Zadok, E. (2009): Energy and performance evaluation of lossless file data compression on server systems, In *Proceedings of ACM SYSTOR 2009: The Israeli Experimental Systems Conference*, pp. 4-16.
17. Mahmoud, S., Ahmad, I. (2012): Green performance indicators for energy aware IT systems: Survey and assessment, in: *Journal of Green Engineering*, 3, pp. 33–69.
18. Marciniak, J.J. (2003): *Encyclopedia of Software Engineering*, 2003, Wiley.
19. McCall, J.A. (1994): *Quality Factors*, *Encyclopedia of Software Engineering*, John Wiley & Sons.

20. Murine, G.E. (1983): Improving management visibility through the use of software quality metrics, IEEE Computer Society's 7th International Computer Software & Applications Conference, pp. 638-639.
21. Naumann, S., Dick, M., Kern, E., Johann, T. (2011): The GREENSOFT Model: A reference model for green and sustainable software and its engineering, in: Sustainable Computing: Informatics and Systems, 1, pp. 294-304.
22. Ruhe G, Greer, D. (2003): Quantitative studies in software release planning under risk and resource constraints. In Proceedings of International Symposium on Empirical Software Engineering (ISESE), IEEE, Los Alamitos CA, pp. 262-271.
23. Ruhe, G. and Ngo-The, A. (2009): Optimized resource allocation for software release planning. Trade-off analysis for requirements selection. IEEE Transactions on Software Engineering, 35, pp.109–123.
24. Saaty, T.L. (1980): The Analytical Hierarchy Process. New York: McGraw Hill. Pittsburgh: RWS Publications, 4922 Ellsworth Avenue, Pittsburgh, PA 15213.
25. Saaty, T.L. (2004): Fundamental of the analytic network process-Dependence and feedback in DM with a single network. Journal of Systems Science and Systems Engineering, 13(2), pp. 129-157.
26. Saaty, T.L., and Vargas, L. (2006): Decision making with the Analytic Network Process: Economic, political, social and technological applications with benefits, opportunities, costs, risks. Springer, NY.
27. Tiwari, V., Malik, S., Wolfe, A., Tien-Chien Lee, M. (1996): Instruction Level Power Analysis and Optimization of Software. The Journal of VLSI Signal Processing, 13, pp.223–238.
28. Zhu, L., Aurum, A., Gorton, I., and Jeffery, R. (2005): Tradeoff and sensitivity analysis in software architecture evaluation using analytic hierarchy process, Software Quality Journal, 13(4), pp.357-375.