

R-COVER: Yazılım Büyüklük Ölçümü Hata Tespit Aracı

Gökçen Yılmaz¹, Seçkin Tunalılar^{1,2}, Onur Demirörs¹

¹Enformatik Enstitüsü, Bilişim Sistemleri Bölümü, ODTÜ, Ankara

²MGEO Grubu, Aselsan, Ankara

¹{ygokcen, demirors}@metu.edu.tr

²stunalilar@aselsan.com.tr

Özet. Maliyet ve bütçe tahminleri, süreç kıyaslama ve proje kontrolü gibi yazılım proje yönetimi aktivitelerinin pek çoğu yazılım işlevsel büyüklük ölçümlerine bağlı olduğu için bu değerlerin ölçümü ve güvenilirliği çok önemlidir. Bu sebeple, işlevsel büyüklük ölçümlerinin güvenilirliğini artırmak için, ölçüm sürecinin sonunda büyüklük dokümanları kontrol edilmeli ve gözden geçirilmelidir. Ancak, ölçümlerdeki hataları tespit etmek için yapılan ve kişilere bağlı olan manuel gözden geçirme yöntemi zaman ve emek alıcı bir işlemdir. Ayrıca ölçümlerde var olan hataları gözden kaçırma olasılığı da bulunmaktadır. Bu tür sorunları aşmak için, büyüklük dokümanlarını otomatik olarak kontrol eden bir araç geliştirilmesine ihtiyaç vardır. Bu amaçla, bu çalışmada günümüzün en çok kullanılan COSMIC İşlevsel Büyüklük Ölçüm yöntemi için bir yazılım aracı, R-COVER geliştirilmiştir. Bu makalede, bu aracın geliştirilme süreci ve aracın hataları doğru olarak tespit etme açısından verimliliğini gösteren vaka çalışmalarının ilk sonuçları sunulmuştur. Araç ilk olarak Yönetim Bilişim Sistemleri projeleri için geliştirilmiş olup, ileride yapılacak güncellemeler ile gerçek zamanlı ve gömülü sistem yazılımları gibi pek çok yazılım türüne yönelik kullanılabilir olacaktır.

Anahtar Kelimeler. İşlevsel Büyüklük Ölçümü, Güvenilirlik, Hata Kategorileri, Hata Yakalama, Hata Yakalama Aracı, COSMIC

1 Giriş

Yazılım proje aktiviteleri ve planları büyüklük ölçümlerine dayandığı için ölçümlerin güvenilir ve doğru olması çok kritiktir. Ölçüm sonuçlarının kalitesini artırmak için araştırmacılar, ölçüm sonuçlarındaki tekrarlanabilirliği arttıran yeni kılavuzların oluşturulması [8] [13], ölçüm yaklaşımını standartlaştırmak için standart ölçüm şablonlarının oluşturulması [14], ileride kullanmak için ölçüm sonuçlarının belgelenmesi [16], Yazılım gereksinim Dokümanlarının (YGD) kalitesinin artırılması [21], uzmanlar tarafından karşılıklı kontrol yöntemi ile ölçümlerin doğrulanması [15] [16] [17] ve çeşitli yaklaşımları entegre eden metodolojiler [15] gibi farklı ve değerli yaklaşımları önermektedirler. Tekrarlanabilir ve doğru ölçümlere ulaşabilmek için, bu yaklaşımlar ile ölçüm sırasında hataları önlemek mümkünse de ölçüm

tamamlandığında, ölçüm hatalarının tespitini sağlayan süreçlere de ihtiyaç bulunmaktadır. Bu amaçla, COSMIC tarafından bir kılavuz "Ölçüm Doğruluğunun Sağlanması için Kılavuz" yayımlanmıştır [9]. Bu kılavuz COSMIC İşlevsel Büyüklük Ölçüm (İBÖ) metodu ile ölçülmüş ölçümlerdeki hataların tespiti için bir takım yöntemler tanımlamakta ve genel hata kategorilerini sınıflandırmaktadır. Ayrıca, yazılım projelerinin işlevsel büyüklük ölçümlerinin ortak kusurları ve sorunlarını [6] [7] [8] ve ayrıntılı hata kategorilerini [10] tespit etmek için bir dizi araştırma da yapılmıştır. Fakat tüm bu çalışmalarda, ölçümlerdeki ortak sorunlar ve kusurlar uzman görüşü ile manüel olarak bulunmuştur. Hata tespitinin uzmanlar tarafından manüel olarak yapılması zaman alıcı bir işittir. Daha önceki ölçümlerde karşılaşılan ve standartlarda yazmayan her türlü beklenmedik durumu ve bu durumdaki varsayımları çok iyi hatırlayan, yorumlayan iyi eğitilmiş ve sertifikalı uzmanların bu çalışmalarda görev alması gerekir.

Ölçümlerdeki hataların tespiti için gerekli süreyi azaltmak, ölçüm doğrulamayı standartize etmek, hataların gözden kaçma olasılığını ortadan kaldırarak, insan kaynaklı ölçüm hatalarını önlemek için bu hataları otomatik olarak yakalayan ve ölçüm yapan uzmanları yönlendiren özel bir araca gereksinim vardır. Literatürde, COSMIC İBÖ [11] yöntemini otomatik olarak gerçekleştiren bazı yaklaşımlar [19] bulunmasına rağmen, ölçümlerdeki hataları otomatik olarak yakalayan bir araç ya da bir yaklaşım önerisi bulunmamaktadır.

Bu çalışmada, bu amaçla geliştirdiğimiz "R-COVER Aracı" geliştirme faaliyetleri ve sonuçları sunulmuştur. Hata tespiti yapılacak yazılım cinsleri için öncelikle hata kategorileri belirlenmiş ve bu Hata Kategorilerine (HK) bağlı olarak [10], hata tespit yaklaşımı ve algoritmaları geliştirilmiştir. Araç geliştirilirken, Yönetim Bilişim Sistemleri (YBS) uygulama tipleri göz önüne alınarak HK oluşturulmasına rağmen ileride gerçek zamanlı sistemlere de uygulanabilmesi amacı ile bu yönde de incelemeler yapılmış ve ilk bulgular sunulmuştur.

Çalışmanın geri kalanı şu şekilde düzenlenmiştir; bölüm 2'de ilgili literatür araştırmaları özetlenmiş, bölüm 3'te ise çalışmanın amaçları ve yapılan vaka çalışması verilmiştir. Vaka çalışmalarının sonuçları bölüm 4'te sunulurken, çalışmanın geçerliliğini tehdit eden faktörler bölüm 5'te anlatılmıştır. Son olarak sonuçlar ve gelecekte yapılabilecek olası çalışmalar ise bölüm 6'da verilmektedir.

2 Literatür

Yazılım büyüklüğünü ölçmek için kullanılan metriklerden birisi olan Fonksiyon Nokta, 1979 yılında Albrecht tarafından öne sürülmüştür [22]. Fonksiyon nokta yazılımın büyüklüğünü yazılımın işlevselliği açısından ölçmektedir. Zaman içerisinde Fonksiyon Nokta Analizi (FNA) olarak bilinen yöntem sıklıkla kullanılan bir işlevsel büyüklük ölçüm yöntemi haline gelmiştir ve yöntemin türevleri ortaya çıkmıştır [18].

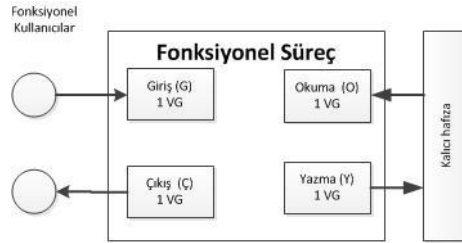
COSMIC İBÖ son yıllarda sıklıkla kullanılan büyüklük ölçüm yöntemlerinden biridir. Şuanda ISO tarafından onaylanmış 5 adet yöntem bulunmaktadır. Bunlar; ISO/IEC 19761 (ISO/IEC, 2003b), ISO/IEC 20926 [1], ISO/IEC 24570 [23], ISO/IEC 29881 [24] ve ISO/IEC 20968 [3] şeklinde sıralanabilir.

Mevcut veri tabanımızda, aracın hata tespit doğruluğunu belirlemek için yapılacak olan durum çalışmasında kullanılacak COSMIC İBÖ metodu ile ölçülmüş çok sayıda projeye ait ölçüm dokümanları bulunmaktadır. Bu nedenle çalışma kapsamında popüler, tüm yazılım tipleri için uygulanabilir ve dünyaca kabul edilen son metotlardan biri olarak COSMIC İBÖ yöntemi [11] seçilmiştir.

COSMIC İBÖ'mü ile yazılım gereksinim dokümanı temel alınarak yazılımın işlevselliği ölçülür. COSMIC İBÖ'nün Fonksiyonel Kullanıcı Gereksinimi (FKG), Fonksiyonel Süreç (FS), Veri Hareketi (VH), İlgili Nesne (İN), Veri Grubu (VG) ve Veri Özniteliği (VÖ) olmak üzere 6 ana bileşeni bulunmaktadır. Yönteme göre bu bileşenlerden VÖ'nin belirlenmesi diğer bileşenler gibi zorunlu tutulmamış, ölçüm uzmanının inisiyatifine bırakılmıştır [11] :

- FKG yazılımın ne yapacağını tanımlayan kullanıcı gereksinimlerinin alt kümesidir.
- FS, bir grup VH'inden meydana gelmiş FKG'lerinin temel parçasıdır.
- İN, fonksiyonel kullanıcı açısından tanımlandığında verini saklandığı varlıktır. İlgili nesnelere veri tabanında tablolara denk gelmektedir.
- VG ise herhangi bir ilgili nesne ile ilişkili olan ve bir grup veri özniteliğinin bir araya gelmesinden oluşmuş veri kümeleridir.
- VÖ, kullanıcı açısından anlam taşıyan, bir veri grubunun en küçük parçasıdır.
- VH, fonksiyonel kullanıcılar ve uygulama arasında veri taşırlar.

Şekil 1'de görüldüğü gibi 4 FS tipi bulunmaktadır. Büyüklük hesaplanırken her FS için "Giriş" "Çıkış" "Okuma" "Yazma" işlemlerinin sayısı göz önünde bulundurulur. Fonksiyonel kullanıcılar "kişi", makine, ya da başka yazılım olabilir.



Şekil 1. VH tipleri ve bunların FS ve VG ile olan ilişkisi [11]

İşlevsel büyüklük ölçümlerinin güvenilirliği yazılım mühendisliği araştırma alanlarının önemli konularından biridir. Yazılım projelerinin efor, maliyet ve bütçelerini doğru kestirmek için, uygulamaların işlevsel büyüklük ölçümleri doğru ölçülmelidir. Ölçüm hataları proje yönetimi ve kontrolünde hatalara sebep olabileceği için, ölçüm hatalarının önlenmesi gereklidir [12]. Bu nedenle işlevsel büyüklük ölçümlerinin güvenilirliğini azaltan hataların kaynaklarını araştırma ve gidermeye yönelik bir dizi araştırma yapılmıştır [1] [2][9].

Kemerer ve Porter, işlevsel büyüklük ölçümlerindeki farklı değerlerin kaynağını bulmak için IFPUG yöntemini kullanarak bir vaka çalışması yapmış ve işlevsel büyüklük ölçümlerinin güvenilirliğini artırmak için tavsiyeler vermiştir. Temel önerile-

rinden birisi, ölçüm süreci ve insan hatası maliyetini azaltmak için işlevsel büyüklük ölçüm sürecinin otomatik hale getirilmesidir [2].

2009 yılında, Ungan ve arkadaşları ölçüm yapan uzmanlar arasındaki farklılıkları saptamak için İBÖ'leri içindeki farklılıkları tespit eden deneysel bir çalışma yapmıştır. Çalışmanın sonuçlarına göre COSMIC ölçümlerindeki farklılıklar bireylerin farklı yorumlar ve varsayımları ile ölçüm yöntemlerindeki temel kurallarının uygulanması sırasında ortaya çıkmaktadır. Bu çalışma, farklı bilgi ve deneyim düzeyine sahip farklı ölçüm uzmanlarının işlevsel büyüklük ölçüm varyasyonlarına neden olabileceğini göstermiştir [6]. 2010 yılında devamında gerçekleştirdikleri çalışmada COSMIC işlevsel büyüklük ölçümlerinin güvenilirliğini arttırmak için ölçücülere durum çalışmaları ve örneklerle zenginleştirilmiş detaylı eğitim verilmesi gerektiğini ve ölçücülerin yazılım gereksinimlerini detaylı olarak anlamaları gerektiğini vurgulamışlardır [8].

Tunalılar ve Demirörs, büyüklük ölçüm sürecini, efor kestirimi metodolojisi (EFES) ile tanımlamışlardır. Bu metodolojinin gereklerini tanımlarken, bazı durumlarda ortak fikir birliği oluşturmak için standart kılavuzların yeterli olmadığını vurgulamışlar, ölçüm sürecini iyileştirmek için, yazılım firmalarına, standart kılavuzlara ek olarak, kurumsal ölçüm prosedürü, şablon ve şirketin proje uygulama tiplerine özel uyarılma notları oluşturmayı tavsiye etmişlerdir. Bu şablonlar gelecekte ölçüm uzmanlarını desteklemek için kullanılmak üzere nadir görülen durumları ya da varsayımları kaydetmektedir [5].

COSMIC güvenilirliğini belirlemek ve sık karşılaşılan COSMIC hatalarını gözlemlemek için, Top [7], 12 sanayi yazılım projesini kullanarak bir çalışma yapmış, ölçümü gerçekleştirmeden önce pilot çalışma yapanların ölçüm raporlarındaki hata miktarlarının daha az olduğunu gözlemiştir. Ayrıca ölçüm uzmanlarının bilgi ve deneyim seviyelerinin ölçüm sonuçlarının doğruluğuna etki eden 2 ana faktör olduğunu belirterek, eğitimlerde sıklıkla yapılan hataların vurgulanması gerektiği sonucuna varmıştır.

Ölçümlerin doğruluğunu arttırmak için 2011 yılında, "Ölçüm Doğruluğunun Sağlanması için Kılavuz" isimli bir rehber yayınlanmıştır [9]. Bu kılavuz Hata Önleme ve Hata Tespiti olmak üzere iki alt başlık içermektedir. Hata tespiti süreci için, sıklıkla karşılaşılan COSMIC hatalarından oluşan bir kontrol listesi verilmiştir. Kılavuzdaki kontrol listesi ile ölçüm uzmanlarının, ölçüm raporlarını kontrol etmeleri ve herhangi bir hata varsa, bu hataları ortadan kaldırmalarına yardımcı olmak amaçlanmıştır. Ancak, kılavuzda sıklıkla yapılan COSMIC hataları detaylı ve açıkça tanımlanmadığı için, önerilen hata tespiti kontrol listesi yeterli değildir. Ayrıca, kontrol listesi çok genel hataları ve onların yüzeysel tanımlarını içermektedir. Ölçüm uzmanlarının ölçüm raporlarını kolaylıkla kontrol etmesi ve gözden geçirmesine izin veren bir yapıya sahip değildir.

2010 yılında, Yılmaz ve arkadaşları, YGD'lerinin kalitesinin İBÖ'lerine etkisi üzerine yapılmış bir çalışmanın ilk sonuçlarını sunmuştur. Bu çalışma, yazılım gereksinim dokümanlarının COSMIC İBÖ'lerinin güvenilirlik ve doğruluğu için ölçüm yöntemleri kadar önemli olduğunu göstermiştir. Ayrıca, çalışmada "Ölçüm Doğruluğunun Sağlanması için Kılavuz" [9] referans alınarak, sıklıkla yapılan COSMIC hataları belirlenip tanımlanmıştır [10].

3 Araştırma Yaklaşımı

Araştırmamız iki adımdan oluşmaktadır:

- COSMIC işlevsel büyüklük ölçümleri için bir hata tespit aracı geliştirmek
- Aracın etkinliğini araştırmak

3.1 Aracın Geliştirilmesi

Şekil 2’de gösterildiği üzere; hata yakalama aracının bir hata tespit yaklaşımı baz alınarak, projelerin İBÖ’lerinin tutulduğu bir ortama entegre olarak geliştirilmesi planlanmıştır. Hata tespit yaklaşımının oluşturulması için, işlevsel büyüklük ölçüm metodunun kendisi [11], onun üzerindeki farklılıklar (varyansları) ve sık karşılaşılan hatalar tanımlanmalıdır. Aracımızın geliştirilmesi için aşağıdaki adımlar gerçekleştirilmiştir:

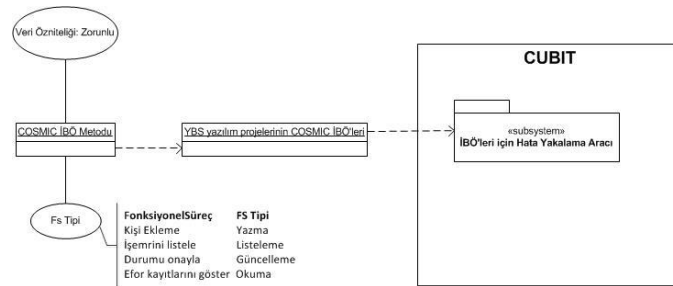
- Hata tespit aracının entegre olarak çalışacağı ortamın kurulması,
- Hata kategorilerinin belirlenmesi
- Hata tespit yaklaşımının geliştirilmesi
- Hata yakalama algoritmalarının ve R-COVER aracının oluşturulması

Hata tespit aracının çalışacağı ortamı kurma. Hata tespit aracının çalışabilmesi için yazılım projelerinin İBÖ’lerinin bulunduğu bir ortam gereklidir. Bunun yanında hata yakalama aracının geliştirilmesi tamamlandıktan sonra, aracın hataları yakalama doğruluğu ve verimliliği de pek çok projenin ölçümlerinin bulunduğu bu ortamda test edilerek araç iyileştirilebilecektir. Bu amaçla, farklı alanlardaki yazılım proje bilgileri ve ölçümlerinin bulunduğu bir veri tabanı gereklidir. CUBIT [20], ODTÜ-Enformatik Enstitüsü’nde geliştirilen, yazılım projelerini ölçmek, ölçüm büyüklüklerini saklamak ve yazılım proje yönetimini kolaylaştırmak için yazılım projelerinin tüm verilerini saklayarak bir veri seti oluşturulmasına yardımcı olan, kolay kullanılabilir, web tabanlı bir araçtır. CUBIT’te halihazırda farklı yazılım projelerinin İBÖ’leri bulunduğu için, sıfırdan bir veri seti oluşturmak yerine hata yakalama aracının CUBIT’e entegre olarak çalışabilen bir modül olarak geliştirilmesine karar verilmiştir.

Hata kategorilerinin belirlenmesi: Ortamın seçiminden sonra, hata yakalama aracının hata tespit işlemi bir "temel"e dayandırılmalıdır. Geliştirilecek olan araç, bu temeli esas alarak İBÖ’lerindeki hataları otomatik olarak bulabilmelidir. Bu amaçla, önceden yapılmış araştırmalarda [7][9] belirtilen ortak ölçüm hatalarını bulmak için literatür gözden geçirilmiş ve ayrıca çeşitli ampirik çalışmalar yapılmıştır [6][8]. Bu çalışmalardan yola çıkarak, otomatik hata tespitine olanak sağlayacak “temel” hata kategorileri tanımlanmıştır.

Hata kategorilerinin belirlenmesi için, YBS uygulama tipi ve ilgili İBÖ’leri seçilmiştir. Bu seçimin sebebi CUBIT ortamının çok sayıda YBS projelerine ait ölçüm bilgilerinin bulunmasıdır. Belirlenmiş bir YBS projesine ait yazılım gereksinim dokümanını 10 farklı ölçüm uzmanı ölçmüştür. Bu ölçümler kullanılarak toplamda 23 hata kategorisi tanımlanmıştır. Hata Kategorilerini (HK) oluşturmak için izlenen yöntem ve süreçlerin detayları bir önceki araştırmamızda belirtilmiştir [10]. 23 hata

kategorisinden 15,'i hata tespit aracının temelini oluşturmuştur. Diğer 8 hata kategorisinin varlığını tespit edebilmek için yazılım gereksinim dokümanlarının kontrol edilmesi gerekmektedir. Araç tarafından otomatik olarak tespit edilebilen 15 HK'si Tablo 1'de verilmiştir. Hata kategorileri, Ölçücü Kaynaklı (Ö-K), Ölçüm Süreci Kaynaklı (ÖS-K) ve Yazılım Gereksinim Dokümanı Kaynaklı (YGD-K) olabilir. Hata kategorilerinin Referans Kaynakları da (RK) Tablo 1'de gösterilmektedir. HK'leri YBS uygulamalarının COSMIC İBÖ dokümanları kullanılarak oluşturulduğu için, çoğu hata kategorisinin kapsamı YBS'ye özeldir. Başka bir deyişle hata kategorilerinin bazıları genel olarak tüm yazılım uygulama alanları için geçerliken bazıları sadece YBS uygulamaları için geçerlidir ve diğer uygulama alanlarında bu tarz hatalara rastlanmayabilir.



Şekil 2. COSMIC İBÖ metodu ile hata yakalama yaklaşımı ve aracı arasındaki ilişki

Hata kategorilerinin belirlenmesi ve hataların ortaya çıkış nedenleri araştırıldıktan sonra, her bir hata kategorisinin yapısı sırası ile tüm ölçüm dokümanlarında analiz edilmiştir. Aynı hata kategorisine ait ölçüm hataları, farklı dokümanlarda da bulunuyorsa hataların otomatik tespiti için bu yapı yeterli ise, yani YGD bilgisine ihtiyaç duyulmadan tespiti sağlanacaksa, ilgili hata kategorisi için hata yapısı olarak tanımlanmıştır. Bu hata yapıları daha sonra aracın geliştirilmesi sırasında algoritmaları oluşturmak için kullanılmıştır. Algoritmaların temelini oluşturan hata yapıları Tablo 1'de belirtildiği gibi İstatistiksel, Anlamsal ve Sözdizimsel olmak üzere 3 sınıf altında toplanabilir.

Hata tespit yaklaşımının geliştirilmesi. Bu çalışmada Tablo 1'deki hata kategorileri tipindeki hataları otomatik olarak tespit etmek için bir yaklaşım geliştirilmiştir ve bu yaklaşım da kısaca "R-COVER yaklaşımı" olarak isimlendirilmiştir. Hata kategorileri YBS uygulamalarının COSMIC İBÖ'leri temel alınarak tespit edildiği için, bu yaklaşım özellikle YBS uygulamalarının İBÖ'leri için geçerlidir. R-COVER yaklaşımı ve hata kategorileri, hata tespit aracının temelini oluşturmaktadır.

R-COVER yaklaşımı, standart COSMIC İBÖ yönteminin [11] bazı bileşenleri üzerinde değişiklikler yapmayı ve zorunluluklar tanımlamayı öngörmektedir. Şöyle ki; COSMIC İBÖ yöntemi ana bileşenlerinden birisi olan veri özniteliği (VÖ)'nin standartta [11] ölçüm sırasında tespiti zorunlu kılınmamış, eklemek ölçücünün kendisine bırakılmıştır. Ancak, R-COVER yaklaşımının kullanılabilmesi için ölçümlerde bu

bilgi mutlaka olmalıdır. Çünkü, ölçümlerde veri özniteliği bilgileri boş bırakılırsa, İN'ne yönelik hatalar tespit edilemez.

Tablo 1. Hata kategorileri

Hata Kaynağı	Hata yapısı	Hata Kategorisi	Referans	Kapsamı	Tanımı	
Ö-K	Anlamsal	1	FS tekrarı	[9]	Genel	İki farklı FS aynı (VG, VH) ikililerine sahipse ve aynı işlevselliği temsil ediyorsa, FS'lerden biri fazla olabilir
ÖS-K	Anlamsal	2	Güncelleme öncesi silme listeleme unutulmuş	[7][9]	YBS özel	Güncelleme tipindeki bir FS için, Listeleme tipindeki FS'in ölçümü unutulmuş olabilir.
ÖS-K	Anlamsal	3	Silme öncesi listeleme unutulmuş	[7][9]	YBS özel	Silme tipindeki bir FS için, Listeleme tipindeki FS'in ölçümü unutulmuş olabilir. Güncelleme tipindeki bir FS için, Okuma tipindeki FS'in ölçümü unutulmuş olabilir.
ÖS-K	Anlamsal	4	Güncelleme öncesi okuma unutulmuş	[7][9]	YBS özel	Güncelleme öncesi okuma unutulmuş olabilir.
Ö-K	Anlamsal	5	Yazma/Silme/Güncelleme tipindeki FS'lerde "W" tipinde VH unutulmuş	-	YBS özel	Yazma/Silme/Güncelleme FS'lerinde Y tipinde VH ölçülmemiş olabilir
Ö-K	Anlamsal	6	Listeleme tipindeki FS "W" tipinde VH içerir	-	Genel	Listeleme tipindeki FS'te W tipindeki VH yanlış ölçülmüş olabilir
Ö-K	Anlamsal	7	Bir FS birden fazla aynı VH içerir	-	Genel	Bir FS'te birden fazla aynı VH bulunuyorsa, bu VH'lerin bazıları fazladan ölçülmüş olabilir
Ö-K	Anlamsal	8	Her FS en az 2 VH'den oluşur	[11]	Genel	Bir FS'te 1 veya daha az VH bulunamaz
Ö-K	Anlamsal	9	Her FS en az 1 "W" veya "X" tipinde VH içerir	[11]	Genel	Bir FS'te Wveya X tipinde VH yoksa
Ö-K	Anlamsal	10	Her FS en az 1 "E" VH içerir	[11]	Genel	Bir FS'te E tipinde en az 1 VH yoksa
ÖS-K	İst.	11	Listeleme tipindeki FS, Güncelleme/Silme tipindeki FS'lerin içinde ölçülmüş	[7]	YBS özel	Listeleme tipindeki FS, güncelleme veya silme tipindeki FS içinde ölçülmüş olabilir
Ö-K	İst.	12	Yazma/Silme/Güncelleme FS'leri tek bir FS olarak ölçülmüş	-	YBS özel	3 farklı FS (Yazma/Silme/Güncelleme) tek bir FS olarak ölçülmüş olabilir
Ö-K	Anlamsal	13	VG tekrarı	-	Genel	Birden fazla aynı VG mevcut olabilir
Ö-K	Sözd.	14	Kullanıcı ara yüz parçaları ve sistem kullanıcıları VG/İN olarak düşünülmüş	[9]	Genel	Ölçüm uzmanı "ekran", "düğme", "menü" gibi kullanıcı arayüzü bölümlerini İN veya VÖ olarak düşünerek FS olarak ölçülmüş olabilir
Ö-K	Sözd.	15	İN yanlış isimlendirilmiş	-	Genel	İN'lerini isimlendirirken "kaydet", "seç", "bul" gibi filleri kullanmış olabilir

Ayrıca, R-COVER yaklaşımımızda, “FS tipi” isimli “yeni bir işlevsel büyüklük bileşeni” tanımlanmıştır. Bu yeni bileşenin örneği Şekil 2’de YBS uygulamaları için gösterilmiştir: “Silme”, “Yazma”, “Okuma”, “Güncelleme”, ve “Listeleme”. Herhangi bir FS tipine uymaması durumu için “Diğer” olmak üzere bir FP tipi daha eklenmiştir. **Hata yakalama algoritmalarının ve R-COVER aracının oluşturulması.** Hata yakalama aracımız CUBIT’in bir modülü olarak geliştirileceği için, CUBIT’in E-R şeması ve mevcut veri tabanı yapısı fonksiyonel süreç tipi ve veri özniteliği büyüklük bileşenleri göz önünde bulundurularak incelenmiştir. “VÖ” büyüklük bileşeni CUBIT’in mevcut veri tabanı tablolarında bulunmaktadır ancak, “FS tipi” yeni tanımlanmış bir büyüklük bileşeni olduğu için veri tabanına eklenmesi gerekmektedir. Bu nedenle CUBIT’in bazı veri tabanı tablolarına FS tipi alanı eklenerek, CUBIT veri tabanı ve E-R şeması güncellenmiştir.

```

Start
  If(outerDM=innerDM & outerDG =innerDG)
    Counter++
    If(Counter=size of outerFP & Counter=size of innerFP & outerFPtype=innerFPtype)
      print warning
    End if
  End if
End

```

Şekil 3. HK1’in sözde kodu

Tablo 1’de gösterilen 3 farklı hata yapısına göre farklı algoritmalar geliştirilmiştir. “İstatistiksel” kontrol yapan algoritmalar, bu hesaplamaları yapabilmek için CUBIT veri tabanındaki yazılım projelerinin büyüklük ölçümlerini kullanmaktadır. “Anlamsal” hata yapıları oluşturan hata kategorilerine ait hataları yakalamak için ölçüm dokümanlarını anlamsal olarak kontrol eden algoritmalar oluşturulmuştur. “Sözdizimsel” hata yapılarına sahip hataları yakalamak için ise dokümanları sözdizimsel olarak kontrol eden algoritmalar oluşturulmuştur.

COSMIC List

1- Existing number of R DM is 50, List FP is 3
 - Expected number of R DM is 44, List FP is 5
 - While Existing R is equal to Expected R, existing List is less than expected List, List FP may be defined as part of FP Update/Delete.
 2- Existing number of W DM is 20, Create FP is 7, Update FP is 5, Delete FP is 0
 - Expected number of W DM is 24, Create FP is 6, Update FP is 4, Delete FP is 4
 - Existing values are less than the expected ones, Create/Update/Delete operations might be combined. They should be measured as separately.

Name	Description	Data Movements	Count	Verification Results
FP1	List Budget Category	BudgetCategoryInfo BudgetCategoryInfo BudgetCategoryInfo	E R X	3
FP3	Retrieve Budget Category	BudgetCategoryInfo UpperBudgetCategoryInfo BudgetCategoryInfo BudgetCategoryInfo BudgetCategoryInfo	E R X R X	5 Warning, there are multiple occurrences of same data movements BudgetCategoryInfo within the same FP.
FP2	Add Budget Category	BudgetCategoryInfo UpperBudgetCategoryInfo UpperBudgetCategoryInfo BudgetCategoryInfo Error/Confirmation	E R X W X	5

Şekil 4. FS ve VH hataları sonuç ekranı

Hata yakalama algoritmalarının geliştirilmesi için her bir hata kategorisi için tespit edilen hata yapıları kullanılarak öncelikle sözde (pseudo) komut, daha sonra ilgili algoritmaları oluşturulmuştur. Bir sözde kod örneği Şekil 3’te verilmiştir. Bu sözde

kod, HK1 için oluşturulmuştur. Bu sözde koddan oluşturulan algoritma, eğer iki farklı fonksiyonel süreç aynı (VG, VH) çiftlerini taşıyorsa ve iki fonksiyonel sürecin büyüklükleri aynı ise ölçümde HK1 tipinde bir hata olduğuna dair uyarı vermektedir. Aslında iki farklı fonksiyonel süreç aynı (VG, VH) ikililerine sahip olabilir, burada belirleyici olan bileşen fonksiyonel süreçlerin tipidir.

Araç kullanıcıya, uyarı ve hata mesajı olmak üzere 2 farklı mesaj vermektedir. Hata mesajı, COSMIC İBÖ yönteminin en temel kurallarının yanlış kullanılması sonucunda verilmektedir. Uyarı mesajlarının hepsinde ölçüm hatası olmayabilir. Ölçüm uzmanlarının uyarı mesajlarındaki hataları düzeltmeden önce YGD'lerini kontrol etmeleri gerekir. R-COVER aracının ürettiği bir sonuç raporu Şekil 4'te sunulmuştur.

3.2 Aracın Hata Yakalama Doğruluğunun ve Verimliliğinin Tespit Edilmesi

Durum Çalışmasının Planı. Aracın hata yakalama doğruluğunu ve verimini tespit etmek için, 7 farklı YBS uygulamasının 26 COSMIC İBÖ'ü ile 17 gerçek zamanlı ve gömülü yazılım projesinin ölçümleri kullanılmıştır. Durum çalışması sırasında farklı verim testleri için YBS uygulamalarının ölçümleri 3 ayrı grupta toplanırken, gerçek zamanlı ve gömülü yazılım projelerinin İBÖ'leri 1 grup altında toplanmıştır. Aracın hata yakalama doğruluk ve performansını farklı durumlarda gözlemek için, deneyimli ve deneyimsiz ölçücüler tarafından ölçülmüş çok çeşitli COSMIC İBÖ'ler durum çalışmasına dahil edilmiştir. Tablo 2'de hangi grupta hangi projenin olduğu ve kaç adet işlevsel büyüklük ölçümü kullanıldığı özetlenmektedir.

Tablo 2. Proje ve ölçüm grupları

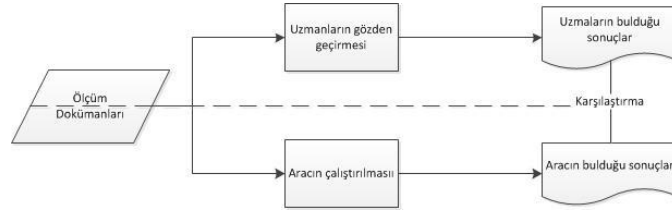
Gru p	Prj No v e Adı	Ref. Anahtar	Firma	Uygulama Tipi	Ölçümlerin Sayısı	Ölçücü Seviyesi	Amaç
G1	1, Proje1	Anahtar1	ODTÜ	YBS	10	Sınırlı deneyim, eğitim almış	Aracın doğruluğunu test etmek
G2	1, Proje2	Anahtar2	ODTÜ	YBS	11	Sınırlı deneyim, eğitim almış	Aracın doğruluğunu test etmek
G3	5, Proje [3-7]	Anahtar3- Anahtar7	Firma A	YBS	5	3 yıl deneyimli	Endüstriden alınmış gerçek proje ölçümlerinde aracı test etmek
G4	17, Proje [8-24]	Anahtar8- Anahtar24	Firma B	Gerçek zamanlı ve Gömülü	17	5 yıl deneyimli	Farklı uygulama alanlarında aracın uygulanabilirliğini test etmek

Şekil 5'te gösterildiği gibi, aracın hata tespit doğruluğunu gözlemek için, uzmanlar tarafından manüel olarak gözden geçirme yöntemi ile bulunmuş hata sonuçları ile aracın bulduğu hataların her ölçüm için tek tek kontrol edilmesi planlanmıştır.

Ölçüm dokümanlarındaki hataların kayıt altında tutulması için, aracı çalıştırırken ve gözden geçirme yöntemi sırasında kullanılmak üzere "Hata İşaretleme Formu" oluşturulmuştur. Her bir ölçüm dokümanındaki hatalar gözden geçirme yöntemi ve

araç ile tespit edilirken, hatalar bu forma girilecektir. Hata İşaretleme Formunun bir kısmı örnek olması için Tablo 3’te verilmiştir.

Uzman gözden geçirme süreci. Gözden geçirme sürecine başlamadan önce, işlevsel büyüklük ölçüm dokümanlarındaki hataları tespit etmek için her farklı yazılım projesi için referans olarak kullanılacak doğruluğu onaylanmış COSMIC işlevsel büyüklük ölçümüne ihtiyacımız vardır. Tablo 2’de her grup için kaç adet referans olarak kullanılacak cevap anahtarları oluşturulduğu bilgisi verilmiştir. Örneğin, G1’de Projeler bulunmaktadır ve bu proje için Ref. Anahtar1 oluşturulmuştur. Bu grupta toplamda 10 işlevsel büyüklük ölçümü bulunmaktadır. Bu ölçümler Proje1’in yazılım gereksinim dokümanı kullanılarak 10 farklı ölçücü tarafından gerçekleştirilmiştir. Ölçüm dokümanları sistemde Excel formatında bulunduğu için referans cevap anahtarları da excel formatında hazırlanmıştır.



Şekil 5. Durum çalışmasının akışı

Gözden geçirme sürecinin hata kategorileri bazında sürdürülmesi öngörülmüştür. Örneğin, Hata Kategorisi 1 tipindeki hataların varlığının kontrolüne ilk ölçüm dokümanından başlayarak, son ölçüm dokümanına gelinceye kadar devam edilecek, her hata bulunduğu ilgili ölçüm dokümanı için “Hata İşaretleme Formuna” kayıt edilecektir. Hata kategorisi-1 için bu süreç tamamlandığında, aynı süreci uygulamak üzere hata kategorisi 2’ye geçilecektir.

Tablo 3. G1 ölçümlerinin hata işaretleme formundan bir kesit

	Hata Kategorileri	Kontrol Tipi	Toplam	ÖD1	ÖD2	...	ÖD10
1	FS tekrarı	Manüel	60	4	16	...	11
		Araç	62	0	16	...	11
2	Güncelleme öncesi listeleme unutulmuş	Manüel	17	1	1	...	1
		Araç	27	5	1	...	0
15	İlgi nesnelere isimleri fiil içeriyor	Manüel	66	0	1	...	2
		Araç	64	0	1	...	0

Aracın Çalıştırılması. Aracı kullanarak hata tespiti yapmadan önce, CUBIT veri tabanında bulunan Tablo 2’de gösterilen şekilde gruplandırılmış bütün işlevsel büyüklük ölçümlerinin R-COVER yaklaşımında belirtilen kurallara uygunluğunun kontrol edilmesi gerekmektedir. Bu kuralları ihlal eden eksikler varsa, ölçümler veri tabanında güncellenir.

Her bir ölçüm dokümanı için, her bir fonksiyonel sürecin FS tipi bilgisinin tespit edilip veri tabanında güncellenmesi gerekmektedir. Ayrıca, tanımlanmamış VÖ'ler tespit edilecek ve CUBIT veri tabanına girilecektir. FS tiplerinin ve VÖ'lerinin tespiti için her bir projenin yazılım gereksinim dokümanı kullanılmalıdır.

Bu çalışmamızda aracın etkisini görmek amacıyla, G1 ölçümleri için manüel ve otomatik R-COVER araç kullanımı eforunun karşılaştırılması da planlanmıştır. Bileşenleri tespit etmek için harcanacak efor, bu yöntemlerin performans kıyaslaması için kayıt altında tutulmuştur.

Aracın çalışması için gerekli her bir ölçümün eksik bilgileri CUBIT veri tabanında güncellendikten sonra, araç sırası ile her bir ölçüm dokümanı için çalıştırılmalıdır.

Durum Çalışmasının Uygulanması. Planlamaya uygun olarak 2 aktivite şeklinde uygulanmıştır.

Uzman gözden geçirme süreci. Gözden geçirme sürecinin ilk adımı olarak, her projenin güvenilir ve doğru bir işlevsel büyüklük ölçümü gerçekleştirilmiş, daha sonra bu ölçümler ölçüm deneyimine ve COSMIC sertifikasına sahip uzman bir ölçücü tarafından kontrol edilmiştir. Bulunan hatalar düzeltilmiş ve her farklı yazılım projesi için güvenilir bir referans işlevsel büyüklük ölçümü hazırlanmıştır.

Durum çalışmasının planı kısmında anlatıldığı gibi gözden geçirme yöntemi ile ölçümlerdeki hata tespitlerine G1 ölçümlerinden başlanmıştır. Gözden geçirme yöntemi kullanarak hata tespiti için harcanan süreler sadece G1 grubu ölçümleri için kayıt altında tutulmuştur ve bu süreler Tablo 5'te verilmektedir. G1 ölçüm dokümanlarında 15 hata kategorilerinin kontrolü tamamlandıktan sonra, aynı süreç diğer gruplar için gerçekleştirilmiştir.

Aracın Çalıştırılması. Araç çalıştırılmadan önce yazılım projeleri ölçümleri CUBIT veri tabanında R-COVER yaklaşımına göre güncellenmiştir. Bu nedenle YBS uygulamalarının olduğu G1, G2 ve G3 gruplarında bulunan yazılım proje ölçümlerinin fonksiyonel süreçlerinin tipleri ve ilgi nesnelerinin veri öznitelikleri belirlenmiş, bu bileşenler CUBIT veri tabanında güncellenmiştir. Ancak G4'teki gerçek zamanlı ve gömülü yazılım projelerinin FS tiplerini belirlemeye çalışırken bazı zorluklarla karşılaşmıştır. R-COVER yaklaşımın tanımladığı fonksiyonel süreçlerin tiplerinin (Yazma, Silme, Güncelleme, Listeleme, Okuma ve Diğer) gerçek zamanlı ve gömülü sistemlerin ölçümlerinde geçerli olmadığı bu uygulamalarda fonksiyonel süreç tiplerinin bu tanımlı tiplerden tamamen farklı olduğu fark edilmiştir. Bu nedenle, G4'teki ölçümlerin sadece veri öznitelikleri CUBIT veri tabanında güncellenmiştir. Aracın hata tespit doğruluğunu ve verimli çalışmasını analiz etmek için, uzman gözden geçirme süreci sonunda bulunan hatalar ile aracın bulduğu hatalar karşılaştırılmıştır. Analiz sonuçları bölüm 4'te özetlenmiştir.

4 R-COVER Uygulama Sonuçları

Tablo 4, G1, G2 ve G3 ölçümleri için gözden geçirme yöntemi ile bulunmuş hata sonuçları ile aracın bulduğu hata sonuçlarının karşılaştırılmasını göstermektedir. G4 gerçek zamanlı ve gömülü yazılım projelerinin ölçümlerini içerdiği için bu grubun

ölçümlerindeki hata sonuçlarını YBS uygulamaları ile birlikte değil kendi içinde ayrıca değerlendirmek daha uygundur.

Tablo 4'teki "aynı hataların tespiti" kolonu gözden geçirme yöntemi sonucunda bulunan hatalar ile aracın bulunduğu hataların aynı olma oranını, "Tespit edilemeyen hatalar" kolonu gözden geçirme yöntemi ile bulunmuş ancak aracın bulamadığı hataların oranını ve son olarak "Yanlış bulunan hatalar" kolonu ise gözden geçirme yöntemi sonucunda bulunamamış ancak aracın fazladan bulunduğu hataların oranını göstermektedir.

Proje1 ve Proje2 hem deneyimsiz ölçücüler (öğrenciler) tarafından ölçüldüğü hem de benzer süreçler sonucunda ortaya çıktığı için, G1 ve G2 ölçüm dokümanlarında bulunan hata sonuçlarının birbiri ile tutarlı olması beklenmekteydi. Ancak Tablo 4'e göre aracın gözden geçirme yöntemi ile bulunan hatalar ile aynı hataları bulma oranı G2'de daha yüksektir. Bunun nedeni araştırıldığında G1 ölçüm dokümanlarında hemen hemen her hata kategorisinden hatanın bulunduğu ve hata sayısının çok olduğu ortaya çıkmıştır. G2 ölçümlerinde ise daha az hata olduğu görülmüştür. Bu nedenle, aracın doğru hata tespit etme oranı G2 ölçümlerinde daha yüksektir.

Gözden geçirme yöntemi ve araç tarafından herhangi bir hata tespiti yapılamamış ise, bu hata kategorisi aracın doğru hata tespiti hakkında fikir vermemiştir. Bu kategoriler ise Tablo 4'te "N/A" olarak belirtilmektedir.

Tablo 4. G1, G2 ve G3 için aracın doğru hata tespit etme verimliliği [4]

HK	Aynı hataların tespiti			Tespit edilemeyen hatalar			Yanlış bulunan hatalar		
	G1	G2	G3	G1	G2	G3	G1	G2	G3
1	88%	100%	100%	12%	0%	0%	15%	36%	90%
2	76%	100%	100%	24%	0%	0%	52%	59%	0%
3	96%	100%	N/A	17%	0%	N/A	8%	44%	N/A
4	70%	100%	N/A	35%	0%	N/A	18%	46%	N/A
5	73%	100%	N/A	27%	0%	N/A	5%	0%	N/A
6	100%	N/A	N/A	0%	N/A	N/A	40%	N/A	N/A
7	84%	100%	100%	16%	0%	0%	16%	0%	0%
8	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
9	100%	100%	N/A	0%	0%	N/A	0%	0%	N/A
10	100%	100%	100%	0%	0%	0%	0%	0%	0%
11	14%	40%	NA	86%	0%	NA	50%	60%	100%
12	67%	0%	NA	33%	100%	NA	43%	100%	100%
13	100%	100%	N/A	0%	0%	N/A	3%	0%	N/A
14	98%	100%	0%	2%	0%	0%	3%	26%	100%
15	97%	100%	100%	3%	0%	0%	0%	45%	0%

YBS uygulamaları işlevsel büyüklük ölçümleri için tanımlanan FS tipleri, gerçek zamanlı ve gömülü yazılım büyüklükleri için kullanılamamıştır. Bu nedenle araç G4 ölçümleri için anlamlı hata tespitleri yapamamıştır. Örneğin, G4 ölçümleri araç tarafından doğrulanırken, HK1 ve HK7 için bulunan hata sayıları ile diğer hata kategorilerinin hata sayıları arasında büyük bir fark olduğu ortaya çıkmıştır. Bunun sebebi

incelendiğinde, HK1 Ve HK7'ye ait hata uyarılarının hepsinin yanlış olduğu gözlemlenmiştir. FS tipleri G4 ölçümleri için CUBIT veri tabanında boş bırakıldığı için aracın yanlış uyarı verdiği ortaya çıkmıştır. FS tiplerinin gerçek zamanlı ve gömülü sistemlerin ölçümleri için belirlenmiş ve veri tabanında güncellenmiş olmasının bağlı olarak; gerçek zamanlı ve gömülü sistemlerin İBÖ için aracımızın hata tespit sonuçlarının doğru olması öngörülebilir.

Her hata kategorisi için aracın davranışı incelendiğinde, hata kategorilerinin Genel ve YBS uygulamaları için özel olmak üzere 2'ye ayrılması gerektiği ortaya çıkmıştır. Hangi hata kategorisinin genel, hangilerinin YBS uygulamaları için özel olduğu bilgisi Tablo 1'de belirtilmiştir.

Bulgular ve Tablo 4, 23 hata kategorisinden 15'ine ait hataların ve YGD bilgisi olmadan otomatik olarak ve kolaylıkla tespit edilebileceğini bize göstermiştir.

Uzman gözden geçirme ve aracın çalıştırılma süreci birkaç işlemde oluşmaktadır. Her bir işlem için harcanan efor Tablo 5'te verilmiştir. Efor adam*saat cinsinden kaydedilmiştir. G1 ölçümleri için uzman gözden geçirme süreci için harcanan efor 116 adam * saattir. Aracı kullanmak için gerekli ön hazırlık ve aracın hata tespiti için harcanan efor 9 adam * saattir. R-COVER aracı COSMIC İBÖ'lerindeki hataların tespiti için harcanan efordan büyük bir tasarruf sağlar.

Tablo 5. G1 için efor kayıtları [4]

İşler	Efor (dk)	Efor (saat)
Cevap anahtarı hazırlama	860	14
Uzman gözden geçirme için Hata İşaretleme Formunun hazırlanması	510	8,5
Hata kategorilerinin belirlenmesi ve seçilmesi	2440	41
Uzman gözden geçirme süreci	2910	48,5
İstatistiksel hesaplamalar	240	4
Uzman gözden geçirme (Toplam)	6960	116
FS tiplerinin belirlenmesi ve CUBIT veri tabanında güncellenmesi	180	3
Veri özniteliklerinin belirlenmesi ve CUBIT veri tabanında güncellenmesi	315	5,25
Aracın çalıştırılması (10 ölçüm dokümanı için)	30	0,5
Aracın çalıştırılması (Toplam)	525	9

5 Durum Çalışmasının Kısıtları

Durum çalışmasının uzman gözden geçirme süreçleri, bu makalenin bir yazarı tarafından yapılmıştır. 3 yıllık deneyime ve COSMIC ölçüm sertifikasyonuna sahiptir. Çalışmadaki hataları en aza indirmek için 5 yıl ölçüm deneyimi olan ve COSMIC İBÖ sertifikasına sahip bir uzman yazara kritik noktalarda yardım etmiş, 9 yıllık bir başka COSMIC ölçüm uzmanı tarafından sonuçlar gözden geçirilmiştir. Belirlenen hata kategorileri ve R-COVER aracı, kısıtlı sayıda YBS uygulamalarının COSMIC İBÖ'lerinden yola çıkılarak geliştirilmiştir. Aracın daha farklı uygulama alanları veya daha çok sayıda YBS uygulamasının ölçümünde çalıştırılması hata tespitlerinde benzer sonuçlar yaratmayabilir.

6 Sonular ve İleriye ynelik alıřmalar

Bu alıřmada, COSMIC İBÖ'lerin gvenilirlik ve doęruluęunu arttırmak iin, COSMIC İBÖ'leirindeki lm hataların otomatik olarak yakalanmasına olanak saęlayan, bir yaklařım ne srlmř, bu yaklařım temel alınarak R-COVER aracı geliřtirilmiřtir. Ara YBS uygulamalarının COSMIC İBÖ'lerindeki hataların, yazılım gereksinim dokmanındaki herhangi bir bilgiye ihtiya duyulmadan otomatik olarak yakalanmasını saęlar.

Sonraki alıřmamızda, R-COVER aracının uygulandıęı yazılım uygulama alanlarının geniřletilmesi planlanmıřtır. Yaklařım ve ara COSMIC İBÖ'lerindeki hataları tespit etmeye ynelik olarak geliřtirilmiřtir. Ancak, IFPUG, UniFSM gibi farklı iřlevsel byklk lm metotları ile lmlř İBÖ'lerdeki hataları bulmaya ynelik olarak da ileride eklemeler yapılacaktır.

Kaynaklar

1. ISO/IEC (2003c). IS 20926: Software Engineering - IFPUG 4.1 Unadjusted FSM Method - Counting Practices Manual.
2. Kemerer, C. F., & Porter, B. S. Improving the reliability of function point measurement: an empirical study. *IEEE Transactions on Software Engineering*, 18(11), 1011 –1024. doi:10.1109/32.177370 (1992).
3. ISO/IEC (2002c). 20968:2002: Software Engineering - MkII Function Point Analysis - Counting Practices Manual, (2002).
4. Yılmaz G., An Automated Defect Detection Approach For COSMIC Functional Size Measurement Method, Middle East Technical University, Ankara (2012).
5. Tunalılar S., Demirors O., “EFES, An Effort Estimation Methodology”, Joint Conference of the 22nd International Workshop on Software Measurement and the Seventh International Conference on Software Process and Product Measurement (2012).
6. Ugan, E., Demirors, O., Top, ., & zkan, B. An Experimental Study on the Reliability of COSMIC Measurement Results. *Software Process and Product Measurement, Lecture Notes in Computer Science* (Vol. 5891, pp. 321–336) (2009).
7. Top, O. O., Demirors, O., & Ozkan, B Reliability of COSMIC Functional Size Measurement Results: A Multiple Case Study on Industry Cases. *Software Engineering and Advanced Applications, Euromicro Conference* (Vol. 0, pp. 327–334). Los Alamitos, CA, USA: IEEE Computer Society . (2009).
8. Ugan, E. Evaluation of Reliability Improvements for COSMIC Size Measurement Results., *IWSM* (2010).
9. The Common Software Measurement International Consortium (COSMIC): Guideline for Assuring the Accuracy of Measurements, Version 1.0 (2011).
10. Yılmaz, G., Ugan, E., Demirors, O. The effect of the quality of software requirements document on the functional size measurement. [Online].Available: <http://www.uk sma.co.uk/conference2011/presentations/05GokcenYilmazTheEffectoftheQualityofSoftwareonFSM.pdf> (2010).
11. ISO/IEC (2003b). 19761: COSMIC Full Function Points Measurement Manual, v. 2.2.
12. Glass, R. L. *Facts and Fallacies of Software Engineering* (1st ed.). Addison-Wesley Professional (2002).

13. Stern, S., Guetta, O. "Manage the automotive embedded software development cost by using a Functional Size Measurement Method (COSMIC)", European Congress of ERTS (2010).
14. Khelifi, A., Abran, A., "Design Steps for developing Software Measurement Standard Etalons for ISO 19761 (COSMIC-FFP)", WSEAS International Conference on COMPUTERS (2007).
15. Tunalilar, S. "EFES: An Effort estimation Methodology", PHD Thesis, Middle East Technical University, (2011).
16. Zubrow, D., Can you trust your data? Measurement and Analysis Infrastructure Diagnosis, Presentation, <http://www.sei.cmu.edu/library/assets/meas-infrastructure.pdf> (2011).
17. Ebert, C., Dumke, R., Bundschuh, M., Schmietendorf, A. "Best Practices in Software Measurement – How to use metrics to improve project and process performance". Springer-Verlag (2004).
18. Symons, C. Come Back Function Point Analysis (Modernized) – All is Forgiven!, 4th European Conf. on Software Measurement and ICT Control, FESMADASMA, 2001.
19. Diab H., Koukane F., Frappier M., St-Denis R., "µcROSE: Automated Measurement of COSMIC-FFP for Rational Rose Real Time", Information and Software Technology, Volume 47, Issue 3, 1 pp 151-166 (2005).
20. CUBIT , Benchmarking Database and Tools <http://smrg.ii.metu.edu.tr/cubit/auth/login?targetUri=%2F>
21. Trudel, S., Abran, A., Improving Quality of Functional Requirements by Measuring Their Functional Size, Software Process and Product Measurement, pp 287-301 (2008).
22. Albrecht, A. (1979). Measuring Application Development Productivity. *Proc. of IBM Application Development Symp.* (pp. 83–92).
23. ISO/IEC (2005b). IS 24570: Software Engineering – NESMA Functional Size Measurement
24. ISO/IEC (2008). 29881:2008 Information Technology-- Software and systems engineering—FISMA 1.1 functional size measurement method.