

COSMIC Fonksiyonel Büyüklük Ölçüm Yöntemi için Bir Hata Önleme Modeli

Murat Salmanoğlu, Onur Demirörs

Enformatik Enstitüsü, Orta Doğu Teknik Üniversitesi, Ankara, Türkiye

{musalman,demirors}@metu.edu.tr

Özet. Yazılım geliştirme projelerinin yönetiminde en önemli girdilerden bir tanesi iş gücü kestirimidir ve bu kestiriminin gerçeğe yakın yapılabilmesi için geliştirilecek yazılımın büyüklüğünün doğru ölçülmesi gerekmektedir. Yazılım büyüklüğü için en doğru ve tutarlı sonucu veren yöntemlerden bir tanesi yazılımın fonksiyonel büyüklüğünün ölçülmesidir. Büyüklük ölçülürken oluşabilecek hatalar iş gücü kestirimlerini etkileyeceği için yazılım projelerinin başarısını olumsuz etkileyecektir. Bu çalışmada önerilen model COSMIC fonksiyonel büyüklük ölçümü yöntemi kullanılarak yapılan ölçümler sırasında oluşabilecek hataları en aza indirmeyi hedeflemektedir. Hataları engellemek için; farklı fonksiyonel süreç türlerine yönelik tanımlanan veri hareket şablonları ve ilgi objeleri arasındaki ilişkileri gösteren varlık-ilişki tablosu kullanarak ölçüm yapan kişiyi yönlendirir. Model iki farklı vaka çalışması ile test edilmiş ve doğrulama amaçlı çalışmadaki toplam hata sayısını %90, ölçülen büyüklük ile gerçek büyüklük arasındaki farkı %93 oranında azaltabilmiştir.

1 Giriş

Yazılım geliştirme projelerinin en büyük maliyet kalemini iş gücü harcamaları oluşturmaktadır. Bu nedenle proje başlangıcında gerçekleştirilen iş gücü kestirimi projenin bütçe ve takvim hedeflerini tutturabilmesi için oldukça önemlidir. İş gücü kestirimi için kullanılacak olan yazılım büyüklük ölçümleri de bu nedenle oldukça önemlidir. İş gücü kestirimlerinin gerçeğe olabildiğince yakın yapılabilmesi için yazılım büyüklük ölçümü yöntemlerinin de organizasyonlar içerisinde hatasız ve tutarlı bir şekilde uygulanması gerekmektedir.

Yazılım büyüklüğü için kullanılan farklı yöntemler vardır. Geleneksel olarak yazılımlar sıklıkla kod satır sayısı ile ölçülse de, projenin ilk safhalarında kod satır sayısı ile ilgili ölçümler yapmak oldukça zordur. İlk defa Albrecht [1] tarafından ortaya atılan fonksiyonel büyüklük ölçümü, projenin neredeyse her aşamasında uygulanabildiği için projelerde sıklıkla kullanılır. Bu çalışmada ISO tarafından kabul edilmiş 5 farklı fonksiyonel büyüklük ölçüm yöntemi [2][3][4][5][6] arasından otomatik ölçüm çalışmalarına da oldukça uygun bir yöntem [7] olan COSMIC [8] fonksiyonel büyüklük yöntemi kullanılmıştır.

Proje başlangıcında yapılacak büyüklük ölçümleri projenin yaşam döngüsü ile ilgili planlarda belirleyicidir ve bu ölçümlerin mümkün olduğunca hatasız yapılması gerekir. Fonksiyonel büyüklük ölçüm yöntemleri ile ilgili en sık yapılan eleştiri sonuçların ölçümü yapan uzmana göre değişebileceğidir. Fonksiyonel büyüklük ölçümü, satır sayısı kestirimlerine oranla daha tutarlı sonuçlar veriyor olsa da kendi içlerinde bu tür eleştirilere yönelik iyileştirilmesi gereken yönleri vardır [9].

Ölçüm yapan kişiye ve ölçümün yapıldığı organizasyona göre değişebilen sonuçların önüne geçmek için oluşturulmuş bazı yöntemler [10] [11] olsa da, bu yöntemler genel olarak ölçüm yapan kişinin deneyimini ve bilgisini arttırmayı amaçlamakta, yapısal ve yönlendirici bir çözüm sunmamaktadır.

Bu araştırmada kapsamında hedef, ölçüm yapan kişiden bağımsız, ölçümü yönlendirecek ve bu sırada oluşabilecek hataların önüne geçebilecek bir yöntem oluşturmaktır. Bu amaçla COSMIC ölçümleri sırasında karşımıza çıkan temel fonksiyonel süreçlere yönelik oluşturulmuş veri hareketi şablonlarını temel alan bir model ve bu modeli kullanan bir araç geliştirilmiş ve oluşturulan model ve araç vaka çalışmaları ile test edilmiştir.

Bu çalışma yazarların daha önceki araştırmalarından aldıkları sonuçlarla birlikte yapılan geliştirmeleri içeren bir devam çalışmasıdır. Daha önce yayınlanmış olan çalışmaya [12] ek olarak; sunulan yöntem etkinliğinin artması için geliştirilmiş, ikinci bir vaka çalışması ile etkisi test edilmiş ve yöntemin kullanımını kolaylaştıracak bir araç geliştirilmiştir. Gelecek bölümlerde açıklanan ilk vaka çalışması daha önceki çalışma kapsamında yapılmış olup, bu çalışmada konu bütünlüğünün sağlanması için tekrar açıklanmıştır.

İlk vaka çalışmasında modelin önleyebileceği hataları bulmak ve bunlar için veri hareketi şablonlarını oluşturmak, ikinci vaka çalışmasında ise, birinci analiz sonucunda ortaya çıkan yöntem ve aracın ne kadar etkili olabileceğini öğrenmek amaçlanmıştır.

Gelecek maddede araştırma ile ilgili daha önce yapılmış çalışmalardan bahsedilmiştir. 3. bölümde model detaylı olarak açıklanmış, 4. bölümde de yürütülen vaka çalışmaları ile ilgili detaylar verilerek sonuçlarından bahsedilmiştir. Çalışmanın son bölümünde ise çıkarılan sonuçlar ve bu araştırmayı geliştirmek için gelecekte yürütülebilecek çalışmalar özetlenmiştir.

2 İlgili çalışmalar

Yazılım projeleri yönetiminde kullanılan en önemli girdilerden bir tanesi yazılımın büyüklüğüdür. Harcanacak işgücünün ve planlanacak bütçenin kestirilmesi için mümkün olduğunca hatasız büyüklük ölçümleri proje yöneticileri için olmazsa olmazlardandır [13]. Yazılım projeleri ölçümünde tarihsel olarak en çok kod satır sayısı kullanılmış olmakla birlikte proje başlangıcında kod satır sayısı cinsinden ölçüm yapmanın zorlukları nedeniyle iş gücü kestirimi çalışmalarında girdi olarak kullanımı kabul görmemektedir [14].

Kod satır sayısından sonra en çok kullanılan büyüklük ölçüm birimlerinden bir tanesi de fonksiyonel büyüklüktür. Yazılım fonksiyonel büyüklüğü gereksinim dokümanı, tasarım dokümanı, süreç modeli gibi çok farklı ürünler kullanılarak ölçülebildiği

için proje başlangıcında da rahatlıkla ölçülebilmektedir. Bu nedenle kestirim çalışmalarında kullanılabilirliği de oldukça yüksektir.

1979 yılında fikir olarak ilk defa ortaya atılan fonksiyonel büyüklük ölçümü için bugün hali hazırda birçok yöntem bulunmaktadır. Bunlardan 5 tanesi ISO tarafından yayınlanmıştır [2] [3] [4] [5] [6] ve bu 5 yöntem en sık kullanılan yöntemlerdir. Bu çalışmada temel alınan COSMIC fonksiyonel büyüklük ölçüm yöntemi bunlar arasından IFPUG'dan sonra en çok kullanılan [15] yöntemdir.

COSMIC, iş uygulamaları, gerçek zamanlı uygulamalar ve bunların karışımı olan uygulamalarla ilgili fonksiyonel büyüklük ölçümü yapılmasına olanak sağlar. COSMIC ölçümü yapılmadan önce ölçüm amacı belirlenip buna göre ölçüm kapsamı çizilir ve daha sonra bu kapsam içerisinde kalan Fonksiyonel Kullanıcı Gereksinimleri (FKG) kullanılarak ölçüm yapılabilir. FKG'nin içerdikleri Fonksiyonel Süreçlerin (FS) belirlenmesi ölçümün ilk adımıdır. Fonksiyonel süreçler tetikleyici bir olayla başlayan ve sistemi kararlı şekilde bırakan, tek bir amacı olan bağımsız veri hareketleri grubu olarak tanımlanabilir.

FS'ler belirlendikten sonra İlgili Objeleri (İO) ve Veri Grupları (VG) tanımlanmalıdır. İlgili objeleri fonksiyonel kullanıcı gereksinimlerinin bakış açısıyla tanımlanan, yazılımın hakkında bir veri işlediği veya sakladığı herhangi bir şey olabilir. Veri grubu da boş, sıralı ya da gereksiz olmayan, bir ilgili objesi ile ilgili olan bağımsız özellikler grubudur.

Ölçüm aşamasında ölçüm yapacak olan kişi fonksiyonel kullanıcı gereksinimleri içerisindeki veri hareketlerini (VH) tespit edip bunları toplayarak yazılımın COSMIC fonksiyonel büyüklüğüne ulaşabilir. Veri hareketleri fonksiyonel kullanıcı ile yazılım arasındaki veya yazılım ile kalıcı bellek arasındaki sınırı geçen veriyi temsil eder. Dört çeşit veri hareketi vardır, bu hareketler şunlardır (Şekil 1):

- Giriş (E-Entry): Veri fonksiyonel kullanıcıdan fonksiyonel sürece hareket eder.
- Çıkış (X-eXit): Veri fonksiyonel süreçten fonksiyonel kullanıcıya hareket eder.
- Okuma (R-Read): Veri kalıcı bellekten fonksiyonel kullanıcıya hareket eder.
- Yazma (W-Write): Veri fonksiyonel kullanıcıdan kalıcı belleğe hareket eder.



Şekil 1 - COSMIC Veri Hareketleri

Veri hareketleri belirlendikten sonra tüm fonksiyonel süreçlere ait veri hareketleri toplanarak yazılımın COSMIC Fonksiyon Nokta sayısı bulunmuş olur (Formül (1)).

$$\text{Büyükük}(\text{fonksiyonel süreç}_i) = \sum \text{Büyükük}(E_i) + \sum \text{Büyükük}(X_i) + \sum \text{Büyükük}(R_i) + \sum \text{Büyükük}(W_i) \quad [8] \quad (1)$$

Fonksiyonel büyüklük ölçüm yöntemleriyle ilgili olarak yapılan en yaygın eleştiri farklı uzmanların aynı girdileri kullanarak yapacakları ölçümlerde farklı sonuçlara ulaşabilecekleri, bu nedenle ölçüm sonuçlarında farklılıklar görülebileceği yönündedir. Ancak bu yöntemlerin güvenilirlikleriyle ilgili olarak yapılan çalışmalarda ölçüm sonuçlarının ölçümü yapan uzmana bağlı olarak değişebileceğiyle ilgili bazı sonuçlar olsa da [16] [17], fonksiyonel büyüklüğün satır sayısından daha tutarlı bir ölçüm olduğu vurgulanmıştır. Fonksiyonel büyüklük ölçümlerindeki farklılıklar genelde ölçüm yapan kişinin yorum ve çıkarımlarından kaynaklanmakta, farklı organizasyonlarda yapılan ölçümler farklı sonuçlar verse de, aynı organizasyon içerisinde yapılan ölçümlerde benzer sonuçlar alınmaktadır.

Fonksiyonel büyüklük ölçümünün güvenilirliği ile ilgili yapılan çalışmalarda ölçüm yapan kişilerin kurallara ilişkin kişisel yorumlarının ölçüm sonuçlarını etkilediği [9] ayrıca kişilerin tecrübeleri arttıkça daha tutarlı sonuçlar elde edildiği [16] belirtilmektedir. Ungan, Demirörs, Özcan Top, ve Özkan [10] ölçüm güvenilirliği ile ilgili olarak yaptıkları çalışmada ölçücülerin sıklıkla yaptıkları hataları listelemiş ve bu hatalara karşı önlem alınması durumunda güvenilirliğin de artabileceğini belirtmişlerdir. Ungan ve diğerleri [10] bu çalışmada COSMIC yöntemine yönelik eğitimler sırasında, karmaşık kuralların özellikle vurgulanmasının ölçüm hatalarını azaltabileceğini göstermişlerdir.

Ölçüm güvenilirliğini artırma amacıyla kullanıcının yaptığı hataları önlemeye yönelik çalışmalar dışında yapılan hataların ölçüm sonrasında tespitine ve düzeltilmesine yönelik yapılmış çalışmalar da mevcuttur. Yılmaz, Tunalılar ve Demirörs [18] bu amaçla ölçüm sırasında kullanılan kaynaklardan bağımsız olarak ölçüm sonrası hataların tespitini amaçlayan R-COVER aracını geliştirmişlerdir.

Araştırmalardan da görüldüğü gibi, fonksiyonel büyüklük ölçümü genel olarak güvenilir bir yöntem olmakla beraber, ölçüm yapan uzmanların kişisel yorumlarından kaynaklanan bazı farklılıkların ve ölçüm yapanların tecrübe seviyelerine göre yapabildikleri hataların azaltılması ile yöntemin güvenilirliğini arttırmaya yönelik bir katkı oluşturulabilir. Hata sayısını azaltmaya yönelik halı hazırda var olan yöntemler, genel olarak ölçüm yapan kişinin tecrübe seviyesini arttırmaya yöneliktir. Yazarlar geçmiş çalışmalarında temelinin attıkları [12] yöntem ile ölçüm sırasında kullanıcı yorumunu en alt seviyede tutarak kullanıcının hata yapmasını engellemeyi amaçlamışlardır. Bu araştırmanın amacı daha önce bir versiyonu sunulmuş olan [12] COSMIC fonksiyonel büyüklük ölçüm yöntemi ile yapılan ölçümler sırasında yapılabilecek hataları tanımlı kurallar yardımıyla azaltacak yöntemin güncellenmiş sürümünü ve yöntem kullanılarak yapılan analizlerin sonucunu sunmaktır.

3 Önerilen hata önleme yöntemi

COSMIC ile fonksiyonel büyüklük ölçümü yapılması sırasında kullanıcıları yönlendirerek hata yapma ihtimalini azaltmayı amaçlayan yöntem için Özcan Top ve diğerleri [19] ile Ungan ve diğerlerinin [20] çalışmalarında listelenen Tablo 1'deki temel hata çeşitlerini kullanılmıştır. Görüldüğü gibi bu hatalardan bazıları FS türleriyle, bazılarıysa birbirini tetikleyen, art arda gelen işlemlerle ilgili. Önerilen yöntemle öncelikle

bu iki farklı hata grubuna dâhil olan hataların engellenmesi hedeflenmiştir. Hata listesinde çoğunluğu oluşturan bu tür hatalar başarıyla engellendiği takdirde, gözlenen hatalarda önemli bir düşüş olacaktır. Yöntem oluşturulurken bu iki farklı tür hata için iki aşamalı bir model tasarlandı. Birinci aşamada ölçümü yapacak kullanıcı art arda gelen hareketlerin belirlenmesi için ölçümü yapılacak yazılımın varlık-ilişki diyagramını sağlayacak, ikinci aşamada ise ölçümü yapılan her FS için modelde tanımlanmış bir FS türü seçerek, oluşturduğu veri hareketlerinin model içerisinde tanımlanmış olan veri hareketi şablonuna uygunluğunu kontrol edecektir [12].

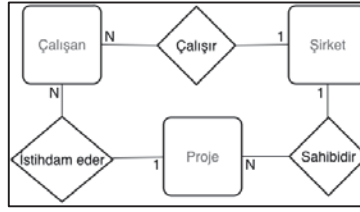
Tablo 1. - COSMIC ölçümü sırasında sık rastlanan hataların sınıflandırması

Farklı hata mesajlarının farklı çıkış hareketleri olarak sayılması	Hata/doğrulama mesajları için çıkış hareketlerinin olmaması
Listele ve getir süreçlerinin birleştirilmesi	Getir ve güncelle süreçlerinin birleştirilmesi
Hata/doğrulama mesajları için fazladan çıkış hareketi	Sorgu ve detaylı listeleme süreçlerinin birleştirilmesi
Güncelle/sil süreçlerinden önce listeleme sürecinin unutulması	Sil/güncelle FS'lerinden önce getir FS'si varsayılması
Güncelle/sil süreçlerinden önce getir sürecinin unutulması	Getir sürecinin farklı bir sürecin parçası olarak tanımlanması
Yazma'dan önce okuma varsayılması	Art arda silme işlemlerinin yok sayılması
Fonksiyonel süreç olmayan işlemlerin fonksiyonel süreç sayılması	Açılır form listelerinin oluşturulması için kullanılacak çıkışların yok sayılması
Sadece ebeveyn-türlerin kullanılması	Özelliklerin veri grubu olarak varsayılması
Tetikleyici girdinin belirtilmemesi	Geçici veri gruplarının yok sayılması
Sadece alt-türlerin kullanılması	Çoklu sayfaların farklı FS olarak sayılması
Koşullu durumlar için fazladan FS/VH eklenmesi	Sorgu sonuçları için çıkış hareketinin olmaması
Parametre tablolarının İO sayılması	

Modelin birinci aşamasında sağlanacak olan varlık-ilişki diyagramı, ilgi objeleri arasındaki bağımlılıkların görülmesine olanak sağlayacak, böylece olası art arda işlem ihtiyaçları ölçüm sırasında gözden kaçamayacaktır. Örnek olarak şirket, proje ve çalışan varlıklarından ve aralarında tanımlı ilişkilerden oluşan bir diyagramda (Şekil 2), kullanıcının şirket varlığıyla ilgili bir silme işlemi yapması durumunda, çalışan varlığıyla ilgili de bir silme işlemi yapması gerekecektir, ancak çalışan varlığıyla ilgili yapılan her işlemin şirket varlığını eklemesi gerekmemeyebilir. Hangi varlıkların hangi varlıklar üzerinde etkileri olduğunu anlamak için varlık-ilişki diyagramındaki nicelik değerleri kullanılabilir. Örnekte her çalışan sadece bir şirkette çalışmak zorundayken, bir şirkette

birden fazla çalışan olabilir, bu nedenle şirket varlığının nicelik değeri 1, çalışan varlığının nicelik değeri N olur. Modelde aynı ilişki içerisinde bulunan varlıklardan nicelik değeri 1 olan varlık üzerinde bir işlem yapıldığında, diğer varlık ile ilgili bir işlem yapılma ihtimali olduğu kullanıcının dikkatine sunulur.

Model tanımlanırken, daha sonra geliştirilecek olan yazılım ihtiyacı göz önüne alınarak, kullanım kolaylığı nedeniyle varlık-ilişki diyagramı varlık-ilişki tablosuna dönüştürülmesinin faydalı olacağı düşünülmüştür. Şekil 2’de görülen örnek varlık-ilişki diyagramından türetilen varlık-ilişki tablosu Tablo 2’de verilmiştir.



Şekil 2 - Örnek varlık-ilişki diyagramı

Tablo 2. - Örnek varlık-ilişki tablosu

Varlık 1	Nicelik 1	Varlık 2	Nicelik 2	İlişki
Çalışan	N	Şirket	1	Çalışır
Şirket	1	Proje	N	Sahibidir
Proje	1	Çalışan	N	İstihdam eder

Modelin ikinci aşamasında FS türleriyle ilgili hatalar hedeflenerek, kullanıcıların veri hareketlerini belirlemeden önce FS türlerini belirlemesi istenir. Modelde temel FS türleri için belirlenmiş hareket şablonları ile kullanıcının girdiği hareketler karşılaştırılarak, olası hatalı hareketlerin girilmesi engellenebilir. Veri hareketi şablonları belirlenen temel FS türleri şu şekildedir: listeleme, getirme, ekleme, güncelleme, silme. Bunlar dışındaki süreçler için “diğer” türü kullanılarak, şablonlardan bağımsız bir şekilde veri girişi yapılabilir. Daha önceki çalışmada [12] tanımlanan hareket şablonları birinci vaka analizi sonrasında güncellenerek Tablo 3’deki şablonlar oluşturulmuş ve doğrulayıcı vaka çalışmasında bu şablonlar kullanılmıştır.

Tablo 3. - Temel FS türleri için veri hareketi şablonları

FS türü	Veri Hareketleri		
	Tetikleyici Girdi	Tekrarlayan kısım	Hata /Doğrulama
Listeleme	E	R-X	X*
Getirme	E	R-X	X*
Ekleme	E*	E-W	X*
Güncelleme	E*	E-W	X*
Silme	E	W	X*
Diğer	E	...	X*

*Bu veri hareketi yok sayılabilir.

Tablo 3'te de görüldüğü gibi veri hareketleri 3 gruba ayrılmıştır. Birinci grupta her FS'in sahip olması gereken tetikleyici girdi bulunmakta ve ölçüm yapan kişiye her FS'nin bir giriş hareketi ile başlaması gerektiğini hatırlatmayı amaçlamaktadır. Ekleme ve güncelleme süreçlerinin tekrarlayan kısımları giriş hareketi ile başladığı için bu süreçlerde tetikleyici girdi kısmı ölçülecek ürüne bağlı olarak yok sayılabilir. İkinci kısımda FS'nin temel işlevini gerçekleştiren hareketler bulunmakta ve FS türüne göre doğru hareketlerin girilmesine olanak sağlamayı amaçlamaktadır. Bu bölümdeki hareketlerin art arda gelen işlem ihtiyacına göre tekrar edecek olması nedeniyle tekrarlayan kısım olarak adlandırılmıştır. Son kısım ise, eğer ölçüm yapılacak kaynaktan belirtilmiş ise, hata ve doğrulama mesajları için bir çıkış hareketi içermektedir. Bu kısmın amacı farklı hata ve doğrulama mesajları için birden fazla hareket girişi yapılmasını engellemektir. Hata mesajı tanımlanmamış olan fonksiyonel süreçlerde bu kısım yok sayılabilir.

Önleyici model oluşturulurken detayları bir sonraki bölümde verilen vaka çalışması yapılmıştır. Yazarların daha önceki yayınlarında [12] açıklanan birinci vaka çalışması ile hem modelin potansiyelini gözlemlemek, hem de modeli geliştirmek için veri toplamak amaçlanmıştır. Bu çalışmada önceki yayından [12] farklı olarak; model elde edilen sonuçlarla olgunlaştırılmış, ODTÜ Enformatik Enstitüsü Yazılım Yönetimi Araştırma Grubu'nun geliştirdiği CUBIT aracı ile [21] bütünleştirilerek yazılım üzerinde çalışabilir hale getirilmiş, bu yazılım kullanılarak doğrulama amaçlı bir vaka çalışması yapılmış ve modelin hata engelleme kapasitesi test edilmiştir.

Önleyici COSMIC adı verilen yazılımda, kullanıcı öncelikle varlık-ilişki tablosunu nicelik değerlerini belirterek sisteme girer, daha sonra fonksiyonel süreç girişi ekranında FS türünü seçerek veri hareketlerini tanımlar. Veri hareketleri girişi Tablo 3'de belirtilen şablonlara uygun şekilde üç farklı ekranda yapılır; birinci ekranda tetikleyici hareket, ikinci ekranda varlık-ilişki tablosu da kontrol edilerek tekrarlayan kısım, son ekranda ise hata/doğrulama mesajlarına ait hareketler girilir. Tekrarlayan kısımda sistem veri hareketi ile ilişkili ilgi objesini varlık-ilişki tablosundan kontrol eder, nicelik değeri 1 olan bir ilişki bulması halinde kullanıcıyı art arda işlem olabileceği konusunda uyarır. Kullanıcının bu kısımda bu uyarı dikkate alarak yeni bir hareket çifti yaratma ya da yok sayarak işleme devam etme seçenekleri bulunur. Tüm süreçte kullanıcı Tablo 3'de belirtilen bazı istisnalar dışında şablonda tanımlı tüm hareketler için bir veri girişi yapmak zorundadır.

4 Vaka çalışmaları

Çalışma süresince iki farklı vaka çalışması yapılmıştır. Bunlardan birincisi modelin temel hatları oluşturulduktan sonra yapılan keşif amaçlı çalışmadır. Bu çalışmada modelin engelleme hedeflenen hata türlerine göre çözüm yöntemlerinin geliştirilmesi amaçlanmıştır. İkinci vaka çalışmasında ise amaç modelin, geliştirilen yazılımın da kullanılmasıyla, doğrulamasının yapılması ve hataları beklendiği gibi önleyip önlemediğini görmektir. Birinci vaka çalışmasının sonuçlarına kadar olan kısım yazarların daha önceki yayınlarında [12] açıklanmıştır, ancak güncellenen model, oluşturulan yazılım ve doğrulama amaçlı vaka çalışması bu çalışmada ilk defa yer almaktadır.

Birinci çalışmada 2’şer öğrenciden oluşan 3 farklı öğrenci grubunun aldıkları ders kapsamında aynı yazılım gereksinim dokümanını kullanarak yaptıkları ölçümler kullanılmıştır. Ölçülen yazılım COSMIC uzmanları tarafından ders öncesinde 463 COSMIC Fonksiyon Nokta (CFP-COSMIC Function Point) büyüklüğünde ölçülmüş bir iş uygulamasıdır. Öğrencilerden alınan ölçüm sonuçları tecrübeli COSMIC ölçücülerinin tarafından incelenerek yapılan hatalar bulunmuş ve sınıflandırılmıştır.

Sınıflandırma sonunda 3 temel hata sınıfı oluşturulmuştur: birinci sınıftaki hatalar FS’ler içerisinde unutulmuş veri hareketlerini, ikinci sınıftaki hatalar art arda işlemlerin unutulmasını, üçüncü sınıftakiler ise kullanıcıların ölçümü yapılan ürünlerle ilgili yorumlarından kaynaklı hataları kapsamaktadır. Üçüncü gruptaki hatalar kullanıcı bakış açısı ve ölçüm yapılan materyalin kalitesine bağlı olduğu için yönlendirici bir yöntemle önlenmesi kolay olmayacaktır, bu tür hataların ölçüm yapan uzmanlara yönelik eğitim ve tecrübe artışına yönelik faaliyetlerle önlenmesi gerekmektedir. Bu nedenle oluşturulan önleme yöntemi için birinci ve ikinci gruptaki hatalar hedef olarak seçilmiştir.

Birinci gruptaki hataları önlemek için FS türlerine bağlı olarak kullanıcıyı önceden belirlenmiş veri hareketi şablonlarıyla yönlendirmek, ikinci gruptaki hatalar için de ölçümü yapılacak yazılıma ait olan bir varlık-ilişki tablosu kullanmak planlanmıştır. Tablo 4’de hangi hata türleri için hangi yöntemin kullanıldığı görülebilir.

Tablo 4. - İlk vaka çalışmasında bulunan hataların sınıflandırılması

Hata	Önleme Yöntemi
Farklı hata mesajlarının farklı çıkış hareketleri olarak sayılması	VH şablonu
Hata/doğrulama mesajları için çıkış hareketlerinin olmaması	VH şablonu
Listele sürecinin farklı bir sürecin parçası olarak tanımlanması	FS türü seçimi
Getir sürecinin farklı bir sürecin parçası olarak tanımlanması	FS türü seçimi
Tetikleyici girdinin belirtilmemesi	VH şablonu
Art arda silme işlemlerinin yok sayılması	Art arda işlem
Listele ve getir süreçlerinin birleştirilmesi	FS türü seçimi
Getir ve güncelle süreçlerinin birleştirilmesi	FS türü seçimi
Sorgu ve detaylı listeleme süreçlerinin birleştirilmesi	FS türü seçimi
Yazma’dan önce okuma varsayılması	VH şablonu
Hata/doğrulama mesajları için fazladan çıkış hareketi	VH şablonu *
Güncelle/sil süreçlerinden önce getir süreci olduğu varsayılması	FS türü seçimi *
Güncelle/sil süreçlerinden önce listeme sürecinin unutulması	FS türü seçimi *
Güncelle/sil süreçlerinden önce getir sürecinin unutulması	FS türü seçimi *

* işaretli hatalar ölçüm yapan kişinin yorumuna da bağlı olabilir.

Hataları sınıflandırdıktan sonra, oluşturulacak modelin başarı potansiyelinin görülebilmesi için, “Eğer bu ölçümlerde önleyici model başarılı bir şekilde kullanılsaydı sonuçlar nasıl değişirdi?” sorusu sorulmuştur. Cevabı bulabilmek için ölçüm yapılan gruplardaki hatalar oluşturulan sınıflandırma tablosuna göre gruplandırılıp yorum hataları dışındaki hataların önerilen model tarafından önlenebileceği varsayılmıştır. Tablo 5’den de görülebileceği gibi bu varsayımlar gerçekleşir ve hedeflenen hatalar önlenirse

yapılan hataların takım 1 için %68, takım 2 için %98, takım 3 için de %90 oranında engellenebileceği öngörülmektedir.

Tablo 5. - İlk vaka çalışması sonuçları

		Takım 1	Takım 2	Takım 3
Beklenen İyileştirme	Toplam hata sayısı (A)	38	45	115
	VH hataları (B)	6	29	97
	Art arda işlem hataları (C)	20	15	7
	Yorum hataları (D)	12	1	11
	Beklenen iyileştirme oranı (E)	%68	%98	%90

Vaka çalışmasında benzer eğitimleri almış kişilerin yaptıkları ölçümler analiz edilmiş olmasına rağmen sonuçlar arasında görülen farklılıklar eğitim geçmişleri ya da tecrübe seviyeleri ile ilişkili olabilir.

İlk vaka çalışmasından alınan sonuçlar yöntemin uygulamada başarılı olma olasılığının oldukça yüksek olduğunu gösterince model uygulama kolaylığı sağlaması için ODTÜ Enformatik Enstitüsü Yazılım Yönetimi Araştırma Grubu tarafından geliştirilmiş olan CUBIT [21] yazılımına entegre edilerek, Önleyici COSMIC (Preventive COSMIC) modülü oluşturulmuş ve oluşturulan yeni modül kullanılarak, modelin doğrulanabilmesi amacıyla ikinci bir vaka çalışması düzenlenmiştir.

Hata önleme yönteminin entegre edildiği yazılım modülü ölçüme başlamadan önce kullanıcının varlık-ilişki tablosunu sisteme girmesini gerektirir, bu tablo sisteme yüklendikten sonra kullanıcı bir FS türü seçerek veri hareketlerini sisteme girmeye başlar. Sistem girilen veri hareketlerinin seçilen FS türü için belirlenen hareket şablonuyla uyumlu olması için kullanıcıyı yönlendirerek hata yapmasını engellemeye çalışır. Seçilen FS için giriş yapılırken sistem aynı zamanda kullanıcının belirttiği ilgi objelerini kontrol eder, eğer girilen ilgi objesinin sisteme yüklenen varlık-ilişki tablosunda nicelik değeri 1 olan bir ilişkisi tanımlanmış ise, kullanıcıyı art arda işlem olasılığıyla ilgili uyarır. Kullanıcı da bu uyarıyı dikkate alarak girişlerine devam eder.

Doğrulama amaçlı yapılan ikinci vaka çalışmasında bir ölçücünün 1 yıl önce ölçtüğü 337 CFP'lik bir yazılım ürününü tekrar, bu sefer önleyici modelin entegre edilmiş olduğu yazılımı kullanarak ölçmesi istenmiş ve alınan sonuçlar bir yıl önceki ölçüm sonuçları ile karşılaştırılmıştır. Tablo 6'da da görüldüğü gibi önleyici yöntemin kullanıldığı ölçümde toplam hata sayısının %90, gerçek büyüklük ile aradaki farkın ise %93 oranında azaldığı görülmüştür.

Tablo 6. - Doğrulama çalışması sonuçları

	Toplam hata sayısı	Gerçek büyüklük ile ölçülen büyüklük arasındaki fark
İlk ölçüm (A)	62	28
İkinci ölçüm (B)	6	2
İyileştirme (C)	%90	%93

Önleyici modelin kullanıldığı ölçümde, kullanıcının yaptığı hatalar incelendiğinde (Tablo 7), önceki ölçüme göre çok daha az hata yapılmış olmasına rağmen, iki çeşit hatanın engellenemediği görülmüştür. Bu hatalar incelendiğinde kullanıcının ölçümü yapılan ürünlerle ilgili yorumlarına bağlı hatalar olduğu gözlenmiştir.

Tablo 7. - Doğrulama çalışmasında karşılaşılan hata sayıları

	Toplam büyüklük (CFP)	Hata tanımı	Toplam etkilenen büyüklük	Hata sayısı
İlk ölçüm	365	Eksik "Listeleme" FS'si	-30	9
		Birleştirilmiş Getirme FS'si	0	11
		Eksik "Getirme" FS'si	-3	1
		Eksik "Ekle" FS'si	-19	2
		Eksik VH	-3	2
		Birleştirilmiş FS	-36	3
		Doğrulamadan önce gereksiz VH	+20	10
		Silmeden önce gereksiz VH	+24	12
		Diğer gereksiz VH	+75	12
İkinci ölçüm	335	Güncellemeden önce eksik getirme FS'si	-6	2
		Gereksiz hata/doğrulama FS'si	+4	4
Beklenen büyüklük	337			

5 Sonuç ve gelecek çalışmalar

Özellikle yazılım projelerinin yönetimi sürecinde önemi tartışılmaz olan büyüklük ölçümü sırasında yapılan hataları önlemek ölçüm doğruluğunu arttıracaktır. Bu çalışmada kullanıcıları ölçüm sırasında yönlendirerek sık görülen hataların oluşmasını engelleyecek bir yöntem oluşturmak amaçlanmıştır.

Oluşturulan yöntemi test etmek için yürütülen vaka çalışmalarında yöntemin kullanılması sonucunda hataların %68 ve %98 arasındaki oranlarda azalabildiğini görülmüştür. Bu sonuçlara göre önerilen model hataları azaltmak için başarıyla kullanılabilir.

Yürütülen vaka çalışmaları yöntemin başarılı olduğunu gösterse de, yöntemin başarı oranıyla ilgili genelleme yapılabilmesi için vaka çalışmalarının farklı organizasyonları ve farklı seviyeden ölçücüleri içerecek şekilde genişletilerek yeni çalışmalar yürütülmesi gerekmektedir.

Oluşturulan önleyici COSMIC aracının geliştirilmesi için; kullanıcıların sisteme girmesi gereken varlık-ilişki tablosu yerine gelecekte hali hazırda bulunan bir varlık-ilişki

diyagramının kullanılmasına yönelik deęişiklikler yapılabilir, ölçüm sonuçlarının güncellenmesi sırasında kullanıcıyı yönlendirebilecek yetenekler eklenebilir.

Çalışmanın sonucu olarak COSMIC fonksiyonel büyüklük ölçümlerinin yapılması sırasında sık görülen hataları önleyen bir yöntem ve bu yöntemin uygulanmasına yardımcı olacak bir yazılım aracı geliştirilerek kullanıma sunulmuştur.

6 Kaynaklar:

1. A. J. Albrecht, "Measuring Application Development Productivity," in *Proceedings of IBM Application Development Symposium*, 1979, pp. 83–92.
2. ISO/IEC, "20926: Software and systems engineering -- Software measurement -- IFPUG functional size measurement method." 2009.
3. ISO/IEC, "20968: Software engineering -- Mk II Function Point Analysis -- Counting Practices Manual." 2002.
4. ISO/IEC, "19761: Software engineering -- COSMIC: a functional size measurement method," 2011.
5. ISO/IEC, "24570: Software engineering -- NESMA functional size measurement method version 2.1 -- Definitions and counting guidelines for the application of Function Point Analysis," 2005.
6. ISO/IEC, "29881: Information technology -- Systems and software engineering -- FiSMA 1.1 functional size measurement method." 2010.
7. A. Abran, "Automating Functional Size Measurement – a Survey," in *UKSMA/COSMIC Conference 2011- 22nd Annual conference on Metrics and Estimating: hosted in collaboration with COSMIC*, 2011.
8. A. Abran, J.-M. Desharnais, S. Oigny, D. St-Pierre, and C. Symons, "COSMIC Method Measurement Manual Version 3.0.1.," (COSMIC), The Common Software Measurement International Consortium, 2009.
9. O. Turetken, O. O. Top, B. Ozkan, and O. Demirors, "The Impact of Individual Assumptions on Functional," pp. 155–169, 2008.
10. E. Urgan, O. Demirors, and Ö. Ö. TOP, "Evaluation of Reliability Improvements for COSMIC Size Measurement Results," in *IWSM / MetriKon / Mensura 2010*, 2010.
11. "The Common Software Measurement International Consortium (COSMIC): Guideline for Assuring the Accuracy of Measurements Version 1.0.," 2011.
12. M. Salmanoglu and O. Demirors, "Exploration of an Error Prevention Model for COSMIC Functional Size Measurement Method," *Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2012 Joint Conference of the 22nd International Workshop on*, p. 7, 2012.
13. CMMI Product Team, *CMMI® for Development, Version 1.3 CMMI-DEV, V1.3*, no. November. 2010, p. 468.
14. Ç. Gencil, L. Buglione, O. Demirors, and P. Efe, "A case study on the evaluation of cosmic-ffp and use case points," in *3rd Software Measurement European Forum*, 2006, pp. 121–140.
15. "OVERVIEW of Function Points," *Total Metrics*, 2013. [Online]. Available: <http://www.totalmetrics.com/function-point-resources/what-are-function-points>. [Accessed: 30-Jul-2013].
16. G. Low and D. Jeffery, "Function points in the estimation and evaluation of the software process," *Software Engineering, IEEE Transactions on*, vol. 16, no. 1, pp. 64–71, 1990.

17. C. Kemerer, "Reliability of function points measurement: a field experiment," *Communications of the ACM*, vol. 36, no. 2, 1993.
18. G. Yılmaz, S. Tunalilar, and O. Demirors, "Towards the Development of a Defect Detection Tool for COSMIC Functional Size Measurement," *Software Measurement and the 2013 Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2013 Joint Conference of the 23rd International Workshop on*, pp. 9–16, Oct. 2013.
19. O. O. Top, O. Demirors, and B. Ozkan, "Reliability of COSMIC Functional Size Measurement Results: A Multiple Case Study on Industry Cases," *2009 35th Euromicro Conference on Software Engineering and Advanced Applications*, pp. 327–334, 2009.
20. E. Ugan, O. Demirors, Ö. Top, and B. Özkan, "An Experimental Study on the Reliability of COSMIC Measurement Results," in *International Conferences IWSM 2009 and Mensura 2009 Amsterdam, The Netherlands, November 4-6, 2009. Proceedings*, 2009, pp. 321–336.
21. "CUBIT." [Online]. Available: <http://smrg.ii.metu.edu.tr/cubit>. [Accessed: 25-Jul-2013].