

Efor Kestirim Doğruluğu İçin Tasarım Büyüklüğü Ve Problem Büyüklüğü Karşılaştırılması

Barış Arman Tabak¹ Onur Demirörs²

^{1,2} Enformatik Enstitüsü, Orta Doğu Teknik Üniversitesi, Ankara, Türkiye

¹baristabak@gmail.com ²demirors@metu.edu.tr

Özet. Yazılım sektöründe hatalı efor kestirimleri maliyeti önemli ölçüde arttırır ve projelerin başarısız olmasına yol açabilir. Belirgin maddi kayıplara ek olarak, ürünü hem satın alan hem de üreten organizasyonlar, çalışan demotivasyonu ve benzeri organizasyonel sorunlar yüzünden negatif etkilenirler. Yazılım kestirim uzmanları kestirim doğruluğunu arttırmak için birçok yöntem geliştirmiştir. Gereksinim analizi aşamasında, problem büyüklüğü esas alınarak kestirim yapmak projenin toplam maliyeti için bir fikir oluştursa da eforun büyük bir kısmı tasarım, kodlama ve sınaama fazlarında gerçekleşmektedir. Tasarım aşamasında ve sonrasında kestirim yapmak önemli kazançlar doğurur. Ardıl etkileşim diyagramları sistemin dinamik davranışını yansıttığı için efor kestirimlerini bu diyagramlara bağlı olarak yapmak doğru sonuçlar verebilir. Bu bildiri, tasarım fonksiyonlile büyüklüğünü problem fonksiyonlile büyüklüğüyle karşılaştırmak için yapılan araştırmayı sunmaktadır. Bu doğrultuda bir yazılım organizasyonundaki 8 proje için kestirim modelleri geliştirildi. Yapılan durum çalışması sırasında farklı regresyon modelleri ve yapay sinir ağları metodolojisini kullanarak en iyi kestirim modeli bulundu ve kestirim modellerinin doğruluğunu ölçmek için MMRE (Mean-Magnitude of Relative Error) ve PRED(30) değerleri kullanıldı.

Anahtar Kelimeler. Fonksiyonel Büyüklük Ölçümü, COSMIC, Ardıl Etkileşim Diyagramları, MMRE, PRED

1 Giriş

Efor kestirimi yapmak proje yönetiminin en temel etkinliklerinden biridir. Başarılı efor kestirimi yapmak yazılım proje planlarını daha gerçekçi yapar ve projelerin başarısızlıkla sonuçlanması olasılığını düşürür. Yanlış tespit edilen riskler, bütçe, takvim ve kaynaklar yüzünden iyi yapılmayan efor kestirimleri, hem alıcı hem de satıcı organizasyonlar için problem teşkil eder.

Projelerin gereksinim analizi fazında proje büyüklüğünü ölçtüktan sonra daha kesin kestirimler yapılabilir çünkü büyüklük ölçümleri efor kestirimleri için girdi

olabilir. Birçok efor kestirim yöntemi olduğu için en iyi olanı seçmek projenin geleceğini belirler.

Fonksiyon nokta efor kestirimi yaparken kullanılan bir ölçüvdir. Fonksiyon noktası analizi ise bir büyüklük bazlı efor kestirim modelidir ve gereksinim analizi fazında, yani projenin baş safhalarında bu yöntemi kullanmak akılcıdır. COSMIC, FISMA, IFPUG ve Mk II fonksiyonel büyüklük ölçme yöntemleridir ve ISO standardı olarak tanımlanmışlardır. Diğer yandan, kaynak kod satır sayısı (KKSS) yazılımın büyüklüğünü ölçmek için çok iyi bir ölçüvdir ancak efor kestirimleri için kullanışlı olduğu söylenemez çünkü KKSS kod yazıldıktan sonra elde edilir ve şelale tarzı bir yazılım geliştirme yaşam döngüsünde efor kestirimi için anlamlı bir etkisi olmaz.

Araştırmalar, efor kestirimleri yaparken fonksiyon nokta analizi yöntemleri kullanılınca en fazla kestirim hatasının kodlama ve sınaama fazlarında ortaya çıktığını, nedeninin ise bu fazların kestirilenden daha uzun sürdüğünü göstermektedir[1]. Bu yüzden, şelale tarzı yazılım geliştirme yaşam döngüsünde kodlama fazından önce efor kestirimi yapmak yüksek kestirim hata oranı problemini çözebilir. Proje büyüklüğünü ölçtükten sonra tasarım elemanlarını kullanarak efor kestirim hata oranını düşürmek mümkün olabilir çünkü şelale yazılım geliştirme yaşam döngüsünde bu aktivite kodlama ve sınaama fazından önce yapılır.

COSMIC, 1998'de bir grup tecrübeli yazılım büyüklük ölçücüsü tarafından ortaya çıkarılan, fonksiyonel kullanıcı ve kalıcı bellek arasındaki veri hareketlerine dayanan en popüler fonksiyonel büyüklük ölçme yöntemlerinden biridir. Amacı, alanı iş uygulamaları, gerçek zamanlı uygulamalar ya da bu ikisinin kombinasyonu olan yazılımların fonksiyonel büyüklüğünü ölçmektir[2]. COSMIC, ayrıca ISO/IEC 19761:2011'de tanımlar ve etkinlikler gibi bazı özelliklerle standartlaştırılmıştır.

UML (Unified Modeling Language) ardıl etkileşim diyagramları somut değişkenler arasındaki etkileşimi gösteren diyagramlar olup UML işbirliği diyagramları ile denktir ve sistemin bir biriminin dinamik davranışını gösterir. Alınan ve verilen mesajlar ardıl etkileşim diyagramları ile sıralı olarak gösterilir. Bu diyagramların en temel amacı yazılım geliştiricilere işin nasıl yürüdüğünü nesnelere göstermektir.

Bu araştırmanın ve durum çalışmasının amacı regresyon analizleri yaparak COSMIC yöntemi kullanılarak elde edilen efor kestirim modelleriyle projelerin ardıl etkileşim diyagramlarını kullanarak elde edilen modeli arasındaki farkları ortaya çıkarıp daha iyi olan yöntemi bulmaktır.

Çalışmaya başlamadan önce Türkiye'nin önde gelen bir yazılım organizasyonu tarafından geliştirilen 8 proje seçildi ve bu projeler hakkında veriler toplanılarak yararlı olanlar elde edildi. Her bir projenin fonksiyonel büyüklükleri yazılım gereksinim dokümanlarının yardımıyla COSMIC Fonksiyon Nokta (CFN) birimi cinsinden ölçüldü. Sonrasında basit doğrusal, çoklu doğrusal, polinomik, üstel, logaritmik ve güç regresyon ile yapay sinir ağları modelleri kullanılarak analizler yapıldı. Ortaya çıkan modeller için MMRE ve PRED(30) değerleri hesaplandı ve modellerin kabul edilebilirliğine karar verildi. Ardıl etkileşim diyagramlarındaki hareket (mesaj) sayıları hesaplandıktan sonra ise belirtilen regresyon modelleri tekrar kullanıldı ve MMRE ve PRED(30) değerleri ardıl etkileşim diyagramlarındaki hareket sayıları için hesaplandı.

Bildirinin geri kalanında ise 2. bölümde konuyla ilgili yapılan arařtırmalar, 3. bölümde durum çalıřmasının tasarımı, planı, gerçekteřtirilmesi hakkındaki detaylar, 4. bölümde durum çalıřmasından alınan sonuçlar ve gelecek çalıřmaları hakkındaki bilgilendirmeler yer almaktadır.

2 Yapılan Arařtırmalar

Bévo, Levesque ve Cao'nun 2008'deki C3S2E konferansında sunduđu arařtırmaya göre COSMIC fonksiyon nokta ile ardıl etkileřim diyagramlarındaki mesaj sayısı arasında güçlü bir iliřki vardır. Çalıřmanın amacı yazılım büyüklük tahminini UML diyagramlarına uyarlamaktır; bu yüzden arařtırmacılar büyüklük ölçüm yöntemlerini UML kullanıcı senaryosu ve ardıl etkileřim diyagramlarında uyguladılar. Sonuç olarak, aynı uygulama için COSMIC fonksiyon nokta sayısının ardıl etkileřim diyagramlarındaki mesajlara eřit olduđunu ortaya çıkardılar[3].

Sellami ve Ben-Abdallah kullanıcı senaryosu açıklamasına göre fonksiyonel büyüklük ölçmeyi gösteren bir arařtırma yapmışlardır. Arařtırmacılara göre, bulunan ölçüler COSMIC fonksiyon noktası cinsinden fonksiyonel büyüklük tahmini yapmak için kullanılabilir. Arařtırmacılar, ayrıca kullanıcı senaryosu diyagramları ile onlara karşılık gelen sınıf ve ardıl etkileřim diyagramlarının arasındaki iliřkinin güçlü olduđunu göstermişlerdir. Bu çalıřmanın sonucu olarak kodlama fazından önce ardıl etkileřim diyagramlarını kullanarak kestirim yapmanın mümkün olduđu yargısına varılabilir[4].

Awan ve çalıřma arkadaşları, 12 yazılım sisteminin verilerini kullanarak yaptıkları arařtırmada varlık-iliřki (Entity-Relationship) diyagramlarındaki varlık sayısı, iliřki sayısı, özellik sayısı ve benzeri özelliklere göre bir yazılım eforu kestirim modeli sunmuşlardır. Arařtırmacılar çoklu regresyon modelini kullanmayı seçmişlerdir ve sonuç olarak kestirilen efor, gerçek efor miktarına oldukça yakın çıkmıştır. 12 proje için ortalama hata oranı %11.54 çıkmıştır. Bu oran, yazılım projeleri için başarılı olarak gösterilebilir. Bu yüzden tasarım fazında yapılan efor ya da maliyet kestirimleri tatmin edici sonuçlar sağlayabilir[5].

3 Durum Çalıřması

3.1 Tasarım

Durum çalıřması 3 alt parçadan oluşmaktadır. Bunlar;

- Veri toplanması: Proje verisinin toplanılıp gerekli olanların edinilmesi
- Ölçüm: COSMIC ölçümü ve ardıl etkileřim diyagramlarındaki hareket sayısının ölçümü
- MMRE analizi: Regresyon analizleri yapıılıp sonuçların deđerlendirilmesidir.

3.2 Plan

Çalışmanın yapılması için 14 haftalık bir plan hazırlanmıştır. Proje verileri elde edildikten sonra gerekli bilgiler alındı. Ölçevlerin güvenilirliği için efor bilgisi, bir araç yardımıyla elde edildi. Aykırı değerlerin ya da projelerin tespit edilmesi için projelerden aşağıdaki bilgiler elde edildi:

- Proje başlama/bitiş tarihi
- Proje türü
- Fonksiyonel alan türü
- Yeni teknoloji kullanımının varlığı
- Kullanılan araçlar
- Geliştirme takımının tecrübesi
- Proje takımının istikrarı
- Proje sırasında kullanılan standartlar
- Platform
- Algoritma karmaşıklığı
- Kaynak kodun yeniden kullanım oranı
- Projelerin toplam eforu
- Proje yönetimi, gereksinim analizi, kodlama, entegrasyon, sınama, sistem kurulum, kalite güvence etkinlikleri için ayrı ayrı efor bilgisi
- Her fazdaki ve proje tesliminden sonra tespit edilen hata sayısı

Problem büyüklüğü ölçmek için COSMIC yönteminin kullanılmasına karar verildi. Bunun için gereksinim analiz dokümanlarının son versiyonları alındı ve doküman sahiplerine doğrulandı. Değişiklik taleplerini de göz önünde bulundurarak büyüklük ölçümü yapılmaya karar verildi.

Problem büyüklüğünü ölçtüktan sonra bir tasarım elemanı olan ardıl etkileşim diyagramlarını kullanarak tasarım büyüklüğünün ölçülmesine karar verildi. Tasarım elemanı olarak ardıl etkileşim diyagramları kullanımının sebebi bu diyagramların sistemin dinamik davranışını yansıtmasıdır ve dinamik davranışlar, statik davranışlardan daha çok şey ifade edebilir. Modelleri oluştururken ardıl etkileşim diyagramlarındaki toplam hareket sayısı kullanıldı ve bu diyagramlar da IDE (Integrated Development Environment) yardımıyla üretildi. Bu sayede temiz veri elde edildi.

Gereksinim analizi fazında 6 farklı türde regresyon modeli ve yapay sinir ağları modelinin her bir proje için ayrı ayrı oluşturulması planlandı. Farklı türler basit doğrusal, çoklu doğrusal, polinomik, üstel, logaritmik ve güç şeklindedir. İlk olarak, 7 modelin CFN sonuçlarına ve her projenin faz bazlı efor verisine dayanarak oluşturulması planlandı. Daha sonra benzer şekillerde her projenin ardıl etkileşim diyagramlarındaki toplam hareket sayısı ve faz bazlı efor verisine dayanarak regresyon modelleri oluşturuldu. Yapay sinir ağları için, çok fazla gizli katman ve düğüm olması kestirim sonuçlarına çok etki etmediği için 3 düğüm ve 1 gizli katman kullanıldı. Bütün modeller her bir projeye uygulandıktan sonra en iyi model seçilerek MMRE ve

PRED(30) deęerleri kullanılarak bu modelin kabul edilebilir olup olmadıęı kontrol edildi.

3.3 Gerçekleřtirme

Veri Toplama. Bu faz, bir durum alıřmasının gvenilirlięinin en nemli dayanaklarından olduęu iin byk nem arz etmektedir. Bu faz, alıřmanın temelini oluřturduęu iinse n iř olarak adlandırılabilir. Yapılan durum alıřmasından doęru ve uygun sonular elde etmek iin verilerin gerek olması gerekir.

Veriler, Trkiye'nin byk bir yazılım organizasyonundan alınmıřtır ve projeler řelale yaklařımıyla geliřtirilmektedir. Efor verileri, bir ara yardımıyla toplanmıřtır ve bu efor verileri alıřan tarafından gnlk olarak girilmektedir. Bu sayede proje yneticileri rapor ve grafikler ıkartarak alıřanların eforlarını ve projenin gidiřatını izleyebilmektedir

Bu durum alıřması iin 8 proje seildi. Btn projeler geliřtirme projesi olup hepsi aynı endstri alanıyla ilgili projelerdir.

lm. Yapılan 2 lmden birisi olan COSMIC lm nceden de belirtildięi gibi problem byklęn lmek iin yapıldı. COSMIC lmnn sonuları Tablo 1'de belirtilmiřtir.

Tablo 1. COSMIC lm Sonuları

Pr oje	Giriř Hareket Sayısı	ıkıř Hareket Sayısı	Oku- ma Ha- reket Sayısı	Yazm a Ha- reket Sayısı	Top- lam Ha- reket Sayısı
A	17	11	13	15	56
B	31	23	12	6	72
C	64	62	74	21	221
D	70	49	52	20	191
E	31	29	39	7	106
F	39	39	44	20	142
G	115	102	96	84	397
H	88	97	77	19	281

Efor ve tasarım elemanları arasındaki iliřkiyi analiz ederek bir efor kestirim modeli oluřturmak iin gerekli lm yntemi olan ardıl etkileřim diyagramındaki hareket sayılarının lmnn sonucu ise Tablo 2'deki gibidir:

Tablo 2. Projelerin Ardıl Etkileşim Diyagram Hareket Sayıları

Proje	A	B	C	D	E	F	G	H
Toplam								
Ardıl								
Etkileşim	11	22	8	6	16	13	37	22
Diyagram	47	47	73	78	60	68	24	16
Hareket								
Sayısı								

MMRE Analizleri. Bu analizler ardıl etkileşim diyagramlarındaki hareket ve yazılım geliştirme yaşam döngüsündeki fazların eforları arasındaki ilişkiyi tanımlayan matematiksel modelleri oluşturmak için yapılmıştır. Model geliştirmenin yanında, bu durum çalışması COSMIC ile ölçüm yapmanın mı, yoksa ardıl etkileşim diyagramları ile ölçüm yapmanın mı daha etkili olduğunu ortaya çıkartıyor. Tablo 3’te her bir projenin efor ve büyüklük bilgileri yer almaktadır.

Regresyon analizi değişkenler arasındaki ilişkileri araştıran bir araçtır [6]. Bu yüzden, regresyon analizi sayesinde bağımsız değişkenlere göre bağımlı değişkenlerin kestirimi yapılabilir. COSMIC ölçüm girdisi olan çoklu regresyon ve yapay sinir ağları modelleri geliştirilirken, diğer modellerin aksine 4 değişken girdi olarak verildi. Bu değişkenler giriş hareket sayısı, çıkış hareket sayısı, okuma hareket sayısı ve yazma hareket sayısıdır. Diğer tekniklerde toplam COSMIC büyüklüğü ve toplam ardıl etkileşim diyagram hareket sayısı girdi olarak alınmıştır.

Kestirim yaparken tasarım büyüklüğünün (ardıl etkileşim diyagram hareket sayısı), problem büyüklüğünden (COSMIC fonksiyon nokta sayısı) daha etkili olacağı düşünüldüğünden dolayı, ardıl etkileşim diyagram hareket sayısı; tasarım, kodlama ve sinama aktivitelerinin hepsinin toplam eforunun kestirim modelini oluşturmak için kullanmakla beraber kodlama ve sinama aktivitelerinin toplam eforunun kestirim modelini oluşturmak için de için kullanılmıştır.

Yapay sinir ağları yöntemi ise bir makine öğrenimi ve veri madenciliği yöntemi olup beyindeki nöronların yapısından esinlenilmiştir. Ardıl etkileşim diyagramında sadece 1 bağımsız değişken olduğu için (toplam hareket sayısı), pratikte mantıklı olmamasına rağmen yapay sinir ağlarına girdi olarak bu değişken verilmiştir. MRE (Magnitude of Relative Error), MMRE (Mean Magnitude of Relative Error) kestirim doğruluğu için sık kullanılan ölçülerdir. MRE tekil kestirim hata oranıdır, MMRE ise bir veri kümesinde göreceli hataların yüzde olarak ortalamasıdır.

PRED(n) kestirim doğruluğu için bir başka önemli ölçüdür. PRED(n), kestirimlerin yüzde kaçının n% hata oranı içinde olduğunu gösterir. Örneğin, PRED(30) = 95 ise yapılan kestirimlerin %95’i gerçek değerden maksimum yüzde 30 sapmıştır. PRED değeri hesaplanırken “n” sayısı 30 olarak alındı çünkü bu sayı endüstride kestirim modelleri yaratırken sık kullanılan bir sayıdır.

4 Sonular

Ölümler, MMRE ve PRED deęerlerinin hesaplanmasından sonra en iyi MMRE ve PRED deęerlerini bulmak birincil ama olmuştur. En iyi MMRE en düşük MMRE deęeridir ünkü hata oranının en düşük seviyede olması kestirimler iin en iyi sonucu vermektedir. Dięer yandan en iyi PRED(30) deęeri en yüksek deęerdir ünkü bu deęer geliştiren matematiksel modellerin doęruluęunu göstermektedir.

COSMIC ile problem büyüklüęü ölçüldüęü iin COSMIC deęişkenlerini kullanarak kestirim yaptığımızda en iyi sonucu gereksinim analizi eforunu ve toplam eforu tahmin ettiğimizde elde edilen modeller arasında bulmayı hedefliyorduk. Bunlar arasındaki en iyi sonucu MMRE deęeri 21.03 olan ve PRED(30) deęeri 87.5 olan oklu doęrusal regresyon modeli ile toplam eforun kestirimini yaptığımız zaman elde ettik.

Ardıl etkileşim diyagramlarındaki toplam hareket sayısı ile tasarım büyüklüęünü ölçtüğümüz iin bu sayıları kullanarak kestirim yaptığımızda en iyi sonucu tasarım, tasarım-kodlama, tasarım-kodlama-sınama aktivitelerinin eforunu ve toplam eforu tahmin ettiğimizde elde edilen modeller arasında bulmayı hedefliyorduk. Bunlar arasındaki en iyi sonucu MMRE deęeri 19.00 olan ve PRED(30) deęeri 87.5 olan logaritmik regresyon modeli ile tasarım-kodlama eforun kestirimini yaptığımız zaman elde ettik.

Tasarım, kodlama ve sınama fazları iin yaptığımız kestirimlerde ardıl etkileşim diyagramlarını kullanarak yaptığımız ölçümlere baęlı kestirimlerin, COSMIC ölçümlerine baęlı yapılan kestirime göre daha doęru sonuç vermesini bekliyorduk. Buna benzer bir şekilde gereksinim analizi fazı iin yaptığımız kestirimlerde COSMIC ölçümlerine baęlı kestirimlerin, ardıl etkileşim diyagramlarına baęlı yapılan kestirime göre daha doęru sonuç vermesini bekleniyordu.

Elde edilen sonuçlara göre tasarım, kodlama ve sınama fazı iin ayrı ayrı kestirim yapıldığı zaman elde edilen sonuçlar başarılı olmadı. En yüksek PRED(30) deęeri 62.5 sonucunu verdi. Bu deęer de kestirimler iin kabul edilebilir bir deęer olmadığı iin ve MMRE deęerleri de tasarım, kodlama ve sınama aktivitelerinin sentezlerinden daha başarısız sonuçlar verdiği iin kestirim sonuçlarını ayrı ayrı deęerlendirilmedi.

COSMIC girdilerine göre yapılan gereksinim analizi fazının efor kestirimleri ardıl etkileşim diyagramlarından alınan girdilere göre yapılan kestirime göre daha iyi sonuçlar verdi. Dięer yandan, ardıl etkileşim diyagramlarından elde ettiğimiz girdilerle en iyi sonucu veren logaritmik regresyon modeli ile tasarım-kodlama eforunun kestiriminin doęruluęunun COSMIC girdileriyle en iyi sonucu veren modelden daha kötü olması beklenmeyen bir sonuç oldu ünkü bir tasarım elemanı olan ardıl etkileşim diyagramlarından ıkartılan girdiler ile tasarım ve sonraki fazlar iin yapılan kestirimlerin daha doęru olması bekleniyordu.

Gelecek alışmalarda 8'den fazla sayıda projeyle analizi gerekleştirerek daha iyi bir model oluşturmak hedefleniyor. Böylelikle, daha ok veri toplanıp daha fazla ölçüm yapılarak daha doęru sonuçlar veren modeller oluşturulabilir. Bu modeller, proje yönetim etkinliklerinde daha etkili olur. Gelecek alışmalarda yapılabilecek daha da önemli bir iş ise ardıl etkileşim diyagramlarındaki toplam hareket sayısını asenkron giren aęrı, senkron giren aęrı, asenkron ıkan aęrı, senkron ıkan aęrı,

asenkron giren mesaj, asenkron çıkan mesaj, alternatifler, döngüler, seçimliler ve ardıl etkileşim diyagramının karmaşıklığı şeklinde bölerek eforlar ve bu seçenekler arasındaki ilişkiyi bire bir bulup ardıl etkileşim diyagramlarının eforlar üzerindeki etkisi daha etkili bir şekilde bulunabilir.

Kaynaklar

1. Abrahamsson, P., Moser, R., Pedrycz, W., Sillitti, A., Succi, G. : Effort Prediction in Iterative Software Development Processes – Incremental Versus Global Prediction Models, In First International Symposium on Empirical Software Engineering and Measurement, pp 344-353, IEEE Press, New York (2007)
2. COSMIC, Measurement Manual, Version 3.0.1, Common Software Measurement International (2009)
3. Levesque, G., Bevo, V., Cao, D.T.: Estimating Software Size with UML Models. In C3S2E Conference, pp. 81-87, ACM, New York City (2008)
4. Sellami, A., Ben-Abdallah, H.: Functional Size of Use Case Diagrams: A Fine-Grain Measurement. In Fourth International Conference on Software Engineering Advances, pp. 282-288, IEEE Press, New York (2009)
5. Arshid, A., Qadri, S., Muhammad, S. S., Abbas, J., TariqPervaiz, M., Awan, S.: Software Cost Estimation through Entity Relationship Model, Report and Opinion, vol. 2, pp. 36-40, Marsland Press, New York City (2010)
6. Sykes, A. O.: An Introduction to Regression Analysis, Chicago Working Paper in Law & Economics. Chicago (1992)