

COSMIC İşlevsel Yazılım Büyüklüğü Ölçüm Yönteminin Kurumlarda Uygulanmasında Dikkat Edilmesi Gereken Noktalar

Murat Salmanođlu¹, Ali Yıldız², Onur Demirörs¹

¹ODTÜ Enformatik Enstitüsü, Ankara, Türkiye
{musalman, demirors}@metu.edu.tr

²Bilgi Grubu, Ankara, Türkiye
ali.yildiz@bg.com.tr

Özet. Yazılım projelerinin başarı ile yönetilebilmesi amacıyla iş gücü, bütçe, takvim gibi önemli kısıtlar için proje yaşam döngüsü başlangıcında yapılacak kestirimler oldukça önemlidir. Yazılım işlevsel büyüklüğü bu kestirimler için kullanılacak objektif ve karşılaştırılabilir önemli bir girdidir. Bu çalışmada yazılım işlevsel büyüklük ölçüm yöntemlerini uygulamaya karar veren kurumların bu yöntemleri süreçlerinde başarılı bir şekilde uygulayabilmeleri için dikkat etmeleri gereken noktalardan bahsedilmiştir.

Anahtar kelimeler: İşlevsel büyüklük ölçümü, COSMIC, iş gücü kestirimi

1 Giriş

Proje yönetiminde en önemli noktalardan bir tanesi projeye başlamadan önce iş gücü, maliyet, bütçe, takvim gibi proje için kritik öneme sahip kısıtlarla ilgili kestirimleri güvenilir yöntemler kullanarak olabildiğince gerçeğe yakın öngörebilmektir. Bu tür kestirimlerin, basit tahminlerden öteye giderek, kabul edilebilir sınırlar içerisinde yapabildiği için objektif ölçütlere dayalı yöntemler kullanılması gerekmektedir. Nispeten genç bir dal olan yazılım mühendisliğinde bu tür kestirimler için geleneksel olarak uzman görüşü, karşılaştırma, benzetme gibi tecrübeye dayalı yöntemler [1] kullanılmıştır. Tecrübeye dayalı bu yöntemler hızlı gelişen teknoloji ve sürekli yenilen iş gücü karşısında yetersiz kalmaktadır. Bunlar yerine kullanılacak objektif ölçütlere ve bu ölçütlere bağlı algoritmalara dayanan yaklaşımlar [1] modern yazılım mühendisliği dünyasının ihtiyacını karşılamak için daha uygundur .

Objektif kritere dayalı ölçütler içinde en bilineni kaynak kod satır sayısı olmasına rağmen bu ölçütün kullanımı proje yönetimi açısından uygun değildir. Bunun en önemli nedeni, kod satır sayısını doğru olarak belirlemek için yazılım ürününün tamamlanmış olması gerekliliğidir. Bunun dışında, farklı kodlama alışkanlıkları, farklı kodlama dillerinin kullanımı, yazılım kodlarını otomatik oluşturan teknolojilerin yaygınlaşması gibi nedenlerle kod satır sayısı yazılımın büyüklüğüyle ilgili bir ölçüt olarak artık kabul görmemektedir.

Hedeflediğimiz amaçlara uygun bir şekilde kullanılabilir, dünya çapında kabul görmüş en yaygın ölçütlerden bir tanesi işlevsel büyüklüktür. Üretilen yazılımın kullanıcılarına sunacağı işlevleri ölçmeyi hedefleyen bu yaklaşım, yazılımın sunacağı hizmetlerin bir ölçütü olduğu için yazılımın nasıl geliştirildiğinden bağımsızdır. Bu nedenle yazılımın hangi kodlama diliyle, hangi yazılım yaşam döngüsü modeli ile ya da hangi teknolojiler ile geliştirileceğinden bağımsız olarak yazılımın büyüklüğünün ölçülebilmesine olanak sağlar.

COSMIC (Common Software Measurement International Consortium); ISO (International Organization for Standardization) standardı yayınlanmış [2] işlevsel büyüklük ölçüm yöntemleri içerisinde en genç yöntem olmasına rağmen, diğer yöntemlere göre kullanım ve öğrenme karmaşıklığı en düşük, en geniş kapsama sahip, ve otomatikleştirmeye daha uygun bir yöntem [3] olması nedeniyle özellikle işlevsel büyüklük ölçümüne yeni başlayacak kurumlar tarafından tercih edilmektedir [4].

COSMIC kullanmaya karar veren kurumların, yöntemden en iyi verimi alabilmeleri için uygulamaya geçmeden önce yöntemin uygulanmasının süreçlerine uyumu ile ilgili çalışmaları yürütmeleri gerekmektedir. Bu makalenin ikinci bölümünde COSMIC yönteminin var olan süreçlere uyumu çalışmaları sırasında karşılaşılan zorluklardan ve bunları aşmaya yönelik çözüm önerilerinden bahsedilecektir. Bu makalede anlatılanlar COSMIC yönteminin uygulaması sırasında edinilen gözlemleri gösterse de tüm işlevsel büyüklük yöntemleri için benzer gözlemlerden bahsedilebilir.

2 Geçmiş Çalışmalar

İşlevsel büyüklük analizi yöntemi Albrecht'in ilk defa önerdiği günden [5] günümüze kadar yazılım çevrelerinde kullanılmaktadır. Şu an ISO'nun yayınladığı bir işlevsel büyüklük ölçümü standardı [6] ve 4 farklı ölçüm yöntemi için standartlar [2], [7]–[9] mevcuttur. Her ne kadar bu çalışma kapsamı tüm işlevsel büyüklük ölçüm yöntemleri için kabul edilebilir olsa da, yazarların tecrübesi COSMIC [10] iledir.

İşlevsel büyüklük ölçüm yöntemlerinin kurumsal süreçlere uyumu sırasında bu yöntemlerin kullanımıyla ilgili tereddütler olabilir [11]. Kurumların bu tereddütleri gidermeye yönelik doğru bilgilere ulaşarak aşmaları gerekmektedir. Geçmiş literatür incelendiğinde işlevsel büyüklük ölçümünün doğruluğu ile ilgili bir çok yayına [12][13][14][15][16] ulaşılabilmesine rağmen, yöntemin süreçlere uyumu konusunu doğrudan inceleyen bir yayına rastlanamamaktadır. Bunun bir nedeni de kurumların bu konudaki süreç iyileştirme tecrübelerini ticari nedenlerle yayınlamak istememeleri olabilir.

3 Süreçlere Uyumda Dikkat Edilmesi Gereken Noktalar ve Öneriler

Bu bölümde işlevsel büyüklük ölçümü yöntemini süreçlerinde kullanmak isteyen organizasyonların dikkat etmeleri gereken noktalar ve öneriler detaylandırılmıştır (Tablo 1).

3.1 İş gücü bilgilerinin düzgün/tutarlı tutulmaması:

Yazılımın işlevsel büyüklüğünün ölçülmesinin yazılım projeleri yönetimi için en önemli faydalarından bir tanesi gerekli olan iş gücü, maliyet ve takvim gibi verilerin proje yaşam döngüsü başında kabul edilebilir sınırlar içerisinde kestirilebilmesine olanak sağlamasıdır. Bu kestirimler yapılırken kurumun geçmiş projeleri ölçülüp elde edilen büyüklükler bu projelere ait iş gücü verileriyle birlikte kullanılarak kurum için bir iş gücü modeli oluşturulur. Bu model yeni projelerde ihtiyaç duyulacak iş gücünün ve buna bağlı olarak maliyet ve takvimin kestiriminin yapılmasına olanak sağlar.

Yazılım işlevsel büyüklüğüne bağlı iş gücü modelinin doğru çalışabilmesi için bu modelde girdi olarak kullanılan geçmiş iş gücü verilerinin de doğru ve tutarlı olması gerekmektedir. Çalışanlar genel olarak iş gücü verilerini kaydederken gerek zorunluktan gerekse yeterli altyapı olmayışından dolayı elektronik postalarını cevaplamak, telefon görüşmesi yapmak gibi proje dışı bazı faaliyetlere harcadıkları iş gücünü de ayrı kaydetmek yerine proje faaliyetine kaydetmektedir. Bazı durumlarda iş gücü verisini toplayan sistemlere veri girişi anlık (günlük) değil uzun süreli (haftalık, aylık) olarak topluca yapılmaktadır, bu durumda çalışanların geçen zamandan dolayı farklı projelere harcadıkları iş gücünü doğru olarak giremeyişleri nedeni ile kayıtların kesinliği azalmaktadır. Hatta bazı durumlarda çalışanlar haftalık işgüçlerini çalıştıkları projelere bölerek gerçek veriden tamamen uzak girişler yapabilmektedir. Bu tür hatalı kayıtlar ilgili projelere ait toplam iş gücü verilerinin hatalı tutulmasına, sonuç olarak da oluşturulacak iş gücü kestirim modellerinde hataya sebep olmaktadır.

Kestirim modellerinin hata payının azaltılabilmesi için kurumlar iş gücü verisinin düzgün/tutarlı kaydedilmesi için kurum yapısına ve süreçlerine uygun bir alt yapısı oluşturulmalı ve bu verilerin düzgün/tutarlı kaydedildiğinden emin olunmalıdır. İş gücü verilerinin girilirken proje dışı faaliyetler için de veri girilmesi sağlanmalı ve gerçekleşen iş gücü verilerinin günlük ve kesinlikli girilmesini sağlayacak bir yapı oluşturulmalıdır. Girilen iş gücü verilerinin kullanım amacı çalışanlarla paylaşılmalı, çalışanların hiç mola vermeden günlük mesailerinin tamamını proje faaliyetlerine harcamalarını beklemenin anlamsızlığının farkında olarak proje dışı faaliyetler için girilen iş gücü verilerinin ceza amaçlı kullanılmayacağı konusunda güvence sağlanmalıdır.

İşgücü modeline girdi olarak kullanılacak verilerin tutarlı olması oluşturulacak modelin hata payını azaltarak kurumların uzun vadede tutarlı kestirimler yaparak proje yönetim faaliyetlerinin etkinliğini artırma şansı verecektir.

3.2 Kurumsal ölçüm yönteminin oluşturulmaması:

COSMIC yöntemi kurallarına uygun olarak yapılan ölçümlerin ölçen kişiden bağımsız olarak aynı sonucu vermesi beklenmektedir. Bu özellik yöntemin en önemli avantajlarından birisidir. Ancak bazı durumlarda ölçüm yapan kişilerin kuralları farklı/yanlış yorumlamaları nedeniyle ölçümler arasında farklılıklar görülebilmektedir. Bu tür farklılıkların önüne geçebilmek için, çalışanlara yeterli COSMIC eğitiminin verilmesinin yanı sıra, kurumsal ölçüm kuralları tanımlanarak çalışanlarla paylaşılmalıdır.

Kurumsal ölçüm kuralları COSMIC ölçüm kılavuzunda tanımlanan kurallardan farklı olarak, bu kuralların kurumun alan bilgisiyle harmanlanarak çalışanların yoru-

ma gerek kalmadan rahatlıkla anlayabilecekleri bir dille tanımlanması ve çalıştıkları alanlarla ilgili örneklerle zenginleştirilmesiyle oluşturulacaktır. Örneğin COSMIC ile ölçüm yaparken çalışanlar ilgi nesnesi [10] kavramını anlamakta zorluk yaşamakta, özellikle yöntemi öğrenmeye başladıkları ilk dönemlerde nasıl tanımlayacaklarıyla ilgili yardıma ihtiyaç duymaktadırlar. Bu konuda yardımcı olmak için çalışılan alanlarla ilgili ilgi nesneleri kurum genelinde listelenip, yapılacak ölçümlerde bu listenin kullanılması sağlanabilir. Tüm uygulamada ilgi nesnelerinin doğru belirlenmiş olması ölçüm doğruluğunu arttıracak ve hatalı kestirimlerin önüne geçecektir.

İlk aşamalarda ortak bir kural seti geliştirmek için ortak çalışmalar ve çalıştaylar yapılması gerekeceğinden harcanan iş gücü artacaktır. Ancak bu iş gücü öğrenme ve geçiş döneminin parçası olarak görülmeli ve daha sonraki ölçümlerin doğruluğunu ve tutarlılığını artırarak hataların önüne geçeceği göz önünde bulundurulmalıdır.

3.3 Yeniden kullanım (reuse) oranının net belirlenememesi:

Yazılım geliştiren kurumlarda tekrarlanan işlevler için daha önce mevcut olan gereksinim, tasarım veya kod parçaları uygun olan durumlarda yeniden geliştirilme eforu harcanmadan daha önceki şekillerde kullanılmaktadır. COSMIC ile işlevsel büyüklük ölçümü sırasında tekrarlayan işlevsel süreçler için büyüklük ölçümü hesaplanırken tekrarlamalar göz önüne alınarak ölçüm yapılır. Ancak tekrarlayan özelliklerin daha önce geliştirilen projelerde olması ya da analiz ekibinin tasarım sırasında tekrarlanacak özelliklere hakim olmaması gibi nedenlerle bazı durumlarda yeniden kullanılan özellikler tekrar sayılabilir.

Yeniden kullanılan özelliklerin tekrar sayılması iş gücü modellerinde gereğinden fazla iş gücü kestirimi yapılmasına neden olacaktır. COSMIC ölçümü yapılırken yeniden kullanım oranlarının ölçüm yapan ekip tarafından biliniyor olması bu hataların kolayca engellenmesini sağlayacaktır. Yeniden kullanım oranlarının doğru hesaplanabilmesi için yazılım geliştirme süreçlerinde gerekli düzenlemeler yapılmalı, yeniden kullanıma ilerleyen aşamalarda karar verilen durumlarda da konuyla ilgili paydaşlar arasında iletişim kurularak ölçümün güncellenmesi sağlanmalıdır. COSMIC ölçüm rehberinde de belirtildiği gibi kurumlar yeniden kullanım oranlarını belirledikten sonra yeniden kullanılacak olan bölümün türüne ve büyüklüğüne göre iş gücü kestirim modellerini güncelleyebilirler.

3.4 İş gücü bilgilerinin kişi üretkenliğini belirlemede kullanılması:

İşlevsel büyüklük yazılım projeleri için önemli olan birçok konuda kurumlara kolaylıkla uygulanabilecek çözümler sunmaktadır. Bu yöntemi benimseyen kurumların bu çalışmada da özetlenen faydalara ek olarak bu yöntemi çalışanların bireysel performansını ölçmek için de kullanma eğilimi gösterdiği gözlenmiştir. Özellikle araştırma ve geliştirme faaliyetlerinde bireysel performans ölçümünün kolay tanımlanabilen nitel ölçütlere bağlanabilmesinin zorluğundan dolayı kurumlar işlevsel büyüklüğü çözüm olarak görmektedirler.

Bu tür bir yaklaşım çalışanlar üzerinde önemli baskılara neden olarak kurum için ciddi problemlere yol açabilir. COSMIC yöntemi ile yazılımın işlevsel büyüklüğünü ölçebiliriz ancak bu büyüklük yazılımın kullanıcıya sağladığı faydanın büyüklüğüdür. Yazılım geliştirme sürecine dahil olan çalışanlar işlevsel süreçlerin geliştirilmesine ek

olarak çok farklı görevler üstlenebilmektedir. Örneğin yazılımın performansını iyileştirmek için çalışan bir geliştiricinin harcayacağı iş gücü yazılımın işlevsel büyüklüğünden bağımsız olabilir. Ya da yazılımın kullanıcı ara yüzünü geliştiren bir görsel tasarımcı, kullanıcıya sağlanan işlevsel faydadan bağımsız olarak kullanıcı deneyimi iyileştirmeye yönelik çalışmalar yapabilir. Bütün bu işlevsel olmayan geliştirme işlemleri yazılımın bütünü incelendiğinde işlevsel büyüklükle ilişkili olabilir ancak bireysel seviyedeki performanslar işlevsel büyüklükle ölçülemez.

İşlevsel büyüklüğü performans yönetimi amacıyla kullanmak isteyen kurumlar iyi tanımlanmış kurallar yardımıyla proje ekipleri için performans yönetim modelleri oluşturabilirler, ancak bu yaklaşımın bireysel performans ölçümü için kullanılması hem kurum hem de çalışanlar için olumsuz sonuçlar doğurabilir.

Tablo 1. İşlevsel büyüklük ölçümünde dikkat edilecek noktalar

	İş gücü	Ölçüm yöntemi	Yeniden kullanım	Üretkenlik değerlendirme
Karşılaşılan Sorun	İş gücü bilgilerinin düzgün tutarlı tutulmaması	Ölçümlerin kişiye bağlı değişiklik göstermesi	Yeniden kullanım (reuse) oranının net belirlenmemesi	Üretkenlik değerlerinin kişi üretkenliği değerlendirme amacıyla kullanılması
Çözüm Önerisi	Alt yapı oluşturulması Düzgün ve tutarlı veri girildiğinin kontrolü Proje dışı faaliyetlerin de dikkate alınması	Ölçüm yöntemi eğitimlerinin verilmesi Ortak ölçüm yöntemi oluşturulması Ölçüm örnekleri kütüphanesi oluşturulması	Yeniden kullanım etkisinin belirlenmesi Yeniden kullanım durumunda ölçümlerin güncellenmesi Yeniden kullanım etkisi için hesaplama yöntemi oluşturulması	Üretkenlik değerlendirme kriterlerinin belirlenmesi Üretkenlik değerlendirmesinin belirlenen kategoriler içinde bireysel açıdan değil takım olarak yapılması

4 Sonuç

İşlevsel büyüklük ölçümü yazılım geliştirme süreçlerinde ve süreçlerin yönetiminde kuruma önemli faydalar sağlamaktadır. Özellikle kullanımının, öğrenilmesinin ve süreçlere uyumunun kolaylığı nedeniyle COSMIC işlevsel büyüklük ölçüm yöntemi en çok tercih edilen yöntemlerden birisidir. Ancak bu yöntemleri uygulamak isteyen kurumların dikkat etmesi gereken birkaç temel nokta vardır. Dikkat edilmesi gereken noktalar ve çözüm öneriler başlığında detaylı olarak ele alınan bu noktalar, işlevsel büyüklük ölçme yöntemlerini süreçlerinde uygulamaya karar veren kurumların başarılı uygulamalar gerçekleştirebilmeleri için kritik öneme sahiptir. Kurumların geçiş süreçlerinde bu noktalara dikkat etmeleri bu yöntemlerden sorunsuz bir şekilde en yüksek faydayı elde etmelerine yardımcı olacaktır.

Referanslar

- [1] Richard D. Stutzke, *Estimating Software-Intensive Systems*. SEI Series in Software Engineering, Addison-Wesley, 2005.
- [2] ISO/IEC, “19761: Software engineering -- COSMIC: a functional size measurement method,” 2011.
- [3] 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.
- [4] “OVERVIEW of Function Points,” *Total Metrics*, 2013. [Online]. Available: <http://www.totalmetrics.com/function-point-resources/what-are-function-points>.
- [5] A. J. Albrecht, “Measuring Application Development Productivity,” in *Proceedings of IBM Application Development Symposium*, 1979, pp. 83–92.
- [6] “ISO/IEC 14143-1: Information Technology – Software Measurement - Functional Size Measurement - Part 1: Definition of Concepts 1998, updated in 2007.”
- [7] ISO/IEC, “20968: Software engineering -- Mk II Function Point Analysis -- Counting Practices Manual.” 2002.
- [8] 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.
- [9] ISO/IEC, “20926: Software and systems engineering -- Software measurement -- IFPUG functional size measurement method.” 2009.
- [10] A. Abran, J.-M. Desharnais, S. Oligny, D. St-Pierre, and C. Symons, “COSMIC Method Measurement Manual Version 3.0.1.,” (COSMIC), The Common Software Measurement International Consortium, 2009.
- [11] B. Özkan, “İşlevsel Büyüklük Ölçümünde Yedi Efsane,” in *5. Ulusal Yazılım Mühendisliği Sempozyumu*, 2011, pp. 59–64.
- [12] G. Low and D. Jeffery, “Function points in the estimation and evaluation of the software process,” *Softw. Eng. IEEE Trans.*, vol. 16, no. 1, pp. 64–71, 1990.
- [13] C. Kemerer, “Reliability of function points measurement: a field experiment,” *Commun. ACM*, vol. 36, no. 2, 1993.
- [14] C. F. Kemerer and B. S. Porter, “Improving the reliability of function point measurement: an empirical study,” *IEEE Trans. Softw. Eng.*, vol. 18, no. 11, pp. 1011–1024, 1992.
- [15] S. Abrahao, G. Poels, and O. Pastor, “Assessing the reproducibility and accuracy of functional size measurement methods through experimentation,” *Proceedings. 2004 Int. Symp. Empir. Softw. Eng. 2004. ISESE '04.*, pp. 189–198, 2004.
- [16] O. Turetken, O. Demirors, C. Gencel, and O. O. Top, “The Effect of Entity Generalization on Software Functional Sizing : A Case Study,” in *PROFES 2008*, 2008, pp. 105–116.