

# Scrum Uygulamalarında İşlevsel Büyüklük Tabanlı İşgücü Kestirimi ile Hikaye Puanı Tabanlı İşgücü Kestiriminin Karşılaştırılması

Erdir Ugan<sup>1</sup>, Numan Çizmeli<sup>2</sup>, Onur Demirörs<sup>2</sup>

<sup>1</sup> Bilgi Grubu Danışmanlık Ltd. Ankara, Türkiye  
erdir@bg.com.tr

<sup>2</sup> Orta Doğu Teknik Üniversitesi, Enformatik Enstitüsü, Ankara, Türkiye  
numancizmeli@gmail.com, demirors@metu.edu.tr

**Öz.** Bu çalışmada, Scrum uygulamalarında işgücü kestirimi için kullanılan iki yöntem, kestirim sonuçlarının başarısı açısından karşılaştırılmıştır. Scrum yönteminin kendi işgücü kestirim yaklaşımı olan Hikaye Puanı ile COSMIC işlevsel büyüklük ölçümü kullanılarak oluşturulan kestirim yöntemleri, bir vaka çalışması ile karşılaştırılmıştır. Çalışmada farklı regresyon yöntemleri ve yapay sinir ağı yöntemi kullanılmış, her iki girdi için de bu yöntemlerle kestirim modelleri oluşturulmuş ve kestirim çalışmalarında yaygın olarak kullanılan başarı kriterleri üzerinden değerlendirilmiştir. Değerlendirme sonucunda, COSMIC ölçümü kullanan kestirim modellerinin, Hikaye Puanına kıyasla daha başarılı kestirim sonuçları verebileceği tespit edilmiştir.

**Anahtar Kelimeler:** COSMIC, İşlevsel Büyüklük Ölçümü, SCRUM, İşgücü Kestirimi

## 1 Giriş

Yazılım projelerinde işgücü kestirimi, proje planlama etkinliklerinin en önemli basamaklarından birini oluşturur. Kestirimler, proje takvimi, bütçe planlama, maliyet değerlendirilmesi, risk analizi, kaynak planlaması ve kaynak yönetimi etkinliklerine girdi olarak kullanılmaktadır. Bu nedenle, başarılı proje yönetimi için, başarılı kestirim yapabilmek büyük önem taşımaktadır.

Birçok çalışmada, başarısız yazılım projelerinin çoğunun, hatalı kestirimler sonucu başarısız olduğu tespit edilmiştir.

Yakın geçmişte, yazılım projeleri için birçok işgücü kestirim yöntemi geliştirilmiştir. Bu yöntemlerin çoğu, birçok faaliyet alanını kapsayacak, genel geçer yöntemlerdir. Bu genel geçer yöntemlerin dışında, yazılım geliştiren kurumlar, kendi verilerine ve etkinlik alanlarına dayanan, özgün kestirim yöntemleri de geliştirmiştir.

Yazılım projelerinde işgücü kestirim yöntemleri, genel olarak üç ana grupta değerlendirilebilir.

- **Uzman Görüşü:** Bu yöntemler belli bir uygulama alanında yazılım geliştirme konusunda deneyim sahibi uzmanların, geçmiş deneyimlerine dayanarak kestirim yapımlarına dayanan yöntemlerdir.
- **Kurallı (formel) Kestirim Yöntemleri:** Bu yöntemler, nicel verilere ve bu verileri kullanan matematiksel modellere ve formüllere dayanan yöntemlerdir.
- **Bileşik Kestirim Yöntemleri:** Bu yöntemler, yukarıda bahsedilen iki türün yaklaşımlarını birleştiren yöntemlerdir. Özel değerlendirmeler ve matematiksel modellerin beraber kullanımını içerir.

Kurallı kestirim yöntemlerinde farklı istatistiksel yöntemler kullanılabilir. En yaygın kullanılan yöntemler, çoklu regresyon analizi, sınıflandırma ve karar ağacı, yapay sinir ağları, Bayes istatistiği, genetik programlama, doğrusal programlama ve bulanık mantık modelleri olarak özetlenebilir.

Bu yöntemler, gerekli işgücünü hesaplarken, geliştirilecek yazılımın büyüklüğünü ya tek ya da ana girdi olarak değerlendirir. Görece daha eski yöntemler, büyüklük olarak, kod satır sayısını (SLOC) kullanır. Ancak, yeni yöntemlerde Fonksiyon Puan Analizi (FPA), IFPUG ve COSMIC gibi işlevsel büyüklük yöntemleri tercih edilmeye başlanmıştır.

İşlevsel büyüklük ölçüm yöntemleri, kod satır sayısı tabanlı büyüklük ölçümlerine göre, proje yaşam döngüsünün ilk aşamalarında kullanılabilirler, geliştirme teknolojilerinden ve programlama dilinden bağımsız olmaları gibi üstünlüklere sahiptir.

## 1.1 COSMIC İşlevsel Büyüklük Ölçüm Yöntemi

Bu çalışmada, COSMIC işlevsel büyüklük ölçüm yöntemi kullanılmıştır. COSMIC son yıllarda gerek akademide gerekse sektörde giderek daha yaygın olarak kullanılan ve en çok kabul gören işlevsel büyüklük ölçüm yöntemlerinden biridir. Bu yöntem; bir ölçüm stratejisi, eşleştirme ve ölçüm aşamalarını içeren bir standart ölçüm modeli tanımlar.

Yazılım, Fonksiyonel Süreç (Functional Process) adı verilen işlevsellik seviyelerine ayrılır ve bu seviyede ölçülür. Bu tanım, işlevsellik için standartlaştırılmış bir soyutlama seviyesine işaret eder. Her bir Fonksiyonel Süreç, Veri Hareketleri (Data Movement) ve Veri İşlemeler (Data Manipulation) içerir. COSMIC yönteminde, Veri Hareketleri; Giriş (Entry – E), Çıkış (eXit X), Okuma (Read – R) ve Yazma (Write – W) olmak üzere dört tiptir. Bu veri hareketleri, yöntemin temel işlevsel bileşenlerini oluştururlar (Base Functional Process – BFC).

COSMIC yöntemi, iş uygulama yazılımlarının da, gerçek zamanlı yazılımların da büyüklüğünü ölçmek için kullanılabilir. Yöntem, aynı zamanda, bir ISO standardı ola-

rak kabul edilmiş, “ISO/IEC 19761: Software Engineering – COSMIC-FFP – A functional size measurement method” adıyla yayınlanmıştır. COSMIC hakkında detaylı bilgi için yöntem kılavuzundan yararlanılabilir [2]

## 1.2 Çevik yazılım geliştirme yöntemleri

Günümüzde, çevik yazılım geliştirme yöntemleri geliştiriciler arasında giderek daha çok destek bulmakta ve sektörde faaliyet gösteren kurumlar tarafından tercih edilmektedir. Çevik yöntemler, genelde, yöntem tanımlarında, direkt olarak uzman görüşüne dayalı kestirimler tanımlamakta veya bu tip kestirimlerin kullanılmasını tavsiye etmektedir.

SCRUM yöntemi, ülkemizde ve dünyada en yaygın olarak kullanılan çevik yöntemlerden biridir. Bu yöntem temelde, yinelemeli bir yaşam döngüsü tanımlamakta ve büyük iş kalemlerinin, küçük ve yönetilebilir işlere bölünmesi ve önceliklendirilmesi ile ilgili yöntemler içermektedir. SCRUM, uyarlanabilir, değiştirilebilir bir işleyiş tarzı önerir. Bunun için, yinelemeli bir yaşam döngüsü ile evrimsel geliştirme ve teslimat prensiplerini öne çıkarır. Olası değişikliklere hızlı tepki vermeyi öncelikli kabul eder.

SCRUM’da, işgücü kestiriminin, Planlama Pokeri adı verilen bir yöntemle yapılması önerilir. Bu yöntem, Wideband Delphi yöntemini temel alan, oylamaya dayalı bir kestirim yöntemidir. Her bir iş için gerekli işgücü, (genellikle) Fibonacci sayıları kullanılarak tahmin edilir. Yazılım ekibinin üyeleri, her bir iş için, poker kurallarını kullanarak kendi tahminlerini belirtir ve sonunda topluca bir karara varılır. Kestirim birimi olarak Hikaye Puanı (Story Point) kullanılır. Bu birim, bir büyüklük ölçüm biriminden çok, direkt olarak işgücü miktarını gösteren bir birimdir.

SCRUM yöntemi ve etkinlikleri hakkında detaylı bilgi için [www.scrum.org](http://www.scrum.org) adresine başvurulabilir. SCRUM’ın detayları bu bildirinin kapsamı dışındadır.

## 1.3 Çalışma

Bu çalışmada, SCRUM yönteminde tanımlana kestirim yöntemi ve kestirim birimi olan Hikaye Puanı yöntemi (Story Points) ile COSMIC işlevsel büyüklük yöntemi kullanılarak oluşturulan kestirim modelleri ile gerçekleştirilen kestirimlerin başarısı, bir durum çalışması ile karşılaştırılmıştır.

Çalışma, Türkiye’de faaliyet gösteren ve SCRUM yöntemini uygulayan, büyük ölçekli bir yazılım firmasında gerçekleştirilmiştir. Çalışma kapsamında 12 yazılım projesi incelenmiştir. Bu projelerin, COSMIC büyüklükleri ölçülmüş, alan uzmanları ile gerçek proje geliştiricilerinden Hikaye Puanı değerlendirilmeleri alınmış ve kurum ölçüm veri tabanlarından, gerçekleşen proje işgücü verileri toplanmıştır. Sonraki adımda COSMIC büyüklük verileri üzerinden farklı istatistiksel yöntemler kullanılarak farklı kestirim modelleri geliştirilmiştir. Daha sonra da bu modellerin kestirim başarıları ile alan uzmanları ve geliştiricilerin yaptığı Hikaye Puanı değerlendirmelerinin doğruluğu karşılaştırılmıştır.

## 2 Literatürdeki Çalışmalar

Literatürde yazılım işgücü kestirim modelleri ile ilgili birçok çalışma bulunmaktadır. Bu modellerin çoğu, regresyon ve yapay sinir ağı yöntemlerini temel alır. Bu bölümde en çok kabul gören modellerin bazıları özetlenmiştir.

Maxwell, Wassenhove ve Dutta, Avrupa Uzay Ajansı'na (European Space Agency – ESA) ait 108 proje üzerinde yaptıkları çalışmada [3] ESA için bir kestirim modeli geliştirmişlerdir. Daha sonra, birden çok kuruma ait verilere dayanarak oluşturulacak bir modelin, bir kurumun kestirim ihtiyaçları için kullanılıp kullanılmayacağını araştırmışlardır. Çalışmada, her bir proje için önemli olan üretkenlik faktörlerini tespit etmişler ve modelleri geliştirirken bu faktörleri kullanmışlardır.

Çalışma sonunda, en doğru kestirimleri üreten modelin yazılım büyüklüğünü temel alan model olduğunu tespit etmişlerdir. Bunun dışında, ESA verilerine göre, üretkenliği etkileyen ana faktörleri; uygulama kategorisi, programlama dili, talep edilen yazılım güvenilirlik seviyesi, bellek alanı kısıtları ve modern programlama pratiklerinin ve araçlarının kullanımı olarak belirlemişlerdir.

Moløkken ve Jorgensen, yazılım işgücü kestirim hatalarının sebepleri üzerine bir çalışma yapmıştır [4]. Çalışmada, kestirimden sorumlu çeşitli görevlerdeki çalışanlarla görüşmüş, 68 projeye ait kestirim raporlarını incelemişlerdir. Hataların sebeplerini bulmak amacıyla, 68 projenin özellikleri ile kestirim hataları arasındaki ilişkiyi istatistiksel analizler kullanarak incelemişlerdir. Çalışma sonunda, kestirimlerdeki hataların büyük oranda; kestirimi yapanların görevlerinden kaynaklanan bakış açılarından, kurumların veri toplama yöntemlerinden ve uyguladıkları analiz yöntemlerinden kaynaklandığı sonucuna varmışlardır.

Magne Jørgensen, Ulf Indahl ve Dag Sjøberg Sir Francis Galton tarafından 1800lerin sonunda gerçekleştirilen Ortalamaya Gerileme (Regression Toward the Mean – RTM) adlı çalışmanın sonuçlarından faydalanan bir çalışma gerçekleştirmiştir [5]. Çalışmada benzetim ile kestirim yaklaşımını izlemişlerdir. Benzetim için seçilen projeler çok yüksek veya çok düşük üretkenlik değerlerine sahipse, Ortalamaya Gerileme'nin bu değerleri ortalama değerlere çekmesi beklenir. Çalışmada, Ortalama Gerileme yaklaşımını değişik veri kümeleri üzerinde uygulayarak değerlendirmişler ve bu yaklaşımın kestirimlerin başarısını artırdığını tespit etmişlerdir.

Oliveira, çalışmasında kestirim modelleri için vektörel regresyon, radyal taban fonksiyonlu yapay sinir ağları ve doğrusal regresyon yöntemlerini uygulamıştır [6]. Modelleri geliştirirken NASA'ya ait yazılım proje verilerini kullanmıştır. Çalışmanın sonucunda, vektörel regresyon tabanlı modellerin, radyal taban fonksiyonlu yapay sinir ağlarına kıyasla çok daha başarılı olduğunu tespit etmiştir.

Jorgensen 2004'teki çalışmasında [7], uzman görüşüne dayalı kestirimlerde, yukarıdan aşağı ve aşağıdan yukarı yaklaşımları karşılaştırmış, hangi yaklaşımın hangi durumda uygulanması gerektiğini araştırmıştır. Yukarıdan aşağı yaklaşımda, proje için gerekli toplam işgücü, projenin tümüne dair özelliklere dayalı olarak hesaplanıp, proje etkinliklerine dağıtılır. Aşağıdan yukarı yaklaşımda ise, proje etkinlikleri ayrı ayrı değerlendirilir, her bir etkinlik için ayrı kestirim yapılıp bunlar toplanarak toplam proje işgücü hesaplanır. Jorgensen, bu çalışmasında, yukarıdan aşağı yaklaşımların başarılı

olabilmesi için, benzer projelerle ilgili deneyime ve verilere sahip olmanın gerekli olduğunu belirtmiştir. Kurumların, daha önce gerçekleştirdikleri projelere benzemeyen, yeni tip bir proje için kestirim yaparken, aşağıdan yukarı kestirim yaklaşımı kullanmaları gerektiğini belirtmiştir.

Seçkin Tunalılar, çalışmasında [8], kurumların kestirim etkinliklerini yönetmesi ve yürütmesi için bir kestirim süreci tanımlamıştır (EFES). Tunalılar, kurum genelinde ortak bir ölçüm yöntemi kullanılması, bu konuda ortak bir anlayış oluşturulmasının önemine dikkat çekmektedir. Geliştirdiği süreç, literatürde yer alan en iyi uygulamalar, sorunlar, çözümler ve önceki çalışmalara ait deneysel verilere dayanmaktadır. Çalışmada, ölçüm yöntemlerinde ve sonuçlarında farklılıklar olmasının, kurum genelinde kullanılacak bir kestirim yönteminin oluşturulmasını engelleyeceği belirtilmiştir. Çalışmada, veri toplama, büyüklük ölçümü, veri analizi, kalibrasyon ve işgücü kestirim süreçleri; süreç tanımları, kontrol listeleri, şablonlar ve diğer süreç varlıkları ile birlikte tanımlanmıştır. Tunalılar ayrıca, işlevsel benzerlik ve temel işlevsel bileşenlerin ayrı ayrı girdi olarak kullanımı gibi konuları da değerlendirmiş, kestirim modelinde işlevsel benzerliğin de değerlendirilmesinin, farklı geliştirme takımları arasında değişiklik gösterebilecek üretkenlik değerlerinin, daha tutarlı olarak hesaplanmasını sağladığını tespit etmiştir.

Bu bölümde özetlenen çalışmalar daha ziyade, gereksinimlerin tanımlandığı, proje sınırlarının belirli olduğu, alışlagelmiş yazılım geliştirme yaşam döngülerini esas almakta, bu amaçla geliştirilmiş kestirim modellerini incelemektedir. Bu bildiriye konu olan çalışmada ise, çevik/scrum yöntem izlenen projeler için COSMIC ölçüm tabanlı bir kestirim geliştirilip geliştirilemeyeceği incelenmiş ve olası yöntemler çevik yazılım geliştirmede kullanılan planlama pokeri ve hikaye puanı yöntemleriyle kestirim başarıları açısından kıyaslanmıştır.

Çalışmada, COSMIC büyüklük ölçümleri ile gerçek işgücü verileri arasındaki ilişki incelenmiştir. Daha sonra, bu büyüklük verilerini kullanan bazı matematiksel modeller oluşturulmuş, bu modeller ile elde edilen kestirimler ile hikaye puanı kestirimleri isabetlilik açısından karşılaştırılmıştır.

### 3 Vaka Çalışması

Çalışma kapsamında, Türkiye'nin en büyük devlet bankalarından biri için yazılım geliştiren büyük ölçekli bir yazılım kurumunda bir vaka çalışması gerçekleştirilmiştir. Kurumda 500 yazılım geliştirici bulunmaktadır ve kurum genelinde Scrum yöntemi uygulanmaktadır.

Kurumda, kestirimler iki seviyeli bir planlama pokeri ile gerçekleştirilmektedir. İlk aşamada, bütün geliştirme ekiplerinden birer temsilci ile oluşturulan bir üst kurulda kestirimler yapılmakta, daha sonra geliştirme ekibi içinde ikinci bir kestirim toplantısı gerçekleştirilmektedir. Bu uygulamanın, kestirimlerde kullanılan hikaye puanı büyüklüklerinin, ekipler arasında, normalize edilmesini sağladığı belirtilmiştir. Bu durum, ekiplerin performansı değerlendirilirken de tutarlı bir ölçüt kullanılmasını sağlamaktadır.

### 3.1 Vaka Çalışmasının Tasarımı

Vaka çalışmasının gerçekleştirileceği kurum seçilirken, Scrum yönetiminin tüm kurumda kullanılıyor olması, kurumda Scrum yöntemi ile en az 10 proje gerçekleştirilmiş olması ve bu projeler için gerekli ölçüm ve verilerin ulaşılabilir olması şartı aranmıştır. Vaka çalışmasında kullanılacak projelerde aşağıdaki verilerin varlığı aranmıştır:

- Projenin türü (Yeni geliştirme, bakım, iyileştirme, yeniden geliştirme)
- Yeni bir iş alanı mı?
- Yeni bir teknoloji kullanılmış mı?
- Yazılım ekibinin deneyim seviyesi.
- Ekibin üye sayısı.
- Yineleme (sprint) sayısı.
- Gereksinim tanımlama için harcanan işgücü.
- Yazılım geliştirme için harcanan işgücü.
- Test için harcanan işgücü
- Hikaye Puanı
- İş listesindeki kullanıcı hikayeleri (Backlog User Stories)
- Kullanıcı hikayelerinde referans verilen belgeler
- Kullanılan araçlar (Geliştirme ortamı – IDE, Konfigürasyon yönetim aracı, yazılım mühendisliği araçları – CASE)
- Programlama dili
- Veri tabanı yönetim sistemi
- Mimari (bağımsız uygulama, sunucu/İstemci, P2P vb.)
- Uygulama platformu (.Net, Java vb.)

Bu veriler, projelerin işlevsel büyüklüklerini hesaplamak, aykırı projeleri belirlemek ve işgücü modellerini geliştirmek için gereklidir.

Seçilen projelerin büyüklükleri COSMIC yöntemi ile ölçülürken, iş listesinde yer alan kullanıcı hikayeleri ve bu hikayelerde bahsedilen, referans verilen diğer belgelerin kullanılması kararlaştırılmıştır. Bu ölçüm sonuçları oluşturulacak kestirim modellerine temel teşkil edecektir.

Modelleri oluşturmadan önce, aykırı projeleri bulmak ve doğru sonuçları kullanmak amacıyla, ölçüm sonuçları için bir doğrulama aşaması planlanmıştır. Doğrulama için ilk olarak CUBIT adlı COSMIC ölçüm aracının otomatik doğrulama işlevi kullanılmıştır. Bu doğrulamadan sonra, ölçüm sonuçları, bir COSMIC ölçüm uzmanınca değerlendirilmiş ve doğrulanmıştır.

Doğrulamadan sonra, veriler, işgücü/büyükük oranı, kullanılan araçlar, geliştirme ekibinin deneyim seviyeleri, kullanılan platformlar, yeni teknoloji kullanılması açılarından değerlendirilmiş, aykırı projeler tespit edilmiştir. Bu aşamalardan sonra, model oluşturmak için temiz bir veri kümesi elde edileceği varsayılmıştır.

Analiz aşamasında, büyüklük verilerinden yola çıkarak işgücünü kestirmek için, farklı türlerde regresyon analizleri ile yapay sinir ağı yöntemi kullanılmıştır. COSMIC işlevsel büyüklük, hikaye puanı ve işgücü verileri üzerinde basit regresyon, çoklu reg-

resyon, güçlü regresyon, polinom regresyon, üstel regresyon, logaritmik regresyon yöntemleri uygulanmıştır. Bunların yanında COSMIC büyüklük verileri ile bir yapay sinir ağı kurulmuş ve bu model gerçek işgücü verileri ile eğitilmiştir.

Eldeki veriler için, yukarıda sıralanan yöntemler ile kestirim modelleri oluşturulduktan sonra, modellerin kestirim başarıları modellerin MMRE ve PRED değerleri hesaplanarak değerlendirilmiştir.

### 3.2 Veriler ve Analiz

12 yazılım projesi örneklem kümesi için aday olarak seçilmiştir. Tablo 1’de bu projeler ve uygulama türleri gösterilmiştir. Verilerin gizliliğini korumak amacıyla, projelere kısaltma isimler verilmiştir.

Kurum 2011’den beri Scrum yöntemi ile yazılım geliştirmektedir. Bu, çalışmanın yapıldığı tarihte üç senelik bir süreye karşılık gelmektedir. Kurumda çevik yöntemler için özel olarak tasarlanmış bir ölçüm toplama aracı kullanılmaktadır. Çalışmada kullanılan proje verilerinin çoğu bu araç üzerinden edinilmiştir.

**Tablo 1.** Örneklem İçin Aday Projeler

#	Proje	Uygulama Türü
1	BN	EMC Documentum - C# Windows Uygulama Entegrasyonu
2	MEV	C# Windows Uygulaması
3	WP	C# ASP.NET Web Sitesi
4	KK	C# Windows Uygulaması
5	HCZ	C# Windows Uygulaması
6	IH	C# Windows Uygulaması
7	IP	C# ASP.NET Web Sitesi
8	IAM	C# Windows Uygulaması
9	MT	C# Windows Uygulaması
10	NF	C# ASP.NET Web Sitesi
11	HY	SharePoint Portal
12	SGP	SharePoint Portal

#### Veri Toplama.

Tablo 1’deki tüm projeler için, aşağıdaki veriler toplanmıştır:

- Temel Proje Özellikleri: İsim, Başlangıç Tarihi, Bitiş Tarihi, Tür, Kullanılan Araçlar.
- Kullanılan Teknoloji: Programlama Dili, Veri Tabanı Yönetim Sistemi, Mimari, Kullanılan Standartlar.
- İşgücü Verisi: Gereksinim Oluşturma İşgücü, Geliştirme İşgücü, Test İşgücü.

- Hikaye Puanı.
- Hikaye Tanımları.
- Görev Tipi.
- Görevi Gerçekleştiren Ekip Üyesi

Yukarıda belirtildiği gibi, matematiksel modeller oluşturulmadan önce, ölçüm sonuçları, hataları en aza indirmek amacıyla, doğrulanmıştır. Ölçümlerde, araştırma grubumuzca geliştirilen CUBIT adlı ölçüm aracı kullanılmıştır. Doğrulamanın ilk aşaması da, yine araştırma grubumuz bünyesinde geliştirilen ve CUBIT üzerinde çalışan deneysel bir otomatik doğrulama eklentisi ile gerçekleştirilmiştir. Doğrulanmış COSMIC ölçüm sonuçları Tablo 2’de verilmiştir.

**Tablo 2.** COSMIC Ölçüm Sonuçları

#	Proje	E	X	R	W	Toplam CFP
1	HCZ	45	60	47	25	177
2	MEV	47	81	57	31	216
3	IP	34	34	18	16	102
4	IH	24	34	25	9	92
5	IAM	39	47	43	20	149
6	KK	40	44	29	26	139
7	WP	49	53	38	25	165
8	MT	47	68	52	22	189
9	NF	31	30	17	17	95
10	BN	56	47	32	34	169
11	SGP	108	95	42	78	323
12	HY	110	129	57	106	402

Büyükölçüm ölçümünden sonra, üretkenlik değerlerini hesaplamak amacıyla, COSMIC büyükölçüm verileri ile toplam işgücü verileri karşılaştırılmıştır. Proje ekiplerinin yapısı ve deneyimleri birbirine yakın olduğundan, aykırı projeleri belirlemek için bu üretkenlik verileri temel alınmıştır.

**Tablo 3.** Projelerin Üretkenlik Değerleri

#	Proje	Büyükölçüm (CFP)	Toplam İşgücü	Üretkenlik	Ortalamadan Sapma
1	HCZ	177	364	0,4863	-0,1903
2	MEV	216	519	0,4162	-0,2604
3	IP	102	117	0,8718	0,1952
4	IH	92	196	0,4694	-0,2072
5	IAM	149	332	0,4488	-0,2278
6	KK	139	286	0,4860	-0,1906



7	WP	165	188	0,8777	0,2011
8	MT	189	373	0,5067	-0,1699
9	NF	95	189	0,5026	-0,1739
10	BN	169	542	0,3118	-0,3648
11	SGP	323	218	1,4817	0,8051
12	HY	402	319	1,2602	0,5836
			Ortalama	0,6766	

Projelerin üretkenlik değerleri Tablo 3'te verilmiştir. Tabloda, SGP ve HY isimli projeler için üretkenlik değerlerinin diğer projelere göre çok daha yüksek olduğu görülebilir. Bu projeler, Microsoft SharePoint ile geliştirilmiş projelerdir. Bu çerçevede, az bir kodlama ve test işgücü ile yüksek üretkenlik sağlayan bir yapı olduğundan bu beklenen bir sonuçtur. BN isimli proje de diğerlerinden farklı bir teknoloji ile geliştirildiği için, bu üç proje aykırı olarak kabul edilmiş ve örneklemden çıkarılmıştır.

#### **Kestirim Sonuçlarının Analizi.**

Veriler toplandıktan sonra, büyüklük ve işgücü arasındaki ilişkiyi kuracak matematiksel modelleri geliştirmek üzere ilk olarak, veriler konsolide edilmiştir. Daha sonra, COSMIC büyüklük ve işgücü arasında ilişki kuran matematiksel modeller oluşturulmuştur. Analiz aşamasında ise bu modellerin kestirimlerdeki isabetliliği değerlendirilmiş, hikaye puanları ile işgücü arasındaki ilişki kuran modeller ile karşılaştırılmıştır.

Basit ve çoklu doğrusal regresyon yöntemleri uygulanırken, toplam proje işgücünün yanısıra, yazılım yaşam döngüsünün farklı aşamalarına ait işgücü verileri de değerlendirilmeye alınmıştır. Çoklu regresyon yöntemi için, COSMIC yönteminin temel işlevsel bileşenleri olan giriş (E), çıkış (X), okuma (R) ve yazma (W) veri hareketleri bağımsız değişkenler olarak kullanılmıştır. Öte yandan, Hikaye Puanları tek bir değişkenden oluştuğu ve birden çok bileşene ayıramadığı için, Hikaye Puanları için çoklu regresyon yöntemi kullanılamamıştır. Bunlara ek olarak, logaritmik regresyon, üstel regresyon, güçlü regresyon, ve polinom regresyon gibi eğri oturtma yöntemleri uygulanmıştır.

Regresyon yöntemlerine ek olarak, 9 nörondan oluşan bir yapay sinir ağı oluşturulmuştur. Bu modelin girdi parametreleri COSMIC temel işlevsel bileşenleri, bağıl değişkeni ise toplam proje işgücü olarak tanımlanmıştır.

Toplamda, COSMIC büyüklük için 13, Hikaye Puanı için ise 8 işgücü kestirim modeli oluşturulmuştur. Oluşturulan modellerin girdileri, bağıl değişkenleri ve kullanılan analiz yöntemleri Tablo 4'te verilmiştir.

**Tablo 4.** İşgücü Kestirim Modellerinin Yapısı

#	Analiz Yöntemi	Girdi Parametreleri	Bağıl Değişken
1	Basit Regresyon	Toplam Büyüklük (CFP & SP)	Toplam İşgücü
2	Basit Regresyon	Toplam Büyüklük (CFP & SP)	Analiz İşgücü

3	Basit Regresyon	Toplam Büyüklük (CFP & SP)	Kodlama İşgücü
4	Basit Regresyon	Toplam Büyüklük (CFP & SP)	Test İşgücü
5	Çoklu Regresyon	#E, #R, #W, #X (CFP)	Toplam İşgücü
6	Çoklu Regresyon	#E, #R, #W, #X (CFP)	Analiz İşgücü
7	Çoklu Regresyon	#E, #R, #W, #X (CFP)	Kodlama İşgücü
8	Çoklu Regresyon	#E, #R, #W, #X (CFP)	Test İşgücü
9	Logaritmik	Toplam Büyüklük (CFP & SP)	Toplam İşgücü
10	Üstel	Toplam Büyüklük (CFP & SP)	Toplam İşgücü
11	Güçlü	Toplam Büyüklük (CFP & SP)	Toplam İşgücü
12	Polinom	Toplam Büyüklük (CFP & SP)	Toplam İşgücü
13	YSA	#x, #e, #r, #w (CFP) (9 Nöron)	Toplam İşgücü

Modeller oluşturulduktan sonra, kestirim başarıları değerlendirilmiş ve karşılaştırılmıştır. Bunun için, Bağıl Hata (Magnitude of Relative Error – MRE), Ortalama Bağıl Hata (Mean Magnitude of Relative Error - MMRE) ve PRED (N) değerleri kullanılmıştır. MMRE ve PRED (N) bu yazılım kestirim modellerinin isabetliliğini değerlendirmek için yaygın olarak kullanılan iki kriterdir. Bu değerlerin hesaplanmasında kullanılan formüller aşağıda belirtilmiştir.

$$MRE = \frac{|Kesirilen İşgücü - Gerçekleşen İşgücü|}{Gerçekleşen İşgücü}$$

$$MMRE = \frac{100}{T} \times \sum_{i=0}^T \frac{|Kesirilen İşgücü (i) - Gerçekleşen İşgücü(i)|}{Gerçekleşen İşgücü (i)}$$

$$PRED (N) = \frac{100}{T} \times \begin{cases} 1, & MRE(i) < \frac{N}{100} \\ 0 & \end{cases}$$

Kestirimle ilgili çalışmalarda, PRED hesaplamalarında, N değeri genellikle 30 alınmaktadır. Bir modelin isabetliliği, MMRE değeri 0'a, PRED (N) değeri ise 100'e yaklaştıkça artar. Düşük MMRE değeri, modelin başarılı olduğuna işaret eder ve 0,25'ten düşük değerlere sahip modeller, uygulanabilir bir kestirim yöntemi olarak kabul edilir.

PRED ise, tahmin edilen değerlerin ne kadarının gerçek değerlere belirtilen komşuluk dahilinde (N) olduğunu belirten bir değerdir. Bu çalışmada PRED (30) değeri %70'in üzerinde olan modeller, kabul edilebilir modeller olarak değerlendirilmiştir.

Başka bir deyişle, tahmin edilen değerlerin %70'i, gerçek değerlerin %30luk komşuluğunda ise model başarılı kabul edilmiştir.

## 4 Bulgular

COSMIC büyüklük ölçümü ve Hikaye Puanı için MMRE ve PRED (30) değerleri Tablo 5'te verilmiştir.

**Tablo 5.** COSMIC ve Hikaye Puanı Temelli Modeller İçin Değerlendirme Sonuçları

#	Yöntem	Parametreler	Bağıl Değişken	MMRE (CFP)	PRED (30) (CFP)	MMRE (SP)	PRED (30) (SP)
1	Basit Regresyon	Toplam Büyüklük	Toplam İşgücü	22.48	77.78	20.15	77.78
2	Basit Regresyon	Toplam Büyüklük	Analiz İşgücü	70.95	44.44	61.76	66.67
3	Basit Regresyon	Toplam Büyüklük	Kodlama İşgücü	81.28	44.44	46.28	66.67
4	Basit Regresyon	Toplam Büyüklük	Test İşgücü	27.24	66.67	48.34	44.44
5	Çoklu Regresyon	#x, #e, #r, #w	Toplam İşgücü	6.29	100.00	N/A	N/A
6	Çoklu Regresyon	#x, #e, #r, #w	Analiz İşgücü	28.22	66.67	N/A	N/A
7	Çoklu Regresyon	#x, #e, #r, #w	Kodlama İşgücü	19.12	77.78	N/A	N/A
8	Çoklu Regresyon	#x, #e, #r, #w	Test İşgücü	9.25	100.00	N/A	N/A
9	Logaritmik	Toplam Büyüklük	Toplam İşgücü	23.14	77.78	20.68	88.89
10	Üstel	Toplam Büyüklük	Toplam İşgücü	21.52	77.78	20.94	66.67
11	Güçlü	Toplam Büyüklük	Toplam İşgücü	22.57	77.78	18.01	77.78
12	Polinom	Toplam Büyüklük	Toplam İşgücü	20.60	77.78	18.93	88.89
13	YSA	#x, #e, #r, #w (9 nöron)	Toplam İşgücü	7.88	100.00	N/A	N/A

COSMIC büyüklük kullanılarak geliştirilen modeller arasında, en iyi MMRE ve PRED (30) değerleri, bağıl değişken olarak toplam işgücünü alan, çoklu regresyon modelinde görülmektedir. COSMIC yönteminin temel işlevsel bileşenlerini girdi olarak alan bu yöntemin, 6,29 MMRE ve 100 PRED değeri ile oldukça isabetli kestirimler ürettiği ve pratikte uygulanabilir olduğu görülmektedir.

Hikaye Puanı kullanılarak geliştirilen modeller arasında ise en iyi sonuç, güçlü regresyon ve polinom regresyon yöntemlerinde elde edilmiştir. Güçlü regresyon temelli model, 18,01 MMRE ve 77,89 PRED (30) değerlerine sahipken, polinom regresyon temelli model 18,93 MMRE ve 88,89 PRED(30) değerlerine sahiptir.

COSMIC ve Hikaye Puanı kullanılarak geliştirilen modelleri, kestirim başarı değerlerine göre karşılaştırdığımızda, basit yöntemler kullanılarak oluşturulan modellerde, Hikaye Puanı temelli modellerin COSMIC temelli olanlara kıyasla biraz daha başarılı

olduğu gözlemlenmektedir. Öte yandan, çoklu regresyon ve yapay sinir ağları kullanılarak oluşturulan modellerin çok daha iyi isabetlilik değerlerine sahip olduğu görülmektedir. Ancak, bu yöntemler girdilerin birden fazla bileşenden oluşmasını gerektirdiğinden, bu yöntemler sadece COSMIC ölçümleri için kullanılabilmiştir.

## 5 Sonuçlar

Bu çalışmada, Scrum yönteminde tanımlanan işgücü kestirim yöntemi olan Hikaye Puanı ile COSMIC işlevsel büyüklük ölçümü kullanılarak oluşturulmuş bazı işgücü kestirim modelleri, kestirimlerdeki başarılarına göre karşılaştırılmıştır. Bu karşılaştırma, büyük ölçekli bir yazılım firmasında gerçekleştirilen bir vaka çalışması çerçevesinde gerçekleştirilmiştir.

Tek bir girdi parametresine bağlı modellerin kestirim isabetliliğinin genel olarak düşük olduğu gözlemlenmiştir. Bu tip yöntemlerde COSMIC ya da Hikaye Puanı kullanılması arasında, kestirim başarısı açısından kayda değer bir fark görülmemiş, her iki büyüklük için de düşük veya sınırdan başarı değerleri elde edilmiştir.

Bunun yanında, yazılımın büyüklüğünü birden çok boyutta ele alan modellerin, çok daha yüksek başarı değerlerine sahip olduğu gözlemlenmiştir. Ancak, Hikaye Puanı, birden çok boyuta ayrıştırılamayan, tekil bir ölçüm olduğundan, bu tip çoklu girişe sahip modellerde kullanılamamaktadır. COSMIC ölçümleri ise, tanımlı gereği dört ayrı temel işlevsel bileşenden oluşmakta ve çoklu girdi kullanan yöntemlere girdi olarak kullanılabilir. Bu nedenle, COSMIC büyüklüğünün, işlevsel bileşenleri ayrı ayrı kullanıldığında, yazılım büyüklüğünü Hikaye Puanından çok daha iyi temsil ettiğini söyleyebilir. Bu durum da, toplamda COSMIC tabanlı kestirim modellerini, Hikaye Puanı tabanlı modellere göre daha başarılı kılmaktadır.

Öte yandan, Hikaye Puanı kolektif uzman görüşüne dayalı bir kestirim türüdür ve COSMIC işlevsel büyüklük ölçümü ile kıyaslandığında, işgücü kestirimlerini oluşturmak daha az süre ve emek gerektirmektedir. Aynı zamanda, COSMIC ölçümü ile oluşturulacak bir kestirim modeli, kurumda geçmiş proje verilerinin bulunmasına ihtiyaç duymaktadır.

Bu noktada COSMIC tabanlı kestirimlerin Hikaye Puanı tabanlı kestirimlere göre daha isabetli olduğu ancak daha çok emek ve kaynak gerektirdiği söylenebilir. Scrum uygulayan kurumların, ihtiyaç duydukları işgücü kestirim hassasiyeti ile kestirim çalışmalarına ayırmayı planladıkları süre ve emeği kıyaslayarak bu iki yöntemden birini tercih etmeleri önerilir.

## 6 Kısıtlar

Bu bölümde çalışma sonuçlarının genelleştirilmesi ile ilgili bazı kısıtlar özetlenmiştir.

Vaka çalışmasının gerçekleştirildiği kurumda, Kullanıcı Hikayeleri için standart bir şablon kullanılmamaktadır. Bunun yanısıra, bu hikayeleri doğrulanması ve/veya geçerişmesi için bir gözden geçirme mekanizması bulunmamaktadır. Kullanıcı hikayelerinin yanlış anlaşılması ve farklı ekiplerin tanımladığı hikayeler arasındaki soyutlama seviyesi farklılıkları ölçüm sonuçlarında belli bir miktar tutarsızlığa yol açmış olabilir.

Kurumda kullanılan ve Çevik yöntemler için tasarlanmış ölçüm toplama aracı, analiz, kodlama ve test gibi aşamalar için ayrı tip etkinlikler tanımlamamaktadır. Gerçekleşen işgücü hesaplanırken, bu ayırım yazarlar tarafından geriye dönük olarak yapılmıştır. Bu ayırım, iş ve görevlerin tanımları ve atandıkları rollerden yararlanılmıştır. Ancak, çalışma kapsamında en başarılı kestirim modelleri bağıl değişken olarak toplam proje işgücünü alan modeller olduğu için, bu durum çalışma sonuçlarını etkilememiştir.

Çalışmada kullanılan veriler, tek bir kurumdan toplanmıştır. Bu durum ve proje sayısının kısıtlı olması, çalışma sonuçlarının genele uyarlanması göz önünde bulundurulması gereken bir faktördür.

## 7 Referanslar

1. ISO/IEC, IS 14143-1: 2007 - - Information Technology - Software Measurement - Functional Size Measurement - Part 1 definition of concepts, International Organization on Standardization, February 2007
2. COSMIC, Measurement Manual, Version 3.0.1, Common Software Measurement International, 2009.
3. Katrina Maxwell, Luk Van Wassenhove, Soumitra Dutta, Performance Evaluation of General and Company Specific Models in Software Development Effort Estimation. 1999.
4. Magne Jørgensen, Kjetil Moløkken-Østvold, Reasons for Software Effort Estimation Error: Impact of Respondent Role, Information Collection Approach, and Data Analysis Method, 2004
5. Magne Jørgensen, Ulf Indahl, Dag Sjøberg, Software effort estimation by analogy and ‘‘regression toward the mean’’, December 2002
6. Adriano L.I. Oliveira, Estimation of software project effort with support vector regression, December 2005
7. Magne Jørgensen, Top-down and bottom-up expert estimation of software development effort, April 2003
8. Seçkin Tunalılar, EFES : An effort estimation methodology, September 2011.
9. Comparison Of Functional Size And Story Points For Effort Prediction Effectiveness On Scrum Projects
10. Boehm, B.W., Horowitz, E., Madachy, R., Reifer, D., Bradford K.C., Steece, B., Brown, A.W., Chulani, S., Abts, C.: Software Cost Estimation with COCOMO II, Prentice Hall, New Jersey, (2000). ISBN-10: 0-13-026692-2
11. Steve McConnell. Rapid development: taming wild software schedules. Microsoft Press, 1996. ISBN: 1-55615-900-5
12. K. Kavoussanakis, Terry Sloan, UKHEC Report on Software Estimation, The University of Edinburgh December 2001
13. Jeff Sutherland; Ken Schwaber (2013). "The Scrum Guide". www. Scrum.org. Retrieved July 2013