

# Test Metrikleri Üzerinden Test Efor Tahminlemesi

Alper DÖNMEZ<sup>1</sup>, Erdem ÜÇÜNCÜ<sup>2</sup>

<sup>1</sup>Haberleşme ve Bilgi Teknolojileri Sektör Başkanlığı, ASELSAN A.Ş. Ankara

<sup>2</sup>Savunma ve Sistem Teknolojileri Sektör Başkanlığı, ASELSAN A.Ş. Ankara

{aldonmez, eucuncu}@aselsan.com.tr

**Özet.** ASELSAN bünyesindeki yazılım geliştirme bölümlerinde farklı yazılım geliştirme modelleri kullanılarak yazılımlar geliştirilmekte ve süreç içerisinde devamlı olarak test faaliyetleri yürütülmektedir. Süreç içerisinde test tasarım faaliyetlerinin sürekliliği değerlendirildiğinde, proje planlamasında test efor tahmini oldukça önem arz etmektedir. Bu makalede gerçekleştirilecek yazılım projeleri için test efor tahmini sırasında kullanılacak yöntem anlatılmaktadır. Ortaya atılan yöntem, test metrikleri kullanılarak oluşturulmuş olup, yazılım ve sistem test efor tahminlerinde bulunulmuştur. Yapılan efor tahminleri projelerdeki efor gerçekleştirmeleri ile karşılaştırılarak önerilen yöntemin başarı değerlendirilmesi de yapılmıştır.

**Anahtar Kelimeler:** Metrik, Test Eforu, Yazılım Test, Sistem Test

**Abstract.** Within the software development departments of ASELSAN, software products are developed by using different software development models and test activities are carried out progressively within the process. Considering the continuous improvement of test design activities in a specified process, test effort estimation has a significant role during the project planning. In this article, the method for software test effort estimation is described for the software projects. Software and system test effort estimation are made by the method using test metrics. The success of the proposed method is evaluated by comparing calculated test effort estimation with the realization results of the projects.

**Keywords:** Metric, Test Effort, Software Test, System Test

## 1 Giriş

Yazılım ürünlerinin karmaşıklık seviyesi arttıkça oluşabilecek riskleri yönetebilmek zorlaşmaktadır. Ölçülebilir bir sürece sahip olmak proje boyunca karşılaşılabilecek risklere karşı dayanıklı ve çevik olunmasına yardımcı olmaktadır. Test süreci de bu kapsamda değerlendirildiğinde oluşabilecek riskleri azaltmak ve süreci yönetilebilir kılmak amacıyla test sürecine yazılım test efor tahminleme yaklaşımı getirilmelidir.

Proje planlamadaki en önemli nokta projenin ne kadar sürede biteceğinin tahmin edilebilmesidir. Elde somut bir ölçüm olmadığı takdirde güvenilir bir tahminde bulunmak oldukça zordur. Tecrübeler üzerinden yapılan tahminler kesin olmadığı gibi, bazı durumlarda gerçek senaryodan da oldukça uzak olabilmektedir. Bu nedenle metrik belirleyip ölçümleme yapıp, metrikler üzerinden proje bitiş sürecine dair tahminleme yapmak başvurulabilecek en pratik yöntemlerden birisidir. Metrik bir tür ölçüm göstergesidir[1].

Tom de Marco'nun tabiriyle "Ölçülemeyen bir şeyin kontrol edilebilmesi oldukça zordur"[2]. Yazılım sektöründeki tüm çalışanlar projelerle ilgili metriklere sahip olmayı talep eder, hatta birçoğu geçerli geçersiz birçok metriğe de sahiptir. Ancak kısıtlı bir kesim sahip oldukları metriklerin ne anlama geldiğinin ve ne işe yarayabileceğinin farkındadır[3]. Yazılım projelerinde yazılım teste olan ihtiyacın artması yazılım test kaynak yetersizliğini de beraberinde getirmektedir. Kaynak yetersizliği göz önünde tutulduğunda test kaynak kullanımını verimli hale dönüştürmek için efor tahmini oldukça önemlidir. Efor tahminin en önemli girdisi metriklerdir. Bu makalede projeler üzerinde tutulan test metriklerinden, bu metriklerin proje yaşam döngüsünde yazılım test efor tahmini yapılması sırasında nasıl kullanıldığından bahsedilmektedir.

## 2 Metrikler Ne İşe Yarar?

Metrikler genel olarak süreci ölçmek, izlemek ve süreç kalitesini geliştirmek için kullanılmaktadır[1][4]. Metriklerin kullanım alanları özetlenecek/tanımlanacak olursa:

- Projenin belli bir anında, projenin sonlanması için gereken zaman ve bütçenin hesaplanması
- Gelecek projelerin zaman ve bütçe tahmini yapılması
- Güncellenmesi ya da düzeltilmesi gereken süreç ya da teknolojinin belirlenmesi
- Sürecin nasıl ilerlediğinin izlenmesi
- Projenin daha başarılı hale gelmesi için gerekli iyileştirmelerin kullanılması
- Gelecekte ne olacağının tahmin edilmesi
- Projenin kontrolden çıkmamasının sağlanması

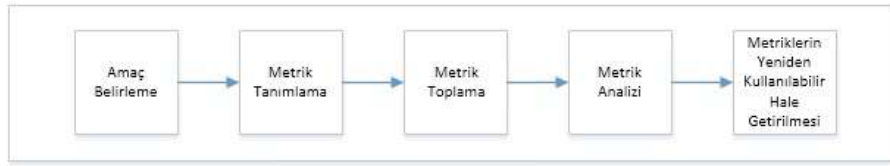
Bu maddeler projeye, şirketin yapısına ve kullanılan teknolojiye göre kurgulanabilir.

Örnek olarak; bir yazılım test raporuna aşağıdaki metriklerin eklenmesi, raporu inceleyen bir kişinin genel olarak yazılım test süreci ile ilgili fikir sahibi olmasını sağlayacaktır[5][6]:

- Gereksinim başına yazılan test senaryosu sayısı
- Toplam kaç test senaryosu koşulduğu bilgisi
- Koşulan test senaryolarının kaçının başarılı, kaçının kaldığı ve kaçının durdurulduğu bilgisi
- Koşulmayan test senaryosu sayısı
- Kaç adet hata kaydının tanımlandığı ve bu hata kayıtlarının önem derecesi bilgisi

### 3 Metrik Yaşam Döngüsü

Teknoloji sektöründe çalışan birçok uzman, metriklerin öneminin farkında olmakta ancak nereden ve nasıl başlayacağını kararını vermekte zorlanmaktadır[1]. Metriklerin de tıpkı yazılım gibi bir yaşam döngüsüne sahip olup bu döngü içerisinde geliştirilmesi gerekmektedir. Metrikler de yaşayan nesnelere olduğundan tutarlı bir yaklaşım altında toplanmaları gerekmektedir. Bu sebeple metrik yaşam döngüsünün tanımlanması ve uygulanması sağlıklı bir efor tahmini için zorunluluk değil ihtiyaçtır. Bu yaşam döngüsü şu adımlardan oluşabilir[1][4]:



Şekil. 1. Metrik Yaşam Döngüsü

#### 3.1 Amaç Belirleme

Neden metrik toplanmak istendiği bu safhada belirtilmelidir. Amacı olmadan yapılan çalışmalar hem zaman hem de bütçe olarak zarar demektir. Ayrıca verimli bir çalışma da olmayacaktır. Amaç belirlenirse rota daha kolay oluşturulup hedefe daha etkin ve daha doğru bir şekilde ulaşılır. Örnek verilecek olunursa:

- Riskli alanlar için daha fazla test süresi ayrılması
- Müşteri kabul testleri memnuniyetinin 3 yıl içerisinde %20 artırılması
- Gelecek projeler için test efor tahmininin yapılması
- Müşteriden geri dönecek hataların önem derecesinin kritik seviyede olmaması

#### 3.2 Metrik Tanımlama

Piyasada yüzlerce metrik bulunmaktadır. Bunların hepsini toplamak verimsiz olacağı gibi kısıtlı olan test süresini boş yere harcama anlamına gelecektir. Belirlenen amaca yönelik metriklerin tanımlanıp onlar üzerinde çalışma yapılmalıdır. Seçilen metrikler kolay toplanabilir, rapor edilebilir ve istatistiksel olarak modellenebilir olmalıdır. Projeler, ihtiyaçlar, kullanılan teknoloji, müşterilerinin ihtiyaçları sürekli olarak değişebilir. Bunlara göre seçili metrikler güncel olmalı ve metrik toplama işlemi canlı tutulmalıdır. Seçilen metrikler tanımlanan amaçlarla ilgili sorulara cevap olmalıdır[2].

#### 3.3 Metrik Toplama

Testlerin koşumu ve raporlanması esnasında test yönetim araçlarını kullanmak metrik toplama işlemi oldukça kolaylaştırabilir. Uygun filtreler ile belirlenen metriklerin

sonuçları toplanabilir. Toplanan metrikler ile üst düzey metrikler de basit matematiksel işlemler ile hesaplanabilir.

### **3.4 Metrik Analizi**

Projenin belirli zamanlarında, tutulan metrikler üzerinden projenin gidişatı hakkında çalışma yapılması projenin risk seviyesini azaltmaya yarar. Projede acil müdahale edilmesi gerekli alanların yöneticilerle konuşulup derhal müdahale edilmesi, çok ciddi kayıpların önüne geçebilir. Örnek verecek olursak test planı başına test senaryosu hata yoğunluğu artıyorsa müşteriye verilecek sürümde çok ciddi sayıda hata olması beklenebilir. Bu durumda hata çözümünden sonra yapılacak etki analizi için koşturulacak test senaryosu sayısının artırılması ya da test senaryolarının gözden geçirilip daha verimli hale getirilmesi düşünülebilir.

### **3.5 Metriklerin Yeniden Kullanılabilir Hale Getirilmesi**

Toplanılan metriklerin kalıcı olması ve gelecek projelerde kullanılabilir olması için anlaşılabilir bir şekilde saklanması gerekmektedir. Metriklerin toplanma amacının, test efor tahmini ihtiyacı sırasında kullanılmak olduğu düşünüldüğünde, metriklerin kalıcılığı ve doğruluk payı yüksek olmalıdır. Bu metrikleri sadece sizin kullanmayacağımızı düşünürseniz diğer paydaşlar tarafından anlaşılabilir ve kolay uygulanabilir metrik kümesi bırakmak oldukça önemlidir.

### **3.6 Metrikler Üzerinden Tahmin Yapılması**

Metrikler üzerinden tahmin işlemi metrik yaşam döngüsünde bulunan ve faydalı görülen yeniden kullanılabilir metrikler üzerinden gerçekleştirilir. Projenin içeriğine ve kullanılacak test metodolojisine göre uygun metriklerin seçilip projede yer alan uygun parametreler üzerinden tahmin işlemi gerçekleştirilebilir. Unutmayalım ki her proje sonunda metriklerimizi güncellemek gereklidir. Test ve süreçler anlamında ya da alan bilgisi olarak kazanılan her tecrübeyle metrikler hassaslaşacak ve daha iyi sonuçların elde edilmesine ortam sağlayacaktır.

## **4 Metrikler Üzerinden Test Efor Tahmini Uygulaması**

Yazılım test efor tahmininde en önemli olan parametrelerden biri zamandır. Proje zaman planlamaları her zaman gerçeklerle uyuşmayabilir. Sistemcisinden, tasarımcısına, üretimcisinden, testçisine bütün proje çalışanlarına daha fazla zaman gereklidir. Test faaliyeti projede son adımlarda olduğundan ve projenin teslim tarihi olduğundan genellikle testçiye de kısıtlı zaman kalır. Bu yüzden daha çok zaman bazlı metriklerin tutulmasında ve kullanılmasında fayda vardır. Günümüzde metrikler farklı kategoriler altında listelenmektedir. Biz bunları bölüm bölüm ayırmak yerine ölçümleri yapılan tüm metrikleri ve tutulan değerleri açıklıyor olacağız. Aselsan içerisinde çalışılan daha önceki ve hala devam etmekte olan projeler sırasında ölçüm yapılan metrikle-

ri(metrikler sadece test faaliyeti için tutulan süreyi ifade etmektedir) sıralayacak olursak eğer[2][3][4][5][7]:

**Tablo 1.** Baz Metrikler

<b>Efor Tahmini için Tutulan Metrikler</b>	<b>Metrik-AD</b>	<b>Metrik-EU</b>
Toplam kullanıcı hikayesi sayısı	100	-
Toplam gereksinim sayısı	-	1000
Kullanıcı hikayesi başına test senaryosu sayısı	7,15	-
Gereksinim başına test senaryosu sayısı	-	1
Test planı başına kullanıcı hikayesi sayısı	16	-
Test planı başına gereksinim sayısı	-	250
Tasarlanan test senaryo sayısı	715	1000
Koşulan test senaryo sayısı	700	975
Başarılı olan test senaryo sayısı	500	900
Hatalı test senaryo sayısı	200	75
Test senaryosu koşum süresi	15dk	5dk
Toplam hata sayısı	280	75
Test senaryosu tasarlanma süresi	20dk	5dk
Kullanıcı hikayesi için gözden geçirme süresi	43dk	-
Gereksinim için gözden geçirme süresi	-	4dk
Test planı başına test ortamının oluşturulması süresi	1,5gün	1gün
Hata kaydının hata kayıt aracına girilme süresi	5dk	5dk
Test planı başına test raporunun hazırlanma süresi	0,5gün	0,5gün
Hatalı test senaryosu için girilen hata sayısı	1,4	1
Test başarı oranı	%71	%92

Tablo 1’de gösterilen Metrik-AD Alper Dönmez tarafından tutulan, Metrik-EU Erdem Üçüncü tarafından tutulan metrikleri göstermektedir. Elimizdeki metrikler üzerinden basit matematiksel işlemler ile başka metrikler de elde edebiliriz. Bunları sıralayacak olursak eğer[3][5]:

- Testlerin başarı oranı =  $\frac{\text{Başarılı Test Senaryosu Sayısı}}{\text{Koşulan Test Senaryosu Sayısı}}$
- Test senaryosu için ortalama test koşum süresi =  $\frac{\text{Toplam Test Koşum Süresi}}{\text{Koşulan Test Senaryosu Sayısı}}$
- Test senaryosu için ortalama hata sayısı =  $\frac{\text{Toplam Hata Sayısı}}{\text{Koşulan Test Senaryosu Sayısı}}$
- Test tahminin doğruluk yüzdesi =  $\frac{\text{Tahmin Edilen Süre}}{\text{Gerçek Test Süresi}} \times 100$
- Hata kabul edilme oranı =  $\frac{\text{Kabul Edilen Hata Sayısı}}{\text{Toplam Hata Sayısı}}$

- Test senaryosu hata yoğunluğu =  $\frac{\text{Başarısız Test Senaryosu Sayısı}}{\text{Koşulan Test Senaryosu Sayısı}}$
- Gözden geçirme etkisi =  $\frac{\text{Gözden Geçirme Esnasında Bulunan Hata Sayısı}}{\text{Toplam Hata Sayısı}} \times 100$

Tablo 1 üzerinde ölçümleri verilen metrikler üzerinden bir sonraki yazılım test projesine ait test efor tahmini yapılabilir. Bu sayede metrikler ile proje planlamasında test eforuna ne kadar zaman ayrılması gerektiği açık bir şekilde hesaplanabilir. Tablo 1’de verilen metrikler kullanılarak Tablo 2 üzerinde efor tahmini çalışması yapılmış ve sonuçları verilmiştir. Girdi olarak sadece kullanıcı hikayesi (Alper DÖNMEZ için)-gereksinim (Erdem ÜÇÜNCÜ için) miktarları kullanılmıştır. İlk çalışmadan 140 kullanıcı hikayesinden oluşan bir küme seçilmiş olup, ikinci çalışma için 1800 adet gereksinim kümesi seçilmiştir. Tablo 1 üzerinde verilen metrikler ile Tablo 2 üzerinde 140 kullanıcı hikayesi ve 1800 gereksinim için diğer tüm satırlar doldurulmuştur. Tablo 1’de 100 kullanıcı hikayesi ve 1000 gereksinim için tahminlerde kullanılacak metrikler hesaplanmıştır. Bu metrikler üzerinden Tablo 2 doldurulmuştur. Birinci regresyon testinden sonra çok ufak bir iş hacmi olacağından test efor tahmini hesaplanmamıştır. Sadece ilk test koşumu ve daha sonrası için ilk regresyon testleri için planlama ve koşul süreleri için tahmin işlemi gerçekleştirilmiştir.

**Tablo 2.** Vaka Çalışması

Test	Test-AD	Test-EU	Regresyon-AD	Regresyon-EU
Toplam kullanıcı hikayesi	140	-	40	-
Toplam gereksinim sayısı	-	1800	-	100
Test planı sayısı	9	7	-	-
Tasarlanan test senaryo sayısı	1001	1800	-	-
Test senaryosu koşul süresi	31 gün	19 gün	9 gün	1 gün
Test senaryosu tasarlanma süresi	42 gün	19 gün	-	-
Kullanıcı hikayesi için gözden geçirme süresi	13 gün	-	-	-
Gereksinim için gözden geçirme süresi	-	15 gün	-	-
Test ortamının oluşturulması süresi	13 gün	7 gün	-	-
Hata kaydının hata kayıt aracına girilme süresi	4 gün	1 gün	-	-
Test raporu hazırlama süresi	4 gün	2 gün	1 gün	0
<b>Toplam Tahmin Süresi</b>	<b>107 gün</b>	<b>63 gün</b>	<b>10 gün</b>	<b>1 gün</b>

Tablo 2’de gösterilen Test-AD ve Regresyon-AD Alper Dönmez tarafından tahmini yapılan, Test-EU ve Regresyon-EU Erdem Üçüncü tarafından tahmini yapılan değerleri göstermektedir.

Tablo 2’de görüldüğü üzere 140 kullanıcı hikayesi yazılım testi için test yaşam döngüsünün ilk regresyon da dahil olmak üzere tamamlanması için 6 adam-ay(120 gün), 1800 gereksinime sahip sistem testi için test yaşam döngüsünün ilk regresyon da dahil olmak üzere tamamlanması için 3,5 adam-ay(70 gün) olacak şekilde süre tahmini yapılmıştır. Bu testlerin gerçekleşme süreleri sırasıyla 7 ay ve 4 ay sürmüştür. Bu durumda tahminlerin doğruluk oranını hesaplayacak olursak:

$$\text{Tahmin Doğruluk Oranı} = \frac{\text{Tahmin Süresi}}{\text{Gerçek Süre}} * 100$$

$$\text{Tahmin 1 Doğruluk Oranı} = \%85 \quad \text{Tahmin 1 Hata Oranı} = \%15 \quad (1)$$

$$\text{Tahmin 2 Doğruluk Oranı} = \%88 \quad \text{Tahmin 2 Hata Oranı} = \%12 \quad (2)$$

Hesaplanan doğruluk ve hata oranları sonuçları(1,2) firma içerisinde daha önceden karşılaşılan proje süreleri uzamaları ile karşılaştırıldığında kabul edilebilir bir ölçüde olduğundan metriklerin kullanımına devam edilmesi kararı alınabilir. Elimizde metrik olmadan baz alınarak tahmin yapılan kullanıcı hikayesi ve gereksinim sayıları göze alındığında mevcut tecrübelerle de yaklaşık süreler tahmin edilebilirdi. Bu durumda metriklerin kullanımına ihtiyaç olmadığı da düşünülebilir. Metriklerin her proje sonrası güncellenmesiyle ve alan bilgisinde kazanılacak tecrübe ile hesaplanan süre ve gerçek süre arasındaki farkın gittikçe azalacağı tahmin edilmektedir. Metriklerin doğruluk oranları tahmin için yeterli seviyede olsa da yeni projelerde tutulacak metrikler ile sentezlendikten sonra doğruluk payının artması düşünülmektedir. Bu nedenle metriklerin kontrol altında tutulması ve her yeni projede gözden geçirme işleminden sonra gerekli güncellemelerin yapılması gerekmektedir.

Metriklerin önemli bir rol oynadığı görüldüğünde ve metrikler baz alınarak süreç iyileştirilmek istendiğinde doğru soruların sorulması gerekmektedir. Ama metrik tutmaya nereden başlamak lazım? Hangi metrikleri tutalım? Ne kadar metrik yeterli? gibi çok fazla soru mevcuttur. Bu sorular tamamen projeye ve şirketin organizasyonel yapısına bağlı olarak değişiklik gösterir. O yüzden en basitinden başlamak lazım. Olabildiğince az, toplaması kolay, gelecekte işimize yarayacak metrikleri seçmek en efektif başlangıç olacaktır[8]. Mesela gelecek projeler komple manüel test iken otomasyon ile ilgili metrik tutmak ne kadar anlamlı?

## 5 Sonuç

Metriklerin tutarlılık seviyesi arttıkça efor tahmini sonucunda oluşacak hata payının düşeceği beklenmektedir. Proje planlamasının düzgün bir şekilde yürütülmesi ve geliştirilmesi için bazı ölçümler olmalıdır. Bu ölçümlerin değişken olabileceğinin unutulmaması gerekmektedir.

Bu makalede metriklerden, neden tutulması gerektiğinden ve metrikler üzerinden yapılan bir test efor tahmini çalışmasından bahsedilmiştir. Bahsedilen metrikler efor tahmininde kullanılan anahtar metrikler olup oldukça kolay toplanabilmektedir. Yapılan tahmin ile gerçek süre karşılaştırılmış olup, tahminin doğruluk oranına göre metriklerin geçerli olduğu kararı alınmıştır. Toplanan metrikler üzerinden yazılım test efor tahmininin nasıl daha verimli ve etkili hesaplandığı gösterilmiştir. Metrikleri test sürecinizin bir parçası yapmak istiyorsanız neden metrik tutmalıyız sorusuna verecek cevabınız olmalıdır.

## **Teşekkür**

Bu makaleye yorumları ve tavsiyelerinden dolayı Ediz ACAR ve Salih ÜNSAL'a teşekkür ederiz.

## **Kaynakça**

1. Narayan, H., Black, R.: Implementing a Metrics Program To Guide Quality Improvements, Testing Experience 11 (2010)
2. Hass, A, M, J.: Guide to Advanced Software Testing, Artech House, (2008)
3. Nair, J.,K.: Go Lean On Your Software Testing Metrics, Testing Experience, 11, 76-79 (2010)
4. Premal, B., Nirpal et al.: A Brief of Software Testing Metrics, International Journal on Computer Science and Engineering, Vol. 3, No 1, (2011)
5. Singh, Y., Malhotra, R.: What Should We Measure During Testing?, Testing Experience, 11, 52-54 (2010)
6. Black, R., Mitchell, J. L.: Advanced Software Testing – Vol. 3: Guide To The ISTQB® Advanced Certification as an Advanced Technical Test Analyst, Rocky Nook
7. Hutcheson, M., L.: Software Testing Fundamentals: Methods and Metrics, John Wiley&Sons (2003)
8. Lazic, L., Mastorakis N.: Cost Effective Software Metrics, World Scientific and Engineering Academy and Society, Issue 6, Volume 7, (2008)