

# Yazılım Geliştirme Süreç İyileştirme Ölçme Deneyimleri: Son mu, Başlangıç mı?

M. Ülkü Şencan

REHİS Elektronik Harp Görev Yazılımları Mdl., Aselsan A.Ş., Ankara, Türkiye

e-posta: ulku@aselsan.com.tr

**Özet.** Bu bildiri de CMMI v1.3 Seviye 3 olgunluğundaki Aselsan REHİS (Radar, Elektronik Harp ve İstihbarat Sistemleri) Sektör Başkanlığı Yönetim Sistemi yazılım geliştirme süreç iyileştirme ölçme deneyimleri sunulmaktadır. REHİS Yazılım İyileştirme Ekibi (YİE) tarafından 2011 yılı süreç güncelleme çalışmaları ile birlikte sürecin etkinlik ve verimlilik göstergelerinin de yeniden belirlenmesi için çalışma yapılmıştır. Bu çalışma sonucunda belirlenen süreç ölçümleri 6 ayda bir derlenmekte, hedef değerlerle karşılaştırılarak olumsuz yönde sapmalar için kök-neden analizleri yapılmaktadır. Bildiri de kök-neden analizlerinde çıkan bulgulardan elde edilen yazılım geliştirme süreç ölçme deneyimleri aktarılmakta, bu deneyimlerden yola çıkarak süreç göstergeleri ve etkileşimde olunan süreçlere yönelik değerlendirme ve öneriler sunulmaktadır.

**Anahtar Kelimeler:** Yazılım süreç göstergeleri, Süreç iyileştirme, Süreç ölçümleri, Yazılım geliştirme süreci, Yazılım kalitesi, CMMI, ISO 9000.

## 1 Giriş

Organizasyonların yönetim sistemleri için süreç yaklaşımını anlatan ISO 9000 rehberinde [1] süreç girdileri çıktılarına dönüştüren, birbiriyle ilişkili ve etkileşimli faaliyetlerin kümesi olarak tanımlanmaktadır. ISO standartlarında [2] süreç yaklaşımının amacı organizasyonun hedeflerini gerçekleştirirken etkinlik ve verimliliğini artırmak olarak verilmektedir. Bu nedenle de süreçlerin performansı hakkında verilerin toplanarak [3] düzeltme veya iyileştirme ihtiyaçlarının analizi önerilmektedir.

CMMI-DEV v1.3 [4] organizasyonlarda iyileştirme için üç ana boyut olarak insan, yönerge ve yöntemler, araç ve teçhizatı görmekte, süreçleri de bu boyutları bir arada tutan şey olarak tanımlamaktadır. Organizasyon süreçlerinin iyileştirilmesine odaklanan CMMI’da bu kapsamda “Ölçme ve Analiz” süreç alanı destek olarak yer almaktadır.

Süreç ölçümleri süreçlerin iyileştirilmesine yönelik göstergelerin sağlanması amacı ile projelerden uzun vadeli olarak toplanır. Süreç ölçümleri organizasyonların süreçlerini iyileştirmek için stratejik değişiklikler yapmasına olanak sağlar.

CMMI v1.3 Seviye 3 olgunluğundaki Aselsan REHİS (Radar, Elektronik Harp ve İstihbarat Sistemleri) Sektör Başkanlığı Yönetim Sistemi'nde "Süreçlerin Yönetimi Süreci (SYS)" kapsamında süreç ölçümleri alınmakta ve "Süreç Ölçme ve Değerlendirme Raporu (SÖDR)" ile iyileştirme durumları takip edilmektedir. Yönetim Sistemi'nde Şekil 1'de verilen temel ürün geliştirme süreçlerinden "Yazılım Geliştirme Süreci (YGS)" için de sürecin etkinlik ve verimliliğine yönelik göstergeler tanımlıdır.



Şekil 1. Ürün Geliştirme Süreci

Yazılım Geliştirme Süreci gösterge değerleri diğer süreç ölçümlerinde olduğu gibi 6 ayda bir derlenen ölçümlere dayanarak oluşturulmakta, gösterge sonuçları hedef değerlerle karşılaştırılarak olumsuz yönde sapmalar için kök-neden analizleri yapılmaktadır.

Bu bildiriç Yazılım Geliştirme Süreci gösterge sonuç analizleri ve kök-neden analizi bulgularından yola çıkarak süreç ölçümlerini etkileyen etkenler, gösterge modellerinin geçerliliği ve birbiri ile etkileşimi olan süreçlerin işletimine yönelik durum değerlendirmeleri ve öneriler üzerinde durulacaktır.

Bildirinin ikinci bölümünde süreç göstergeleri, analizler ve gösterge iyileştirme çalışmaları ile deneyimlerin yer aldığı Süreç İyileştirme Ölçme Çalışmaları anlatılmakta, üçüncü bölümde bu çalışmalara ait Sonuç ve Gelecek Çalışmalar verilmektedir.

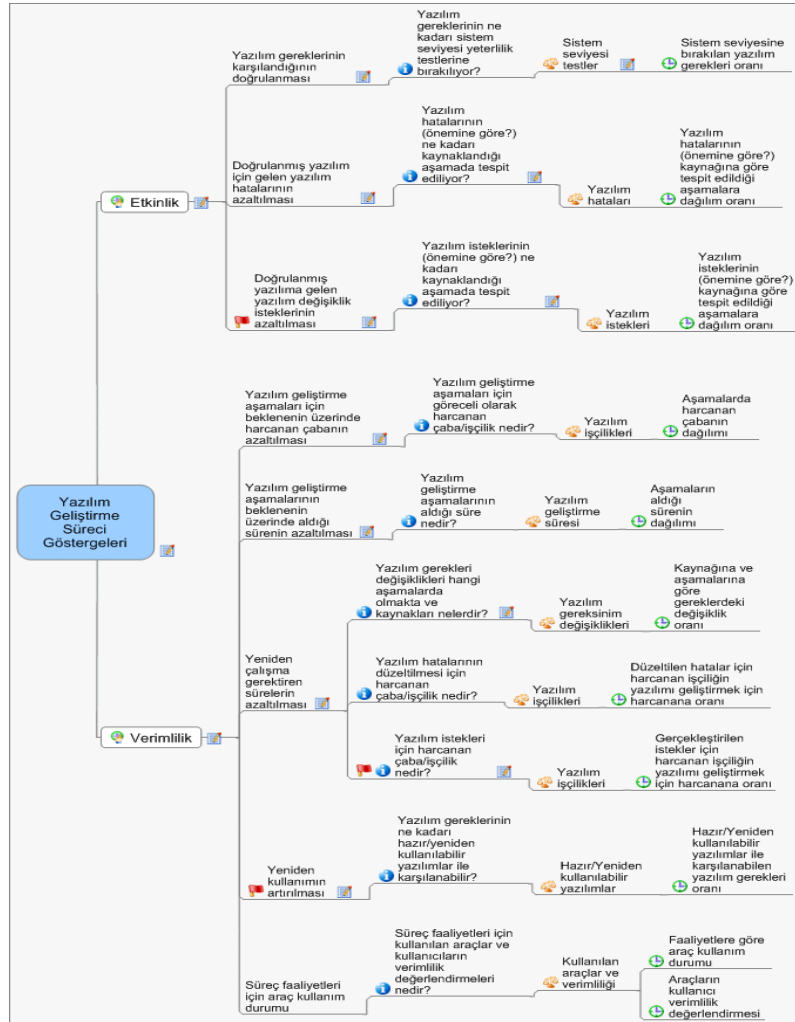
## 2 Süreç İyileştirme Ölçme Çalışmaları

REHİS Yönetim Sistemi'nin Süreçlerin Yönetimi Süreci faaliyetleri kapsamında Ölçme ve Analiz Süreci rehberliğinde tanımlanmış olan süreç iyileştirme göstergeleri için düzenli aralıklarla ölçümler alınmaktadır. Bu ölçümlerde elde edilen sonuçlar belirlenmiş hedef değerlerle karşılaştırılarak sapma durumları için kök-neden analizleri yapılmakta; kök-neden analizi sonucunda iyileştirme yapılabilecek konular belirlenmektedir.

Yazılım Geliştirme Süreci için de bu kapsamda sürecin etkinlik ve verimliliğine yönelik gösterge ve ölçütler tanımlıdır. Bu göstergelerin oluşturulması, gösterge sonuçlarının niteliklere göre analizi, hedef değerlerin karşılanmadığı durumlar için yapılan kök-neden analizleri ile bu analizlerde çıkan bulgu ve deneyimler ilerleyen alt başlıklarda anlatılmaktadır.

## 2.1 Süreç Göstergeleri

2011 yılı sonunda süreç iyileştirme amaçlı güncelleme çalışmaları ile birlikte süreç göstergeleri de gözden geçirilerek yenilenmiştir. Bu kapsamda önce sürecin etkinlik ve verimliliğini modelleyebilecek Şekil 2'deki olası gösterge ve ölçütler çıkarılmıştır.



Şekil 2. Yazılım Geliştirme Süreci etkinlik ve verimlilik göstergeleri modelleme çalışması

Daha sonra bu gösterge modelleri yazılım mühendisliği, yazılım test mühendisliği ve yazılım kalite güvence temsilcilerinin yer aldığı “Yazılım İyileştirme Ekibi (YİE)” tarafından değerlendirilmiş; süreci işletirken karşılaşılan durumları yansıtacağı öngörülen Şekil 3’te verilenler süreç göstergesi olarak belirlenmiştir.

<b>Sistem Seviyesi Test Edilecek Gerek Oranı:</b> Sistem testine bırakılan gerekler / Tüm gerekler
<b>Yazılım Hata Yoğunluğu:</b> Yazılım testleri sonrasında tespit edilen hata sayısı / Tüm gerekler
<b>Sistem Kaynaklı Gerek Değişim Oranı:</b> Sistem tasarımı ile değişen gerekler / Tüm gerekler
<b>Yazılım Kaynaklı Gerek Değişim Oranı:</b> Yazılım seviyesinde değişen gerekler / Tüm gerekler
<b>Yazılım Hatalarına Harcanan İşçilik Oranı:</b> Yazılım hata düzeltme işçiliği / Tüm yazılım işçiliği

Şekil 3. Yazılım Geliştirme Süreci etkinlik ve verimlilik süreç göstergeleri

Belirlenen bu süreç göstergeleri için hedef, türetilmiş ve temel ölçütlerin neler olduğu, ölçümlerin nasıl, hangi sıklıkla alınacağı, nasıl derleneceği ve nasıl yorumlanarak raporlanacağı tanımlandığı ölçme kartları hazırlanmıştır.

Süreçlerin Yönetimi Süreci kapsamında süreç göstergeleri Ocak ve Temmuz aylarında 6 aylık dönemlerde raporlanarak değerlendirilmekte, ölçümler belirlenmiş olan hedef değerlerle karşılaştırılarak sapma durumları için kök-neden analizleri yapılmakta ve analiz sonuçlarına yönelik iyileştirme önerileri oluşturulmaktadır.

Süreç ölçümleri projelerdeki yazılım konfigürasyon birimleri (YKB) için Şekil 4 ve Şekil 5’te örnekleri verilen MS Excel dosyalarında derlenerek gösterge hesaplamaları, hedef değerlerden sapma durumları için pivot tablo ve gösterimler otomatik oluşturulmaktadır. Sonuçlar Süreç Ölçme ve Değerlendirme Raporu (SÖDR) ile Süreçlerin Yönetimi Süreci’ne raporlanmaktadır.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1																	
2																	
3																	
4																	
5																	
6																	
7																	
8																	
9																	
10																	
11																	
12																	
13																	
14																	
15																	
16																	
17																	

Şekil 4. 2012 yılı süreç gösterge ölçüm verileri ve gösterge sonuçları sayfası

Şekil 5. 2014 yılı süreç ölçüm şablonu veri derleme sayfası

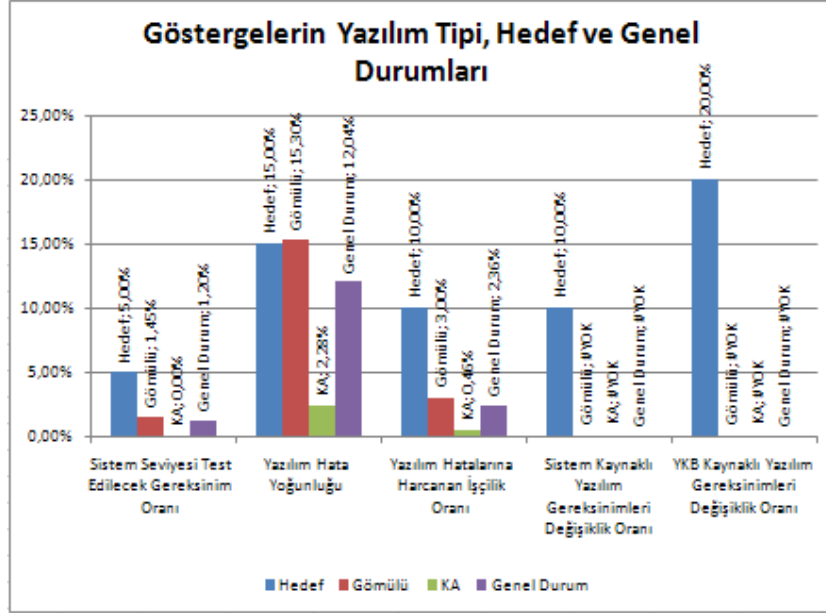
Süreç göstergeleri için ilk hedef değerler toplanan ölçümlerin ortalamasına göre YİE uzman görüşleri ile oluşturulmuştur. Hedef değerler yıllık olarak değerlendirilerek değişim eğilimine göre iyileştirmeye yönelik güncellenmektedir.

## 2.2 Gösterge Sonuç Analizleri

2012 yılı süreç göstergeleri için ölçümlerle birlikte yazılımın ait olduğu proje, yazılımın tipi ve yaşam döngüsü aşaması nitelikleri de derlenmekteydi. Bu niteliklere göre gösterge ölçümü sonuç analizleri için pivot tablo ve ilgili grafik gösterim şablonları hazırlanmıştır. Gösterge ölçüm sonuçları Şekil 6 ve Şekil 7'de örnekleri verilen pivot tablo filtreleme yetenekleri ve grafiklerden yararlanarak farklı niteliklere göre değerlendirilebilmektedir.

	A	B	C	D	E	F	G
1							
2		YKB Tipi	(Tümü)				
3		YKB'nin Bulunduğu Aşama	(Birden Çok Öğe)				
4		Hedef	5,00%	15,00%	10,00%	10,00%	20,00%
5							
6		Sistem Seviyesi	Yazılım Test Edilecek	Yazılım Hata Yoğunluğu	Yazılım Hatalarına Harcanan İşçilik Oranı	Sistem Kaynaklı Yazılım Gereksinimleri	YKB Kaynaklı Yazılım Gereksinimleri
7		Satır Değerleri	Gereksinim Oranı				
8		P1	3,07%	--	--	--	--
9		P2	1,26%	21,20%	6,42%	--	--
10		P3	0,84%	32,37%	3,93%	--	--
11		P4	0,00%	0,93%	0,83%	--	--
12		P5	0,00%	0,56%	0,30%	--	--
13		Genel Durum	1,20%	12,04%	2,36%	--	--

Şekil 6. Süreç gösterge sonuçları analizi – örnek 1



Şekil 7. Süreç gösterge sonuçları analizi – örnek 2

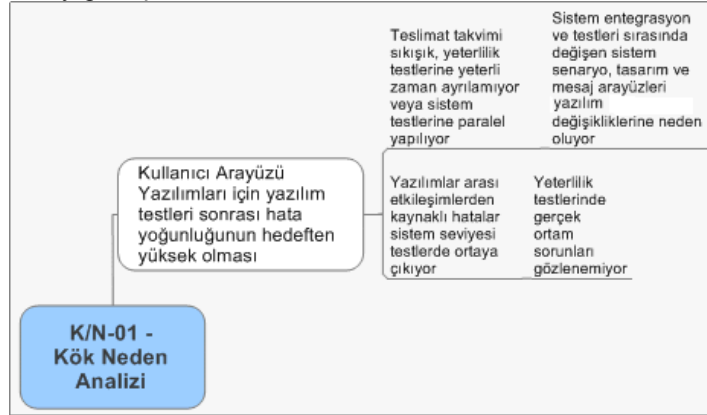
Şekil 7’de süreç gösterge sonuçlarının yazılım tiplerine göre analizi verilmektedir. 2013 yılı itibari ile derlenmiş olan ölçümlere göre oluşan gösterge sonuçlarının analizinde yazılım tiplerine göre ortalama sonuç değerleri arasında büyük farklılık olduğu gözlenmiştir. Bu farklılığın kullanılan programlama dili, geliştirme ortamı ve gerçek zamanlılık ihtiyaçları gibi özelliklerden kaynaklı olduğu öngörüsü ile göstergelerin “Gömülü” ve “Kullanıcı Arayüzü” yazılım tiplerine göre sınıflandırılabilirliği değerlendirilmiştir. Bu kapsamda gösterge hedef değerleri 2014 yılı için söz konusu yazılım tiplerine göre ayrı belirlenmiştir.

Projelere göre yapılan analizlerde ise “Radar” ve “Elektronik Harp” proje tiplerine göre gösterge sonuçlarında farklılıklar olduğu gözlenmiştir. Bu durumun ilgili proje tiplerindeki sistem alan bilgisinin olgunluğu, sistem yazılım senaryolarının karmaşıklığı, yeniden kullanım durumları, ekip deneyimleri, proje süreçlerindeki uygulamalar, yazılım geliştirme ve teste ayrılabilen işgücü ve süreler, yazılım kalite beklentileri, gerçek ortam kullanımlarında ortaya çıkan değişiklik ihtiyaçlarındaki farklılıklar gibi pek çok etkenden kaynaklanabildiği görülmektedir. Bunun sonucunda ileride göstergelerin proje tiplerine göre sınıflandırılmasının yanı sıra farklılıklardaki dağılıma göre örneğin yeniden kullanım, sistem alan bilgisinin olgunluğu, ekip deneyimi gibi diğer etkenlerin de değerlendirilmesi gerekebilecektir. Bir diğer konu ise yeniden kullanım etkeni açısından projelerde ürün hattı yaklaşımı arttıkça yazılım geliştirme süreci ve göstergelerinin de bu yaklaşıma göre güncellenebileceğidir [10].

### 2.3 Kök-Neden Analizleri

Göstergeler için belirlenen hedef değerler iki rapor döneminde karşılanıyor olduğunda iyileştirme amaçlı olarak hedef değerler güncellenmektedir. Göstergeler ölçümleri hedef değerleri karşılayamadığında ise kök-neden analizleri yapılarak iyileştirme yapılacak durumlar belirlenmektedir.

Süreç ölçümlerinin 2013 yılı rapor döneminde “Sistem Kaynaklı Gerek Değişim Oranı” göstergesi, 2014 yılı rapor dönemlerinde “Yazılım Hata Yoğunluğu” göstergesi için hedef değerlerden negatif yönde sapma gözlenmiştir. Bu gözlemler için Süreç Ölçme ve Değerlendirme Raporu’nda ilgili yazılım geliştirme ve test ekibi temsilcilerinin görüşleri değerlendirilerek aşamalı “Neden” sorusu yöntemi ile kök-neden analizleri yapılmıştır.



Şekil 8. Kök-neden analizi - örnek

Bunlardan örnek olarak “Yazılım Hata Yoğunluğu” göstergesi için “Kullanıcı Arayüzü” yazılım tipine ait ölçüm sonuçlarının Şekil 8’de verilen kök-neden analizi değerlendirilecektir. “Yazılım Hata Yoğunluğu” göstergesi yazılım geliştirme sürecinin etkinliğine yönelik tanımlanmış olup yazılımın girdi olduğu Sistem Entegrasyon, Sistem Testleri, Kabul Testleri ve sonrası süreç aşamalarında belirlenen hatalarının gereklere oranını vermektedir. Bu çerçevede “Kullanıcı Arayüzü” yazılım tipi için gösterge ölçüm sonuçları hedef değerden yüksek çıkan yazılımların geçtiği süreçleri bilen yazılım geliştirme ve test ekibi temsilcilerinin görüşleri alınmıştır.

Kök-neden analizi sonucunda ölçüm sonuçlarının hedeften sapma nedenleri olarak yazılım geliştirme sürecine girdi sağlayan ve sonrasında yazılımı girdi olarak kullanan Sistem Geliştirme Süreci ile yazılım geliştirme sürecinin tamamlanma aşaması olan Yazılım Yeterlilik Testleri ile ilgili konular öne çıkmıştır. Göstergenin tanımı dolayısıyla yazılımların gerçekleşmesi aşamalarına yönelik konular verimlilik kapsamında yer almakta, bu aşamalarda hataların süreç içi faaliyetlerde belirlenerek çözüleceği temel alınmaktadır.

Öne çıkan bu konular genel olarak değerlendirildiğinde ise yazılım yeterlilik testlerinde her şeyin gözlenemediği ancak sistem testlerinde gözlenebilecek durumlar olduğu, teslimata doğru son aşamalar olan sistem testleri sırasında ise senaryo ve

tasarım deęişikliklerinin gündeme geldięi görülmektedir. Bu durum Yazılım Geliştirme Süreci'nin Sistem Geliştirme Süreci ile yakın etkileşimi olduğunu, bağımsız değerlendirilmesi durumunda yazılım yeterlilik testlerinin sistem testleri seviyesinde maliyetli simülatör altyapılarında ilgili senaryolarla gerçekleştirilmesinin yanı sıra sistem senaryo ve tasarım deęişikliği ihtiyaçları nedeniyle sistem testleri aşamalarında yazılım deęişikliklerini de öngören daha uzun süreli zaman ayrılmasına ihtiyaç olduğu analiz ve çıkarımına götürmektedir [9]. Bunu dięer şekilde düşündüğümüzde, yani Yazılım Geliştirme Süreci Sistem Geliştirme Süreci ile birlikte bağımlı değerlendirildiğinde ise süreçlerin farklı şekilde kurgulanarak yazılım yeterlilik ve sistem testlerinin birlikte yapılması, ihtiyaç duyulan senaryo ve tasarım deęişikliklerinin erken aşamada gözlenerek maliyet ve teslimat sürelerinin azalacağı, kalitenin artacağı çıkarımına gidilmektedir. Bu çıkarımdaki süreç kurgusu günümüzde çevik yöntemler olarak bilinen uygulamaları çağrıştırırken [5] aynı zamanda Deming'in organizasyon süreçlerine sistem bakış açısını [6] da gündeme getirmektedir.

#### 2.4 Gösterge İyileştirme Çalışmaları

“Yazılım Hata Yoęunluğu” göstergesi için yapılan kök-neden analizi sırasında süreçlere yönelik kök-nedenlerin yanı sıra göstergelerin durumu ne kadar iyi modelleyebildiğine yönelik konular da gündeme gelmiştir.

Bu açıdan “Yazılım Hata Yoęunluğu” göstergesinin hata sayısının gerek sayısına oranı olması dolayısıyla her gereğin eşit olmadığı, farklı büyüklüklerde işleri ifade edebildięi, gereklerin ekiplere göre farklı ayrıntı seviyelerinde tanımlanabildięi, yeniden kullanım durumlarının da bunda etkisi olduğu görüşleri ifade edilmiştir. Dięer yandan “Yazılım Hata Yoęunluğu” göstergesi sürecin etkinliğine yönelik yazılım yeterlilik testleri ardından sürecin tamamlanması ile hata sayısının teorik olarak sıfıra yakın olması beklentisinin uygulamada bir seviye esnetilerek gerek başına düşen sayı olarak tanımlanmasıdır. Bu anlamda gösterge hataların büyüklük ve etkilerine göre hesaplanmamakta, bu nitelikleri modellememektedir.

“Yazılım Hata Yoęunluğu” göstergesi süreç ölçme çalışmaları açısından gereklerin farklı ayrıntılarda ifade edilmesi ve hataların büyüklük etkileri konularını gündeme getirmesi, kök-neden analizi ile gözlenen durumların süreç iyileştirme için belirtilerek çözüm üretilmesine katkısı ve göstergelerin sınırlarının olduğu farkındalığını oluşturması yönleriyle yararlı bir gösterge olmuştur. Bununla birlikte gösterge modelinin hedefi sağlama kaygılarıyla hesaplamaya yönelik araç yerine amaç olarak gündeme gelmeye başladığı, kök-neden analizi sonuçlarına göre etkisi uzun vadede görülebilecek düzenlemeler dışında çok farklı sonuçlar getirmeyeceęi değerlendirilmektedir. Bu nedenle işlevini tamamlayan bu gösterge modeli deneyiminden yararlanarak gereklerden bağımsız hataların etkisini göz önüne alan bir gösterge modeli önerilmektedir. Bunun yanı sıra yazılım geliştirmeyi etkileyen süreçlerin uygulanması, gözden geçirme etkinliği, yazılım faaliyetleri için yeterli işgücü, zaman ve deneyimde personel ayrılması, yeniden kullanım vb. dięer niteliklerin de ölçümlerle birlikte derlenerek ölçümlere etkisinin ve kök-neden analizlerinin geniş çerçevede değerlendirilebilmesi planlanmaktadır.



### 3 Sonuç ve Gelecek Çalışmalar

Bu bildiri de Aselsan REHİS Yönetim Sistemi'nin Süreçlerin Yönetimi Süreci faaliyetleri çerçevesinde tanımlanan ve gerçekleştirilen Yazılım Geliştirme Süreci süreç iyileştirme ölçme deneyimleri anlatılmıştır.

Yazılım Geliştirme Süreci için sürecin etkinlik ve verimliliğine yönelik gösterge ve ölçütler ile hedef değerler tanımlıdır. Süreç Ölçme ve Değerlendirme Raporu ile 6 aylık dönemlerde ölçümlerde elde edilen sonuçlar belirlenmiş hedef değerlerle karşılaştırılarak sapma durumları için kök-neden analizleri yapılmakta; kök-neden analizi sonucunda iyileştirme yapılabilecek konular belirlenmektedir. Yazılım Geliştirme Süreci göstergeleri için yapılan kök-neden analizlerinde sürecin girdilerini sağlayan Sistem Geliştirme Süreci'ne kadar giden etkilerin değerlendirilebildiği, günümüzde çevik yöntem uygulamalarında yer alan bütünsel süreç iyileştirmelerine yönelik çıkarımların yapılabildiği gözlenmiştir.

Bu nedenle yerel süreç iyileştirmesinin yanı sıra ilgili süreçlerin de yer aldığı bütünsel iyileştirmenin etkinlik ve verimlilik açısından değerlendirilebilmesi amacıyla süreç çıktısı olan ürünün kalitesine yönelik gösterge tanımlarının kullanılabilmesi önerilmektedir.

Bildiri de sürece yönelik iyileştirmelerin yanı sıra gösterge modellerinin iyileştirmesine yönelik deneyimlere de yer verilmiştir. Yazılım geliştirmeyi ve hataların oluşmasını etkileyen etkenlerin fazla ve insan odaklı olması nedeniyle etkileşimlerinin de karmaşık olması matematik modellemeyi ve ölçme çalışmalarının nokta atışı ve standart olmasını zorlaştıran durumlardır [7]. Bu nedenle tanımlanmış olan göstergeler seçilen hedef değere göre bir sorun olabileceğini işaret etseler de doğrudan sorunun ne olduğunu veya neden olduğunu ifade edemezler, modelin varsayımlarına göre yorumlanması gerekir. Bu amaçla göstergelerde ihtiyaca yönelik güncellemeler, değişiklikler yapılabileceği önerilmektedir.

Geleceğe yönelik çalışma olarak Yazılım Geliştirme Süreci etkinlik ve verimlilik göstergelerinin kök-neden analizi bulgularında çıkan konular açısından değerlendirilerek güncellenmesi, yeni gösterge modelinin geçerliliği ve model etkenlerinin belirlenmesi için istatistiksel anket yöntemlerinin [8] uygulanabileceği değerlendirilmektedir.

### Kaynaklar

1. ISO 9000 Introduction and Support Package: Guidance on the Concept and Use of the Process Approach for management systems, www.iso.org, 2008.
2. Systems and Software Engineering–Software Life Cycle Processes, ISO/IEC Standard 12207:2008.
3. Systems and Software Engineering–Measurement Process, ISO/IEC Standard 15939:2007.
4. CMMI for Development, Version 1.3, Technical Report, CMU/SEI-2010-TR-033, 2010.
5. Agile Software Teams: How They Engage with Systems Engineering on DoD Acquisition Programs, Technical Report, CMU/SEI-2014-TN-013, 2014.
6. Deming, W. E., www.wikipedia.com, erişim 2015.

7. Unterkalmsteiner, M., Gorschek, T. ve diğeri, "Evaluation and Measurement of Software Process Improvement—A Systematic Literature Review", *IEEE Trans. Software Eng.*, vol. 38, No. 2, March/April 2012.
8. Rocha, D.J., IT University of Göteborg, Strengthening The Validity Of Software Process Improvement Measurements Through Statistical Analysis: A Case Study At Ericsson AB, 2008.
9. Şahin E., Kaynak İ.K., Şencan M.Ü., "A Pilot Study: Opportunities for Improving Software Quality via Application of CMMI Measurement and Analysis", *Proc. IWSM-Mensura*, 2013, IEEE pp. 243 - 246.
10. Mohagheghi, P., Conradi R., "Quality, productivity and economic benefits of software reuse: a review of industrial studies," *Empir. Software Eng.*, Katsuro Inoue ed., Springer Science + Business Media, LLC 2007, 12:471–516, doi 10.1007/s10664-007-9040-x.