

Bir CMMI Seviye 5 Organizasyonel Performans Yönetim Projesi Örneği: Kod Kalitesini İyileştirmek

Deniz GUNGOR, Yasemin Yiğit KURU, Pinar ORGUN, Esmâ ELBİR

Huawei Turkey Research & Development Center Software Quality Department, İstanbul/Türkiye

deniz.gungor@huawei.com, yasemin.yigit.kuru@huawei.com,
pinar.orgun@huawei.com, esma.ayan@huawei.com

Özet. Yazılım süreçleri için Bütünleşik Yetenek Olgunluk Modeli olan CMMI'ye göre, bir organizasyonun kendi iş hedeflerini yakalayabilmesi için o organizasyonun performansını proaktif bir şekilde yönetmek gerekmektedir. Bu deneyim bildirisi, yazılımda geliştirmeler yapan Huawei Türkiye Arge Merkezi'nin CMMI 5. olgunluk seviyesi süreç alanlarından Organizasyonel Performans Yönetimi kapsamında gerçekleştirdiği yazılım kod kalitesini iyileştirme projesini içermektedir. İyileştirme projesi, Huawei Türkiye Arge'nin iş hedeflerine bağlı kalite süreç performans hedeflerinden biri olan Teslimat Sonrası Hata Yoğunluğu'nu azaltma hedefi doğrultusunda kod kalitesini iyileştirme üzerine hazırlanmıştır. Aynı zamanda projenin amacı, yapılan analizler, aksiyon planı ve sonuç değerlendirmesi hakkında bilgi verilmiştir. Bu deneyim bildirisinin yazılım kod kalitesini iyileştirmek isteyen ve CMMI Seviye 5 sertifikası almış veya alacak olan kurum/kuruluşlara örnek olacağı tahmin edilmektedir.

Anahtar Kelimeler: Süreç İyileştirme, CMMI Seviye 5, Organizasyonel Performans Yönetimi, Yazılım Kod Kalitesi

A CMMI Level 5 Organizational Performance Management Project Example: Improving Code Quality

Deniz GUNGOR, Yasemin Yiğit KURU, Pinar ORGUN, Esmâ ELBİR

Huawei Turkey Research & Development Center Software Quality Department, İstanbul/Türkiye

deniz.gungor@huawei.com, yasemin.yigit.kuru@huawei.com,
pinar.orgun@huawei.com, esma.ayan@huawei.com

Abstract. According to CMMI, a Capability Maturity Model Integration for software processes, to achieve the business goals of an organization, it is necessary to proactively manage the performance of an organization. This experience report contains a project to improve the code quality of the Huawei Turkey Research and Development Center which is developing software, within the Organizational Performance Management process area of the CMMI 5th level. The improvement project is based on improving the software code quality in line with the reducing Post-Delivery Defect Density (Downstream Defect Density) which is one of the Huawei Turkey Research and Development Center's quality process performance objective that is connected with the business goal. Meanwhile, information was given about the purpose of the project, analysis that has been made, action plan and the outcome evaluation. It is anticipated that this experience statement will be an example for institutions / organizations that would like to improve their software code quality and get certified or will get certify CMMI Level 5.

Keywords: Process Improvement, CMMI Level 5, Organizational Performance Management, Software Code Quality.

1 Giriş

CMMI Organizasyonel Performans Yönetimi'nde amaç periyodik olarak iş hedeflerini kalite süreç performans hedefleriyle birlikte gözden geçirmek ve gerekirse revize etmektir. İş hedeflerinin günün şartlarına göre halen güncel olduğundan ve iş stratejileri ile aynı doğrultuda olduğundan emin olmak için hedeflerin değerlendirilmesi büyük önem taşır [1].

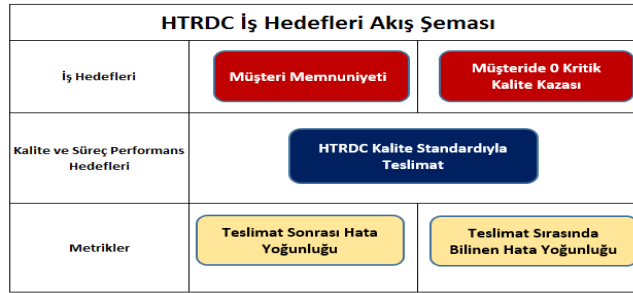
İş hedeflerine ulaşıldığının takibinin ve değerlendirmesinin yapılması için bu iş hedefleri ile bağlantılı olan başlıca süreç metriklerinin organizasyonda belirlenmiş olması gerekmektedir. Hedeflerin gözden geçirilmesi aşamasına kadar belirlenen metriklerin veri alabilme frekansına göre toplanması gerekmektedir. Örnek olarak; belirli miktarda verisi olan metrikler yıllık bazda değerlendirilebilir, değerlendirme sırasında önceden belirlenmiş limitlerin ve hedeflerin içinde olup olmadığı kontrol edilir. Kontrol sonucunda limit dışında çıkan ve iş hedeflerini en çok etkileyen metrikler verilerle analiz edilir ve bu metrikler için iyileştirme projeleri gerçekleştirilebilir. Ayrıca üst yönetim o yıl odaklanılacak iş hedeflerini stratejik olarak belirleyebilir. Yönetimin stratejik kararı ve hedefleri etkileyen metrik analiz sonuçları göz önünde bulundurularak iyileştirme projeleri belirlenir. Bu deneyim bildirisinde bilgisi verilen iyileştirme projesinin amacı, kaliteli ürün teslimatının başlıca hedefine ulaşmak ve iş hedefleriyle aynı doğrultuda olan organizasyonel düzeydeki iyileştirmeleri süreç odaklı ve sistematik bir şekilde uygulamaktır. Bu bildiri, Huawei Türkiye Arge Merkezi'nin süreç hedeflerinden olan Kaliteli Ürün Teslimatı'nı doğrudan etkileyen Teslimat Sonrası Hata Yoğunluğu metriğini konu almıştır.

2 İş Hedefleri ile Değerlendirme

Huawei Türkiye Arge Merkezi'nde her yıl ana odaklanılacak ana iş alanları ve hedefleri belirlenir. Bu alanların belirlenmesinde Çin'de konumlanmış olan idare merkezi (Headquarter) ve üst yönetim rol oynar. Bu iyileştirme projesinin gerçekleştirildiği yıl olan 2016'da belirlenen ana odaklanılacak iş hedefleri ise Müşteri Memnuniyeti ve Müşteride 0 Kritik Kalite Kazası idi. Bu hedefler aşağıda kısaca açıklanmıştır:

Müşteri Memnuniyeti: Müşteri Memnuniyeti HTRDC'de (Huawei Türkiye Arge Merkezi) yönetilen ürün veya projelerin çeşitli özellikleri konusunda müşteri beklentisini ve mevcut memnuniyeti anlamak için kullanılan bir ölçüttür.

Müşteride 0 Kritik Kalite Kazası: Bu hedefin amacı teslim edilen ürünler için müşteride 0 kalite kazası hedefini yakalamaktır. Kritik kalite kazaları, ciddi bir şekilde son kullanıcıyı etkileyen, müşteri yararına ve itibarına önemli kayıplar getiren, takibi için acil olarak kaynak tahsis edilmesini gerektiren kritik problemlerdir. Aşağıdaki şekil Müşteri Memnuniyeti ve Müşteride 0 Kritik Kalite Kazası iş hedeflerinin akışını özetler. (bkz.Şekil. 1)

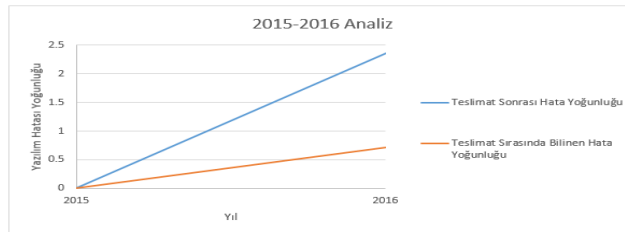


Şekil. 1. Müşteri Memnuniyeti ve Müşteride 0 Kritik Kalite Kazası İş Hedefleri Akış Şeması

Analizler: 2015 ve 2016 yıllarına ait Teslimat Sonrası Hata Yoğunluğu ve Teslimat Sırasında Bilinen Hata Yoğunluğu metrikleri hesaplanıp aşağıdaki analiz yapılmıştır. (bkz.Şekil. 2)

$$\text{Teslimat Sonrası Hata Yoğunluğu} = \frac{[(\text{Teslimat Sırasında Bilinen Hata Sayısı} + \text{Müşteride Bulunan Hata Sayısı}) * 1000]}{\text{İncelenen Yazılım Paketinin Kod Satırı}} \quad (1)$$

$$\text{Teslimat Sırasında Bilinen Hata Yoğunluğu} = \frac{(\text{Teslimat Sırasında Açık Kalan Hata Sayısı} * 1000)}{\text{İncelenen Yazılım Paketinin Kod Satırı}} \quad (2)$$



Şekil. 2. Yıllık Teslimat Sonrası Hata Yoğunluğu ve Teslimat Sırasında Bilinen Hata Yoğunluğu Trendleri

Yorum: Yıl bazındaki verilere bakıldığında Teslimat Sonrası Hata Yoğunluğunun, Teslimat Sırasında Bilinen Hata Yoğunluğundan daha fazla olduğu görülmektedir.

3 Problem Analizi

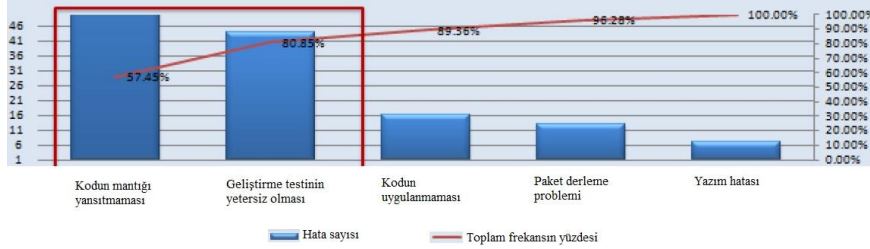
3.1 Genel Hata Analizi

2016 yılında HTRDC bazında yapılan süreç yetkinlik analizine göre, toplam tüm hatalar içerisinde teslimat sonrası hataların yüzdesi yüksektir. Hedeflere göre teslimat sonrası hataların kritik ve major hata içermeyecek şekilde tüm hatalar içindeki yüzdesinin %5'ten fazla olmaması gerekmektedir. Ancak, mevcut süreç yetkinlik analizi [2] (PCB) teslimat sonrası hataların, tüm hatalar içindeki yüzdesinin %18 olduğunu göstermektedir. Organizasyonel bazda yapılan kök-neden analizine göre; teslimat sonrası hataların (Downstream Defect) artmasının nedeni, teslimat sırasında bilinen hatalardan (Delivered Open Defect) ve kod gözden geçirme (code review) & geliştirme testi (development test) aşamalarında kaçan hataların Sistem Tasarım Doğrulama Testi (SDV) ve müşteri bulgu (post-delivery) hatası olarak sızmasıdır. Organizasyonel bazda analizi yapılan projelerin süreç yaşam döngülerinde inceleme ve test aşamaları sırasıyla; Gereksinimlerin Gözden Geçirilmesi, Tasarımın Gözden Geçirilmesi, Kod Gözden Geçirme, Geliştirme Testi ve Sistem Tasarım Doğrulama Testi şeklindedir. Kaliteli Teslimatı iyileştirmek için, kod gözden geçirme & geliştirme testi süreçlerinin verimliliğini %30 iyileştirmek gerekmektedir (Bkz.Bölüm 3.4). Böylece bu süreçlerden kaçan hataların, sistem tasarım doğrulama testi ve müşteri bulgu hatası olarak kaçması engellenebilir. (Teslimat sonrası hatalarda %30 azalma ile). Ayrıca teslimat sonrasına hata sızmasını azaltmak için teslimat sırasında bilinen hatalar analiz edilerek erken fazlarda önleyici aksiyonlar alınmalıdır.

2016 yılında yapılmış olan HTRDC süreç yetkinlik analizine göre; tüm hatalar içerisinde (Teslimat sonrası hatalar dahil) SDV fazında bulunan hataların payı en yüksek oran olan %27'dir. Bu analizde, kod gözden geçirme ve geliştirme testi fazlarında bulunan hataların oranı SDV'ye göre daha azdır.

3.2 Pareto Analizi

Aşağıdaki Pareto Analizi [3] grafiğinde (bkz.Şekil. 3) SDV hatalarının nedenleri çıkarılmış ve bu hataların frekansı analiz edilmiştir.



Şekil 3. SDV Hatalarının Sebepleri için Pareto Analizi

Yorum: SDV hataları için yapılan Pareto Analizi grafiği, çoğu SDV hatasının sebebi kodun mantığı yansıtması ve geliştirme testinin yetersiz olması olarak gösteriyor.

3.3 Problem Tanımı

Kaliteli teslimatı iyileştirmek için, Kalite Departmanı kod gözden geçirme ve geliştirme testi fazlalarının verimliliğinin %30 artmasını ve böylece SDV ve teslimat sonrası hataların azalacağı sonucuna varmıştır. Analiz edilecek problem, SDV ve teslimat sonrası hataların sebebini bulmaktır. Ayrıca istatistiksel bir programda (Minitab) yapılan İlişkilendirme Analizlerine (Correlation Analysis) [4] göre aşağıdaki ilişkiler tespit edilmiştir:

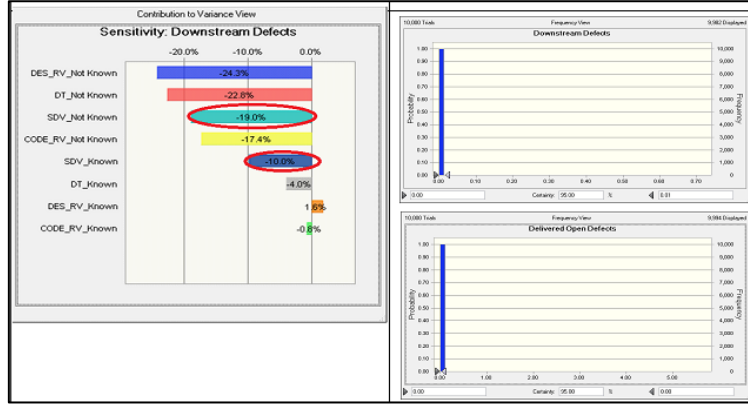
- * Teslimat sonrası hatalar ile Kod Gözden Geçirme Hataları arasında negatif yönlü kuvvetli bir ilişki,
- * Teslimat sonrası hatalar ile Geliştirme Testi Hataları arasında negatif yönlü kuvvetli bir ilişki,
- * Teslimat sonrası hatalar ile SDV öncesi hatalar arasında negatif yönlü kuvvetli bir ilişki,
- * Teslimat sonrası hatalar ile SDV hataları arasında negatif yönlü kuvvetli bir ilişki olduğu görülmüştür.

Negatif yönlü ilişki, bir değer artarken diğerinin azaldığını gösterir. Yapılan analizlerde pozitif yönlü ilişkiye rastlanılmamıştır. İlişkinin negatif/pozitif olduğunun tespitinde korelasyon katsayısının [5] işaretine bakılır; ilişki şiddetini anlamak için ise korelasyon katsayısının değerine bakılır.

3.4 Hedef Belirleme

Projelerde tanımlanan aksiyonlar uygulandığında kod gözden geçirme ve geliştirme testi süreçlerinde %30 daha fazla efor harcanacağı öngörülmüştür. Efordaki bu artış, kod gözden geçirme ve geliştirme test süreçlerinde iyileştirme yapıp, bu aktivitelere daha fazla vakit harcanacağı içindir. Revize edilmiş (kod gözden geçirme ve geliştirme testi aşamalarındaki iyileştirmelerin uygulandığı varsayılıyor) hedeflerle birlikte (bkz. Tablo 1) süreç performans modeli (PPM) [6] ile yapılan tahminlemeye göre teslimat sonrası hatalar ve teslimat sırasında bilinen hatalar 0 olarak tahminlenmektedir. (Süreç Performans Modeli (PPM), Crystal Ball programı ve eski verileri kullanarak tahmin

yapabilen istatistiksel bir programdır.) Ayrıca SDV hataları duyarlılık analizine [7] göre (sensivity analysis) artık iç hatalar içerisinde en çok paya sahip değildir ve kod gözden geçirme & geliştirme test hatalarının payı artmıştır. (bkz. Şekil. 4)



Şekil. 4. Süreç Performans Modeli ile Revize Edilen Hedeflerle Yapılan Analiz

Yorum: Organizasyon bazında yapılan duyarlılık analizine (sensivity analysis) göre, erken fazlarda hataların büyük kısmının yakalandığında, SDV’de ve teslimat sonrasında daha az hata bulunacağı görülmektedir. Süreç Performans Modelinde (PPM) revize edilmiş hedeflerle tahmin edilen (Süreç Performans Modeli ile simülasyon yapılarak istenen metriklerin tahminlenmesi) teslimat sonrası hatalar 0’dır, bu da HTRDC organizasyonel kalite hedefi ile uyumaktadır.

İyileştirme projesinde odak noktası olan süreçlere bağlı metriklerin belirlenen hedefleri, Tablo 1’de gösterilmiştir. Tablo 1’de bulunan Hedef sütunundaki metrikler, bir önceki proje hedeflerine göre %30 artmış veya azalmıştır. Kod gözden geçirme ve geliştirme testi hata yoğunluklarının artmasının sebebi, bu fazlarda daha fazla hata yakalanacağını hedeflenmesidir; SDV hata yoğunluğunun azalmasının sebebi ise, bu erken fazlarda daha fazla hata tespit edileceği ve SDV fazında daha az hata yakalanacağı içindir. Hata Yoğunluğu metriğinin birimi ise, Hata Sayısı / 1000 Kod Satırındır (Kilo Lines of Code).

$$\text{Hata Yoğunluğu} = \frac{\text{Aktivitede Tespit Edilen Hata Sayısı}}{\text{İncelenen Yazılım Paketinin Kod satırı} / 1000} \quad (3)$$

Tablo 1. Süreç Hedefleri

Alt Süreçler	Metrik	Hedef
Kod Gözden Geçirme	Kod Gözden Geçirme Hata Yoğunluğu	3.76 (Kod Gözden Geçirme Hata Yoğunluğu 30% artış)
Geliştirme Testi	Geliştirme Testi Hata Yoğunluğu	2.86 (Geliştirme Testi Hata Yoğunluğu 30% artış)
Sistem Tasarım Doğrulama (SDV)	Sistem Tasarım Doğrulama Hata Yoğunluğu (SDV)	3.24 (SDV Hata Yoğunluğu 30% azalma)

4 Süreç Analizi

4.1 Kök-Neden Analizi

Kalite Departmanı ve ilgili proje yöneticileri veri toplayıp, Teslimat sonrası hataların ve SDV hatalarının nedenini anlamak için detaylı kök-neden analizi yapmıştır. Kök-neden analizinde balık kılıcı yöntemi [8] kullanılmıştır. (Bu yöntemde önerilen kök-nedenler belli kategori başlıkları altında toplanır.) Belirlenen kök-nedenler değerlendirilerek içerisinde çözüm üretilebilecek olanlar seçilmiştir. Seçilen kök-nedenler Tablo 2’de görüldüğü gibidir:

Tablo 2. Kök-Nedenler

İnsan Kaynaklı Kök-Nedenler	Metot Kaynaklı Kök-Nedenler	Çevre Kaynaklı Kök-Nedenler	Ölçü Kaynaklı Kök-Nedenler	Araç Kaynaklı Kök-Nedenler
*Kod gözden geçirme uzmanının azlığı *Geliştiricilerin kod standartlar kılavuzları hakkında bilgi sahibi olmaması *Geliştiricilerin kod standart kuralları hakkında az bilgi sahibi olması *Kompleks hataları çözecek uzman bir takım üyesinin olmaması *Statik kod kontrolü için bir program olmaması	*Ürün bazlı kod gözden geçirme kontrol listesinin olmaması *Benzer hataların örnek alınacağı bir veritabanının olmaması	*Manuel paket derleme problemi kaynaklı eksik paket içeriği olmaması	*Teslimat sonrası oluşan hataları ölçmek için bir metriğin olmaması	*Geliştirme testinde bir otomasyon olmaması *Statik Kod kontrolü programının olmaması

4.2 Çözüm Seçimi

Organizasyonel seviyedeki balık kılıcında seçilen kök-nedenler için bir çok çözüm önerisi listelenmiştir. Bu çözümler arasında seçim yapılırken, çözümlere maliyet/risk, yarar ve yarar-maliyet oranına göre 1-10 arası puanlar verilmiştir. Puanlandırma sonucunda, en ideal puanı alan (maliyeti makul, eforu orta/az ölçekli/yararı fazla) çözümler uygulanmak üzere seçilmiştir.

4.3 Aksiyon Planlama

Önerilen çözüm önerileri değerlendirildikten sonra, seçilen çözümler için Tablo 3’de görüldüğü gibi aksiyon planı oluşturulmuştur.

Tablo 3. Aksiyon Planı

Aksiyon	Etkilenen Süreç	Sorumlu	Hedeflenen Tamamlanma Tarihi	Tamamlanma Tarihi	Statü
Çalışanların yetkinliğini arttırmak için kod standardı eğitimleri düzenlenecektir.	Yetkinlik Yönetimi	İlgili Departman Yöneticisi	23.03.2016	25.03.2016	Kapalı
Kod gözden geçirme için uzman havuzu oluşturulacak.	İnsan Yönetimi	İlgili Departman Yöneticisi	23.03.2016	25.03.2016	Kapalı
Statik kod gözden geçirme programı, kod gözden geçirme kapsamını arttırmak için sisteme entegre edilecek.	Yazılım Kodlama Prosedürü	İlgili Departman Yöneticisi	13.04.2016	13.04.2016	Kapalı
Yazılım geliştiricilerinin kendi yazdığı kodu gözden geçirmesi için ürün bazlı kod gözden geçirme kontrol listesi hazırlanacak.	Yazılım Kodlama Prosedürü	İlgili Departman Yöneticisi	13.04.2016	13.04.2016	Kapalı
Geliştirme Testi aşamasında seçilen bir Test Otomasyon Programı kullanılacak.	Geliştirme Testi	İlgili Departman Yöneticisi	21.04.2016	21.04.2016	Kapalı
HTRDC seviyesinde bilinen hatalar veri tabanı oluşturulacak.	Yazılım Kodlama Prosedürü	İlgili Departman Yöneticisi	21.04.2016	21.04.2016	Kapalı

4.4 Maliyet-Yarar Analizi

Süreç Performans Model’inde Kod Gözden Geçirme Hata Yoğunluğu, Geliştirme Testi Hata Yoğunluğu değerleri %30 artırılarak ve SDV Hata Yoğunluğu %30 azaltılarak 1000 kod satırında yapılan tahminleme sonucunda Üretim Sonrası Hata Yoğunluğu değerinin 12’den 0’a düştüğü gözlemlendi. Bu durumda, eski metrik verileri baz alınarak hesap yapıldığında, 1000 kod satırı için harcanabilecek hata çözüm eforunun azaldığı ve bir saatlik efora denk gelen ortalama Amerikan doları sabiti ile kazanılan efor baz alınarak maliyet hesabı yapıldığında önemli miktarda kazanım sağlanabileceği öngörülmüştür.

5 Sonu çar

Planlanan aksiyonlar belirlenmiş olan bir pilot, iki yaygınlaştırma projesinde denenmiştir. Bu denemeler aşamalı olarak devam etmiştir. Öncelikle pilot proje seçilip aksiyonlar bu projede uygulanmıştır. Projenin bitiminde aksiyonların verimliliği, hedeflere ulaşıp ulaşılmadığı yapılan istatistiksel analizlerle –hipotez testi- [9] belirlenmiştir. (Hipotez testi, istatistiksel bir analiz programında yapılabilen ve eski-yeni veri karşılaştırmasını yapıp, istatistiksel bir şekilde bu iki veri arasındaki farkı gösterebilen bir testtir. Bu test sonuçlarına göre, sonraki verinin önceki veriden veya hedeften ne kadar farklı olduğuna bakılarak yorum yapılır.) Bu analizler sonucunda, pilot projenin SDV fazında daha az hata bulunmuş ve aksiyonlar uygulanmadan önceki duruma göre bu fazdaki hata bulmak ve çözmek için harcanacak 9 saatlik efor kazanılmış ve önemli ölçüde maliyet karı elde edilmiştir. (Şirket gizliliği gerekçesiyle gerçek değer verilmemiştir.)

Uygulamaların pilot projede başarılı olduğu görüldüğünde, belirlenmiş aksiyonlar ve süreç güncellemeleri yaygınlaştırma projelerinde uygulanmıştır. Bu projelerin bitiminde de hipotez testi yapılarak aksiyonların verimliliği kontrol edilmiştir. Hipotez testi sonucuna göre bu projelerde de 9’ar saatlik efor kazanılmış ve önemli ölçüde maliyet karı elde edilmiştir. Ayrıca müşteride çıkabilecek hata riski (yoğunluğu) düşmüş, daha güvenli ve sorunsuz ürün gönderimi yapılmaya başlanmıştır.

Pilot ve yaygınlaştırma uygulamaları sonucunda, yapılan iyileştirme projesinin başarılı olduğu sonucuna varılmış ve bu aksiyonların standart süreç haline getirilmesi kararlaştırılmıştır. Standart süreç olan bu aksiyonların tüm organizasyon genelinde uygulanmasına karar verilmiştir.

Referanslar

1. Chaudhary,M, Chopra,A : CMMI for Development: Implementation Guide (2017)
2. Nandyal,R.S. : Key Features of a Good Process Capability Baseline Report (2015)
3. Litten,D : Project Risk and Risk Management (2010)
4. Doc.Dr.Seval Kul Sayfası, <http://www.p005.net/analiz/korelasyon-analizi>, son erişim 2017/09/12
5. Hacettepe Üniversitesi Biyoistatistik Sayfası, http://www.biyostatistik.hacettepe.edu.tr/lisans/eczacilik/Korelasyon_Regresyon.pps, son erişim 2017/09/12
6. Stoddard, R.W, Linders,B, Sapp,E.M :Exactly What are Process Performance Models in the CMMI? (2007)
7. Oracle Crystal Ball Information Page, <http://www.oracle.com/technetwork/middleware/crystalball/overview/sensitivity-chart-128074.pdf>, son erişim 2017/09/12
8. Tr Wikipedia Sayfası, https://tr.wikipedia.org/wiki/Bal%C4%B1k_k%C4%B1%C3%A7%C4%B1%C4%9F%C4%B1_diyagram%C4%B1, son erişim 2017/09/12
9. En Wikipedia Sayfası, https://en.wikipedia.org/wiki/Statistical_hypothesis_testing, son erişim 2017/09/12