

# PROJE YAŞAM DÖNGÜSÜNÜN PROJE KALİTE MALİYETİNE ETKİSİ

Onur KAYNAK<sup>1</sup>Mehmet Haluk CANBERİ<sup>2</sup><sup>1,2</sup> AR-GE Merkezi, GATE Elektronik San. ve Tic. A.Ş., Ankara<sup>1</sup>e-posta: onur.kaynak@gateelektronik.com.tr<sup>2</sup>e-posta: haluk.canberi@gateelektronik.com.tr

## 1. Giriş

Günümüz yazılım sektöründe, yazılım kalite maliyetinin toplam kalite maliyetindeki yeri ve geliştirilen projelerin başarımına etkisi her geçen gün artmaktadır. Bu kapsamda projelerin geliştirilmesinde benimsenen ve uygulanan proje yaşam döngüsü modelleri önemli rol oynamaktadır. Bu makalede, Çağlayan (Waterfall) ve Artırımsal (Incremental) proje yaşam döngüsü modelleri kullanılarak geliştirilen 2 (iki) adet proje sunulmakta olup ilgili modellerin proje kalitesine ve yazılım kalite maliyetine etkisi “Knox’un Teorik Modeli” üzerinden değerlendirilmiştir.

## 2. Yazılım Kalite Maliyeti Kategorileri

Günümüz Kalite Güvence literatüründe kalite maliyeti; kalitesizlik maliyeti ve kalitenin sağlanması için gereken maliyet olmak üzere iki kategoriye ayrılır [7]. Kalitesizlik maliyeti ürün teslimatından önce (cost of internal failure) ve ürün teslimatından sonra ortaya çıkan kalitesizlik maliyeti (cost of external failure) olmak üzere iki alt kategoriden oluşmaktadır. Kaliteyi sağlamak için gerekli maliyetler ise; değerlendirme kalite maliyetleri (appraisal cost) ve önleyici kalite maliyetlerinden (prevention cost) oluşmaktadır. Bu kalite maliyeti kategorileri bütün sektörlerde kullanılmakla birlikte yazılım sektörüne de uyarlanması yapılmıştır. Kalite maliyeti kategorileri ve ilgili tanımları Tablo-1 ve Şekil-1’de gösterilmiştir.



Şekil 1: Kalite Maliyeti Kategorileri

Tablo 1: Kalite Maliyet Kategorileri Tanımları

Kategori	Tanım	Yazılım için tipik kalite maliyetleri
Dâhili Hatalar	Ürün teslimatı öncesi belirlenen hatalar	Hata yönetimi, yeniden çalışma, yeniden yapılan testler
Harici Hatalar	Ürün teslimatı sonrası belirlenen hatalar	Teknik destek, şikâyet araştırması, hata bildirimleri
Değerlendirme	Ürünün durumunu belirlemek için yapılan faaliyetler	Test, ürün kalite etkileri
Önleme	Ürün kalitesini güvence altına almak için yapılan faaliyetler	Yazılım kalite güvence faaliyetleri, denetleme, süreç iyileştirme, ölçme ve analiz faaliyetleri

Tablo-1’de belirtilen “Dâhili Hatalar” olarak tanımlanan ürün teslimatı öncesi kalitesizlik maliyetini oluşturan kalemler; sistem ve müşteri ile birlikte yapılan kabul testleri sırasında belirlenen hataları düzeltmek için yapılan çalışmalar ve düzeltmeler sonrasında yeniden gerçekleştirilen testlere harcanan eforlardan oluşmaktadır [5]. “Harici Hatalar” olarak tanımlanan ürün teslimatı sonrası kalitesizlik maliyetini oluşturan kalemler, müşteri şikâyetlerini ve müşteriden gelen hata bildirimlerini gidermek için harcanan eforlardan oluşmaktadır. “Değerlendirme” ve “Önleme” olarak tanımlanan ve ürün kalitesini / ürünün durumunu güvence altına almaya yönelik maliyetleri oluşturan kalemler; yazılım yeterlilik ve entegrasyon testleri, ürün ve süreç kalite güvence, denetim ve gözden geçirme faaliyetlerinden oluşmaktadır. Kalitenin sağlanmasına yönelik faaliyetlere yatırım arttırdıkça, kalitesizlik kaynaklı maliyet azalmaktadır.

## 3. Knox’un Teorik Model

Knox’un Teorik Modeli, Yazılım Kalite Maliyeti kategorilerini ve bu kategorilerin aralarındaki ilişkiyi anlaşılmasını sağlayan yol gösterici bir modeldir. Yazılım kalite maliyeti kapsamındaki sınırlı veri havuzu nedeni ile Knox üretim sektöründe gelişmekte olan kalite maliyet modelini kullanmış ve Software Engineering Institute (SEI)’a ait Geliştirme için Yetenek Olgunluk Modeli Entegrasyonu (CMMI-DEV) sürüm 1.2’yi kalite maliyet modeline adapte ederek teorik bir Yazılım Kalite Maliyet Modeli oluşturmuştur. Bu model, temelde iki ana varsayıma dayanmaktadır;

Birinci varsayıma göre, CMMI Seviye-1 düzeyinde olan bir firmada toplam kalite maliyeti (değerlendirme kalite maliyetinin, önleyici kalite maliyetinin, ürün teslimatından önce ve ürün teslimatından sonra ortaya çıkan kalitesizlik maliyetlerinin toplamı), toplam geliştirme maliyetinin %60’ını oluşturmaktadır. Bu varsayım temel olarak literatürden alınan ürün teslimatı öncesi kalitesizlik maliyeti verisine ve ürün teslimatı sonrasında toplanan kalite maliyeti verisine dayanmaktadır. Ayrıca Knox’un teorik modeline göre CMMI

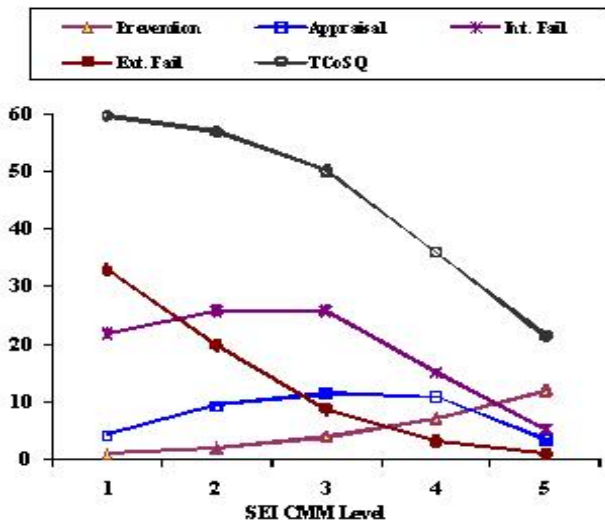
Seviye-1 düzeyinde olan firmalarda, sistem ve kabul testleri sırasında ortaya çıkan hataların giderilmesi için gerçekleştirilen düzeltici faaliyetlere harcanan efor, toplam yazılım geliştirme faaliyetlerine harcanana eforun %30-35'ini oluşturmaktadır. Ürün teslimatı sonrası ortaya çıkan hatalar kapsamında gerçekleştirilen faaliyetler ise geliştirme maliyetinin %35'ini oluşturmaktadır [2]. CMMI Seviye-1 olan firmalarda kalitenin sağlanmasına yönelik faaliyetler tanımlı olmadığı veya en az seviyede gerçekleştirildiği için geliştirme maliyetinin %0-5'ini oluşturmaktadır.

Bu nedenle, CMMI Seviye-1 olan firmalarda toplam kalite maliyeti toplam geliştirme maliyetinin %60'ını oluşturmakta olup, ilgili firmaların kalite maliyeti ürün teslimatı öncesi ve sonrası kalitesizlik maliyetlerinin toplamı üzerinden yapılan hesaplamalar ile öngörülebilmektedir. Şekil-2'de CMMI Seviye-1 organizasyonlar için toplam yazılım kalite maliyetinin, geliştirme maliyetinin %60'ını oluşturduğu gösterilmektedir.

Knox'un teorik modelinin ikinci varsayımına göre geliştirme süreci tam olgunlaşan organizasyonlarda (CMMI Seviye-5) toplam kalite maliyetinin CMMI Seviye-1 olan firmalara göre yaklaşık 2/3 oranında azalmasıdır. Bu varsayımda, Allison-Chalmers firmasının yedi yıllık kalite iyileştirme programından elde edilen veriler kaynak gösterilebilir [6]. Tablo 2 Allison-Chalmers'daki toplam kalite maliyetindeki düşüşü özetlemekte ve ikinci varsayım kapsamında öngörülen düşüşle ilişkilendirmektedir. Bu düşüş Şekil-2'de de gösterildiği gibi, toplam kalite maliyeti CMMI Seviye-5 olan bir firma için %60'dan %20'ye düşmüştür.

Tablo 2: Toplam Kalite Maliyetindeki Düşüş (TCQ)

Stepler	Kalite maliyetinin toplam geliştirme maliyetindeki yüzdesi
Başlangıçtaki TCQ	60.0
İyileştirme sonrası TCQ	18
TCQ'deki Düşüş	67.0%

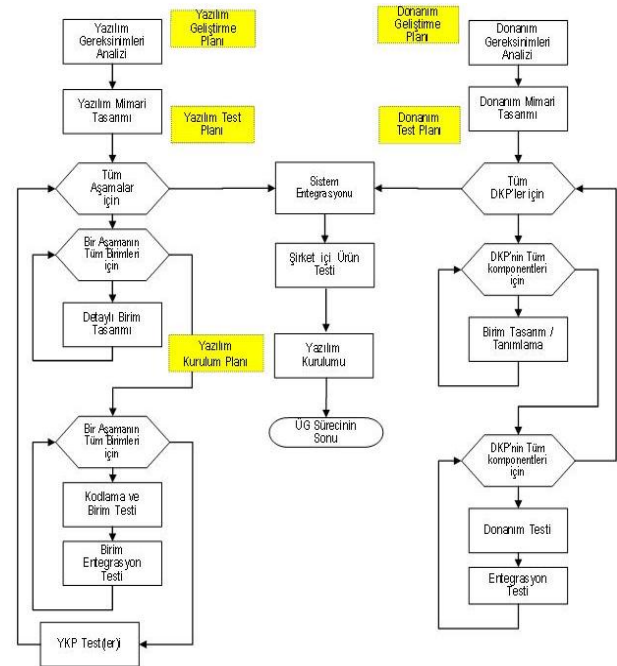


Şekil 2: Knox'un Yazılım Kalite Maliyeti için Teorik Modeli.

#### 4. Örnek Durum-1

Örnek Durum-1'de geliştirilen projede Çağlayan (Waterfall) proje yaşam döngüsü benimsenmiş ve uygulanmıştır. Bununla birlikte Çağlayan yaşam döngüsünün bir AR-GE projesi için beraberinde getirmiş olduğu dezavantajlarda projeye yansımıştır. Bu dezavantajlar özetle sıralanacak olursa;

- Projedeki gereksinimlerin değişkenliği,
- Son kullanıcının projenin analiz ve tasarım aşamasında projeye dâhil edilmemesi ve prototip çalışmalarının yapılmamasından dolayı müşteri geri dönüşlerinin sağlanamaması,
- Doğrulama gibi önleyici faaliyetler projede tanımlanmadığından, ürün ile ilgili hataların sistem ve müşteri testlerine bırakılmış olması,



Şekil 3: Çağlayan Tipi Proje Geliştirme Yaşam Döngüsü

Proje'de, analistler, tasarımcılar, geliştiriciler ve testçilerden oluşan proje ekibinin (verimliliği arttırmak için) aynı ortamda çalışmaları sağlanmıştır. Simülasyon teknolojileri kapsamında geliştirilen proje, makalenin geri kalan bölümlerinde Proje-X olarak tanımlanacaktır. Proje-X'in daha iyi anlaşılabilmesi ve yazılım kalite maliyetinin hesaplanabilmesi için proje büyüklüğü fiziksel kod satır sayısı (physical line of code [PLOC]) cinsinden ve yazılım geliştirme eforu adam-ay cinsinden tanımlanmıştır. Projenin yazılım iş paketleri 18 ayda 5 kişilik bir ekip tarafından geliştirilmiştir. Gerçekleştirilen müşteri kabulleri sonucunda yazılım iş paketleri ile ilgili harcanan efor 51,43 adam-ay olarak hesaplanmış olup, proje büyüklüğü 150 PLOC' tur. Proje kapsamında 253 adet gereksinim tanımlanmış, bu gereksinim tanımlarından 153 adet yazılım gereksinimi ve 100 adet ise donanım gereksinimi olarak sınıflandırılmıştır. Geçerleme faaliyetleri kapsamında 100 adet donanım gereksinimi için 49 adet test tanımı, 153 adet yazılım gereksinimi için ise 332 adet test tanımı oluşturulmuştur.

Proje-X için veri havuzu proje test ve kalite mühendisleri tarafından proje dokümanları da göz önüne alınarak oluşturulmuştur. Proje-X kapsamında 31 adet sistem testi

#### 4. ULUSAL YAZILIM MÜHENDİSLİĞİ SEMPOZYUMU - UYMS'09

gerçekleştirilmiştir. Üç kişilik test takımının ilgili hataları tespit etmek için harcadığı eforun maliyeti 549 adam-saat'tir. Test takımı tarafından tespit edilen hata sayısı 241 olup bu hataların 151 adeti 2 kişilik bir geliştirici ekibi tarafından 1088 adam-saat'lik eforla düzeltilmiş, geri kalan hatalar zaman kısıntısı nedeni ile müşteri ile gerçekleştirilecek kabul testlerine bırakılmak zorunda kalmıştır. Böylece, giderilemeyen 90 adet hata, kabul testlerine bırakılmıştır.

Proje-X'e ait kalite faktörü, (quality factor) hata giderme verimliliği (defect removal efficiency [DRE]) ile belirlenmiş olup, (1), hata giderme verimliliği, kalitesizlik maliyetini hesaplamak için metrik olarak kullanılmıştır.

$$DRE = \frac{FT}{FT + FC} * 100 \quad (1)$$

FT: Sistem testi kapsamında belirlenen hatalar  
FC: Müşteri tarafından belirlenen hatalar

$$DRE = \frac{241}{241 + 90} * 100$$

Proje ekibine ait hata giderme verimliliği %73'tür.

Hata başına hata tespit eforu  $549 / 241 = \sim 2.28$  adam-saat'tir.

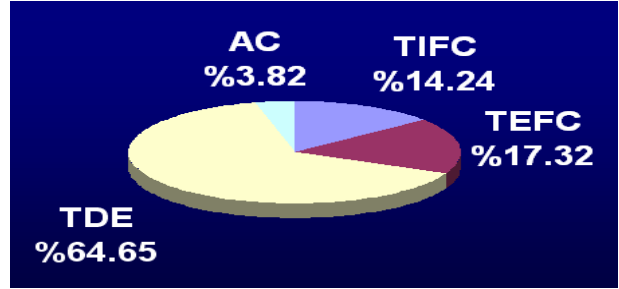
Hata başına hata giderme eforu  $1088 / 151 = \sim 7$  adam-saat'tir.

Tablo 3: Proje-X için Toplam Kalite Maliyeti Parametreleri

		Toplam Kalite Maliyeti (Adam-ay)					
		AC	TEFC	TIFC	TDE	CoQ	CoP
Proje Alt-Sistemleri	Yazılım Kısmı	2.46	7.97	7.66	36.00	18.06	54.06
	Donanım Kısmı	0.20	4.09	2.25	9.00	6.54	15.54
Toplam		2.66	12.06	9.91	45.00	24.60	69.60

Projeye ait sistem geliştirme eforu 45 adam-ay olarak hesaplanmış olup, Ürün kabulü gerçekleşmeden önce hesaplanan projedeki toplam kalite maliyeti 17.88 adam-ay'dır. Bu sonuçla Proje-X'e ait toplam efor 62.88 adam-ay olarak hesaplanmıştır. Proje-X'e ait ürün teslimatından önce ve sonra ortaya çıkan toplam kalite maliyetleri (total internal failure cost [TIFC] and total external failure cost [TEFC]),

değerlendirme maliyeti (appraisal cost [AC]) ve toplam geliştirme eforuna (TDE) ait yüzdeler Şekil-4'te sunulmuştur.



Şekil 4: Proje-X için Toplam Kalite Maliyeti Parametrelerine ait Yüzdeler

Ürün kabulü gerçekleşikten sonra garanti sürecinde projenin yazılım kısmı ile ilgili olarak 29 adet hata belirlenmiş olup, bu hataları gidermek için 2.63 adam aylık bir efor sarf edilmiştir. Ayrıca projenin donanım kısmında belirlenen 45 adet hata için de 4.09 adam aylık efor sarf edilmiştir. İlgili metrikler ile birlikte toplam kalite maliyeti, 24.60 adam ay olarak hesaplanırken; toplam proje maliyetinin 69.60 adam ay'a yükseldiği gözlenmiştir (Tablo 3). Böylece toplam kalite maliyeti toplam proje maliyetinin %35'ine ulaşırken, harici hatalardan kaynaklanan maliyet toplam proje maliyetinin %17'sine ulaşmıştır.

#### 5. Örnek Durum-2

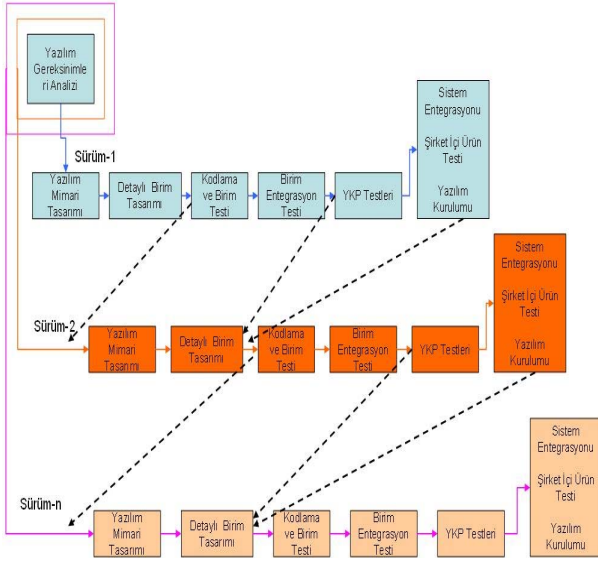
Örnek Durum-2'de geliştirilen projede Artırımsal (Incremental) proje yaşam döngüsü benimsenmiş ve uygulanmıştır. İlgili yaşam döngüsü modelinin benimsenmesinde Proje X'den edinilen dersler önemli rol oynamıştır.

Tablo 4: Proje-Y için Büyüklük Parametreleri ve Harcanan Efor

Rakamlarla Proje Y			
Gereksinim Sayısı	Proje Büyüklüğü (COSMIC)	Kod Satır Sayısı	Tahmini Efor (COCOMO 81)
2269 Adet	21.896 CFSU	657 PLOC	1584 Adam-ay

Proje Y'deki sistemin tek bir seferde geliştirmek için çok büyük olması, dolayısıyla sistemin parça parça geliştirilmesini zorunlu hale getirmiştir. Bu kapsamda belirlenen gereksinimlerin planlanmış bir dizi sürüm ile gerçekleştirildiği artırımsal yaşam döngüsü modeli kullanılmıştır. Projenin geliştirilmesinde; ilk sürümde gereksinimlerin %31 gerçekleştirirken, ikinci sürümde %30'u ve son üründe de

%39 kısmını gerçekleştirilmesi planlanmıştır. Böylece, Çağlayan yaşam döngüsü kullanımının sebep olacağı 182 iş günlük gecikmenin önüne geçilmiştir. Ayrıca, sisteme ait istenen bazı yeteneklerin müşteriye erken teslimi mümkün olmuş, son ürün öncesi teslim edilecek iki prototip ile müşteriden geri dönüşler alınması olanaklı hale gelmiştir. Her sürümde yapılan geçerleme faaliyetleri ve ara ürünlere yönelik doğrulama faaliyetleri ile olası hataların müşteri testlerine yansımalarının önüne geçilmiştir.



Şekil 5: Artırimsal Proje Geliştirme Yaşam Döngüsü

Proje Yenin geliştirilmesinde benimsenen artırimsal yaşam döngüsü doğrultusunda sistem seviyesinde gereksinim analizi ve sistem tasarımı faaliyetleri tüm sürümler için bir kere işletilirken, alt-sistem seviyesinde gereksinim analizi ve tasarım faaliyetleri ilk sürüm için ayrı işletilmiş ve bundan sonra gerçekleştirilecek sürümler için de ayrı işletilmesi planlanmaktadır.

Proje X'den alınan dersler doğrultusunda CMMI Seviye-3 tabanlı süreç iyileştirme projesi başlatılmış ve Proje-Y süreç iyileştirme faaliyetlerinin uygulanmasında pilot proje olarak seçilmiştir. CMMI Destek Süreç Kategorisi kapsamında yeniden ele alınarak iyileştirilen doğrulama ve geçerleme süreç alanları doğrultusunda kalitesizlik maliyetinin en aza indirmek amacıyla çalışmalar başlatılmıştır. Proje Yaşam döngüsü doğrultusunda tanımlanan her sürümde ve her sürümü oluşturan ara fazlarda ürün ve süreç kalite güvence faaliyetleri gerçekleştirilmiş, ürün kalitesinin sağlanmasına yönelik olarak ilk sürüme kadar 133 adam-ay'lık bir efor sarf edilmiştir. Buna göre, ilk sürüme kadar sarf edilen toplam eforun 750 adam-ay olduğu düşünüldüğünde, kalitenin sağlanmasına yönelik harcanan efor toplam proje maliyetinin %17'sine ulaşmaktadır. Ayrıca, Proje-Y'nin ilk sürümündeki kalitenin sağlanmasına yönelik maliyette Proje-X'dekine göre %13'lük bir artış gözlemlenmiştir. İlk sürümün müşteri ile birlikte yapılan yazılım yeterlilik testlerinden 0 (sıfır) majör hatayla çıkmıştır.

## 6. Sonuçlar

Örnek Durum-1'den çıkarılan dersler doğrultusunda proje ve süreç yönetimindeki zayıf noktalar belirlenerek CMMI-Dev v1.2 Seviye-3 tabanlı süreç iyileştirme projesi şirket bünyesinde başlatılmıştır. Örnek Durum-2'de tanımlanan proje, süreç iyileştirme çalışmalarının uygulanacağı pilot proje olarak seçilmiş olup, proje sonunda Knox'un Teorik Modeli doğrultusunda yazılım kalite maliyetindeki düşüşe ulaşılması hedeflenmektedir. Bu düşüşün sağlanabilmesi için kalitesizlik maliyetini en aza indirmek, kaliteyi sağlamak için gereken maliyeti maksimuma çıkarmak esas amaç olarak benimsenmiştir.

Kalitesizlik maliyetini düşürmeye yönelik gerçekleştirilen ürün ve süreç kalite güvence faaliyetleri, proje yönetiminde benimsenen yaşam döngüsü modeli ile paralel yürütülmediği sürece kalitesizlik maliyetinin azalması mümkün olamamaktadır. Bu kapsamda proje yaşam döngülerinin uygulanmasında ürün teslimatından önce ve ürün teslimatından sonra ortaya çıkan kalitesizlik maliyetini en aza indirecek olan ürün ve süreç kalite güvence tetkiklerine, ölçme ve analiz çalışmaları, denetleme ve gözden geçirme öleyici kalite faaliyetlerine proje yönetimi ile koordineli olarak ağırlık verilmesi gerekmektedir. Her ne kadar bütün proje yaşam döngülerine kalite güvence faaliyetleri uygulanabiliyor olsa da, geliştirilecek projenin yapısına uygun olmayan yaşam döngüsü seçimleri kalite güvence faaliyetlerinin etkinliğini doğrudan etkilemektedir.

Hedeflenen başarı seviyesine ulaşamayan projeler, pek çok sorunun yanında CMMI gibi süreç iyileştirme modellerinin uygulanmasına da zemin hazırlamaktadır. Süreç iyileştirme projeleri aracılığı ile proje yönetimi kaotik ortamdan istikrarlı ortama geçtikçe, kalite güvence faaliyetlerinin yalnız kalite güvence departmanının sorumluluğunda olduğunu düşünen anlayış, kalite güvencenin her departmanın sorumluluğunda olduğu bilinciyle yer değiştirmektedir.

## 7. Kaynakça

- [1] CMMI Product Team, CMMI® for Development, Version 1.2 CMMI-DEV, V1.2 CMU/SEI-2006-TR-008, August 2006
- [2] Knox, S.T., "Modeling the Cost of Software Quality", Digital Technical Journal, 1993, Vol.5, No:4, pp 9-16.
- [3] Krasner, H., "Using the Cost of Quality Approach for Software", [www.stsc.hill.af.mil/crosstalk/1998/nov/krasner.asp](http://www.stsc.hill.af.mil/crosstalk/1998/nov/krasner.asp).
- [4] Mandewille, W.A., "Software Cost of Quality", IEEE Journal on Selected Areas in Communications, February, 1990, Vol.8, No:2, February, pp. 315-318.
- [5] Houston, D., Keats J. Bert., "Cost of Software Quality: A Means of Promoting Software Process Improvement"
- [6] Yourdon, E., "The Decline and Fall of the American Programmer (Englewood Cliffs", NJ: Yourdon Press, Prentice-Hall, 1992.
- [7] American Society for Quality Control, ASQC (1986), "Principles of Quality Costs", Milwaukee.
- [8] Sommerville, I., "Software Engineering", Addison Wesley, 2007, pp.65-73