

Yazılım Kalite Maliyetlerinin Kontrol Altında Tutulması Yoluyla Proje Geliştirme Maliyetlerinin Düşürülmesi: Literatüre bir bakış

Burcu Yalçiner, Nebi Yılmaz, Kıvanç Dinçer

Yazılım Mühendisliği Araştırma Grubu, Bilgisayar Mühendisliği Bölümü
Hacettepe Üniversitesi, Ankara, Türkiye

{burcuyalciner, yilmaz}@cs.hacettepe.edu.tr
kivanc.dincer@hacettepe.edu.tr

Özet: Yazılım firmalarının çalıştıkları rekabetçi sektörde varlıklarını sürdürebilmeleri için, yazılım ürünlerini zamanında ve bütçesi dâhilinde tamamlamaları ve ürünün kalitesinin müşteri memnuniyetini sağlayabilmesi gerekmektedir. Bu gerekleri sağlamanın en etkili yöntemlerinden biri, geliştirme sürecince gerekli metriklerin toplanarak yazılım kalite maliyetlerinin hesaplanması ve kontrol altında tutulmasıdır. Akademik literatürde konuya ilişkin çalışmalar incelendiğinde, kalite yatırımlarından optimum düzeyde bir geri dönüş (Return of Investment, ROI) sağlanabildiği durumlarda, yazılım kalite giderlerinin projenin bütünü için ayrılan bütçenin %30-%50'sini oluşturduğu anlaşılmaktadır. Bu yüzden, kaliteyi sağlamak için ayrılması gereken bütçe (conformance costs) ile kalitenin yokluğunda ortaya çıkacak hataları düzeltmek için harcanması gereken bütçe (non-conformance costs) arasındaki dengenin iyi sağlanması gerekmektedir. Bu bildiride, yazılım kalite maliyet modellerinin sektördeki uygulamalarını anlatan vaka çalışmaları üzerine literatür taraması yapılarak, mevcut çalışmalardan en çok atıf alan beş makale tematik analiz yöntemiyle incelenmiş ve sonuçları sentezlenmiştir. Bu bildirinin amacı elde ettiğimiz sonuçları yazılım sektörüne sunarak bu modellerin faydalarını ve uygulanabilirliğini göstermek, uygulamaya yönelik sistematik yöntemleri Türkiye'de yaygınlaştırmaktır.

Anahtar sözcükler: Yazılım kalite maliyeti; önleme maliyeti; tespit maliyeti; başarısızlık maliyeti; değerlendirme maliyeti; yazılım süreç iyileştirme

Abstract. In order to survive in their competitive sector, software companies must complete software products on time and within budget and those products should satisfy the customers' quality expectations. One of the most effective methods to achieve this result is collecting necessary metrics during development and process these metrics to compute and control the quality costs. Based on some previous studies, the cost of quality ranges usually between 30-50% of the total development costs. The cost related to achieving quality are usually divided into two categories: prevention costs and appraisal costs. For this reason,

a good balance between the costs of achieving quality (conformance costs) and the costs due to lack of quality (non-conformance costs) should be achieved. In this paper, we investigate the most cited five papers that address practical application of software quality models using an informal survey method. The results of reviewed papers are synthesized using the thematic analysis technique. We present the benefits of such models and discuss the applicability of these results to enthusiastic software organizations. We also share some systematic approaches to help the implementation by the software companies in Turkey.

Keywords: Cost of quality (CoQ); cost of prevention; cost of detection; cost of failure; cost of evaluation; software process improvement.

1 Giriş

Yazılım projelerinde toplam bütçenin belirlendiği planlama aşamasında yazılım geliştirme maliyetleri, bakım maliyetleri, destek maliyetleri vb. maliyetler hesaplanırken, toplam bütçenin %50'lik kısmını teşkil ettiği rapor edilen [1] yazılım kalite güvence faaliyetlerinin maliyetleri sıklıkla göz ardı edilmektedir. Bu faaliyetler etkin şekilde uygulandığında toplam proje giderlerinde önemli oranda tasarruf sağlanır [2, 3].

Bu alanda yapılan akademik gözden geçirme çalışmalarından (review article) bazılarında [3] çoğu yazılım firmasındaki kalite yönetim programlarında herhangi bir yazılım kalite maliyet hesaplama modelinin kullanılmadığından bahsedilmektedir. Yazarların sektörel tecrübelerine ve Türkiye'deki pratiklere ilişkin yakın zamanda yapılmış olan bazı çalışmalara göre [5, 6, 7] Türkiye'deki yazılım firmalarının yazılım kalite maliyetlerinin nasıl hesaplandığını öğrenmeleri ve maliyetleri kontrol etmelerinin getireceği faydaları anlamaları için rehberliğe ihtiyaç duydukları anlaşılmaktadır.

Bu çalışmanın amacı, yazılım kalite maliyetinin hesaplanması alanında endüstriyle işbirliği içerisinde yapılan vaka çalışmalarını inceleyerek pratikte kullanılan yazılım kalite maliyetini hesaplama modellerinin neler olduğunu belirlemek ve önerilen farklı yöntemlerin toplam proje maliyeti üzerindeki etkisini anlayarak, Türk yazılım firmalarına bu kapsamda yapabilecekleri uygulamalar konusunda yol göstermektir.

Söz konusu çalışma bir sistematik literatür taraması (systematic literature review, SLR) veya sistematik literatür haritalama (systematic mapping, SM) çalışması değildir. Keşifçi (exploratory) literatür tarama (survey) tekniği kullanılarak bu alanda yapılan çalışmalardan en çok atıf alan beş tanesinin tematik analiz metodu [8] kullanılarak analiz edilmesinden ibarettir.

Bildirinin bundan sonraki bölümleri şu şekilde düzenlenmiştir. Bölüm 2'de kalite maliyeti ile ilgili genel bilgiler verilmiştir. Bölüm 3'de araştırma amaçları (soruları) ve yöntemlerinden bahsedilmiştir. Bölüm 4 belirlediğimiz araştırma soruları kapsamında literatürdeki çalışmalardan edinilen bulguların özetini ve analizini içermektedir. Ayrıca yazılım kalite maliyeti hesaplama modellerinin yazılım firmalarına uygulanabilirliği tartışılmıştır. Bölüm 5'te çalışmanın sonuçları ve gelecekte bu çalışmanın devamı niteliğinde yapılabilecekler anlatılmıştır.

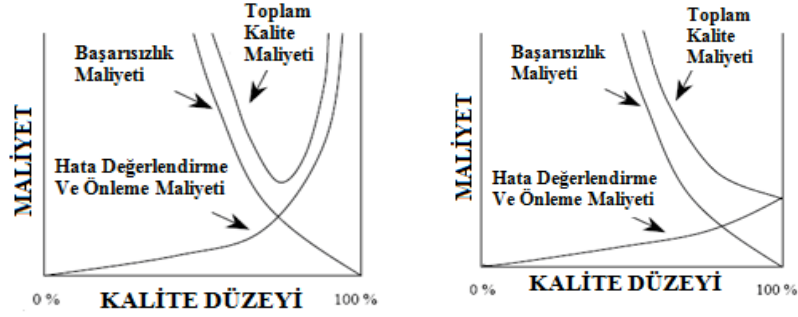
2 Kalite maliyeti hakkında genel bilgi

Kalite maliyeti, 1951 yılında J.M. Juran tarafından kullanıcı, müşteri ve diğer sistem paydaşları tarafından istenilen özelliklerin ve davranışların sağlanması için gerekli kalite süreçlerinin maliyetlerinin hesaplanma tekniği olarak tanımlanmıştır [1]. Kalite maliyeti, hem servis hem de üretim endüstrisinde kaliteyi sağlamak için yapılması gereken çalışmaların maliyetlerini kontrol etmek, kalite maliyetlerini düşürmek için yapılabilecekleri tanımlamak ve bu çalışmaların sağladığı gelir karlılığını göstermek amacıyla kullanılmaktadır [2]. Kalite maliyeti, kalitenin yokluğundan dolayı meydana gelen maliyetler ve kalitenin sağlanması için gerekli maliyetler olmak üzere iki grupta toplanabilir [2, 9]. Kalitenin yokluğundan dolayı meydana gelen maliyetler iç başarısızlık ve dış başarısızlık maliyetleri, kalitenin sağlanması için gerekli maliyetler ise önleme maliyetleri ve değerlendirme maliyetleridir. Tablo 1’de kalite maliyet kategorileri gösterilmektedir.

Tablo 1. Kalite maliyet kategorileri [2, 9]

Kategori	Tanım	Tipik yazılım maliyetleri
İç Başarısızlık	Yazılım piyasaya sürülmeden önce hataları bulup düzeltmek için yapılanlar	Bozukluk yönetimi, bozuklukları düzelttikten sonra tekrar yapılan testler ve aktiviteler (tekrar kodlama yapmak, tekrar doküman hazırlamak, tekrar denetleme yapmak...), vb.
Dış Başarısızlık	Yazılım piyasaya sürüldükten sonra hataları bulup düzeltmek için yapılanlar	Teknik destek, yazılımdaki hataları iyileştirici güncellemeler ve onarımlar, kullanıcıdan gelen bozukluk bildirimleri, üründeki hataların satışlarda sebep olduğu düşüş, şirket profiline sarsılması, hataların müşterilerin özel verilerine verdiği zarar sonucu ödenecek tazminatlar, vb.
Önleme	Yazılım kalitesini sağlamak amacıyla yapılanlar	Gerekli kalite standartlarının belirlenmesi ve müşteri kabul testleri, süreç iyileştirme, yazılım kalite güvence yönetimi, iç ve dış denetlemeler, metriklerin toplanması ve analizi, çalışanların eğitimi, vb.
Değerlendirme	Yazılımın istenilen kalite düzeyine erişip erişmediğini kontrol etmek için yapılanlar	Yazılımın kalite denetlemeleri, testler, kalite güvence prosedürlerinin ve standartlarının denetlenmesi, vb.

Kalitenin yokluğundan meydana gelen maliyetler ile kaliteyi sağlamak için gerekli maliyetler arasında ters ilişki bulunmaktadır. Kaliteyi sağlamak için yapılan yatırımları arttırdığımızda kalitenin yokluğundan meydana gelen maliyetler azalır. Şekil 1’deki geleneksel kalite maliyet modeli bu ilişkiyi göstermektedir [9]. Şekil 1’de de görüldüğü gibi yazılımda meydana gelebilecek hataları önlemek ve değerlendirmek için gerekli yazılım kalite aktiviteleri ne kadar çok uygulanırsa, hata önleme ve değerlendirme maliyetleri artar, ancak buna bağlı olarak yazılımda meydana gelebilecek hata sayısı düşer ve bu da yazılım başarısızlık maliyetlerinin büyük oranda azalmasını sağlar. Bu durum, Şekil 2’de revize edilmiş kalite maliyet modelinde gösterildiği gibi toplam kalite maliyetinin azalmasını sağlar [9].



Şekil 1. Geleneksel Kalite Maliyet Modeli [2, 9] Şekil 2. Revize Kalite Maliyet Modeli [2, 9]

3 Araştırma amaç ve yöntemi

Türkiye’de yazılım sektöründe faaliyet gösteren firmaların yazılım kalite maliyetlerini kontrol etmenin getireceği faydaları anlamaları ve maliyetlerin nasıl hesaplandığını öğrenmeleri için bir rehberlik niteliği taşıyan bu çalışmanın yapılması gerekli görülmüştür. Bu çalışmanın amacı, yazılım kalite maliyetinin hesaplanması alanında yapılan çalışmaların inceleyerek yazılım kalite maliyet hesaplama modellerini anlamak, bu modellerin kullanıldığı sektörel vaka çalışmalarını incelemek, yazılım kalite maliyet hesaplarının/kontrolünün başarısızlık maliyetlerinin azalmasına nasıl bir etkisi olduğunu anlamak ve bu modellerin yazılım sektöründeki kullanılabilirliğini ve uygulanabilirliğini göstermektir.

Bu kapsamda, yazılım kalite maliyeti modelleriyle ilgili yapılan birincil çalışmalardan en çok atıf alan beş tanesi literatür taraması (survey) yöntemi ile incelenmiş ve tematik analiz [8] yöntemi ile analiz edilmiştir.

Makalemizde yukarıda belirttiğimiz amaçlar doğrultusunda üç araştırma sorusu (ArSor, “Research Questions”) oluşturulmuştur:

- **Araştırma Sorusu 1:** Literatürde kalite maliyetlerini hesaplamak için hangi modeller önerilmiştir?
- **Araştırma Sorusu 2:** Yazılım alanında yapılan kalite maliyet modellerinin sektörel vaka çalışmaları nelerdir? Bu çalışmaların raporladığı olumlu sonuçlar nelerdir?
- **Araştırma Sorusu 3:** Başlangıç seviyesindeki yazılım firmalarında yazılım kalite maliyet hesaplama modelleri nasıl uygulanabilir?

4 Sonular ve analiz

4.1 Arařtırma Sorusu 1: Modeller

Yazılım kalite maliyet hesaplama alanında yapılan tarama alıřmalarından elde edilen sonulara gre yazılım kalite maliyet modelleri drt genel gruba ayrılabilir [3]:

- nleme-Deęerlendirme-Başarisızlık (Prevention-Appraisal-Failure, P-A-F) ya da Crosby'nin yazılım kalite maliyet modelleri,
- Fırsat /soyut maliyet (opportunity or intangible cost) modelleri,
- Sre maliyet modelleri (process cost models),
- Aktivite tabanlı maliyet (activity-based costing) modelleridir.

oęu kalite maliyet modeli P-A-F modeline baęlı olarak uygulanmaktadır. P-A-F kalite maliyet modeline gre yazılım geliřtirme srecinde nleme ve denetleme aktivitelerine yapılan yatırımlar başarısızlık maliyetlerini dřrecek, hatta dahası nleme aktivitelerine yapılan maliyetler denetleme aktivitelerine yapılan maliyetleri de dřrecektir [3].

Crosby'nin kalite maliyet modeli P-A-F modeline benzer bir modeldir. Crosby'ye gre kalite maliyeti, kaliteyi saęlamak iin gerekli maliyetler (conformance costs) ve kalitenin yokluęundan dolayı ortaya ıkan başarısızlık maliyetleri (non-conformance costs) olmak zere iki gruba ayrılarak ařaęıdaki gibi formle edilebilir:

$$C_{\text{kalite}} = C_{\text{kaliteyi_saęlama}} + C_{\text{kalitesizlik}} \quad (1)$$

Crosby'nin kalite maliyet modeli ve P-A-F kalite modelinin terminolojisi birbirinden farklı olup bu iki model birbirinin alternatifi olarak kullanılabilir [3].

Soyut maliyet modeline gre kalite maliyeti, kaliteyi saęlamak iin gerekli maliyetler, kalitenin yokluęundan dolayı ortaya ıkan başarısızlık maliyetleri ve fırsat maliyetlerinin toplamı olup ařaęıdaki gibi formle edilebilir:

$$C_{\text{kalite}} = C_{\text{kaliteyi_saęlama}} + C_{\text{kalitesizlik}} + C_{\text{fırsat}} \quad (2)$$

Fırsat maliyeti, kaynakların etkin bir şekilde kullanılmaması, rnn teslimatındaki gecikmeler gibi sebeplerden tr kazanta meydana gelen dřřler ya da mřteri kayıpları olarak tanımlanabilir.

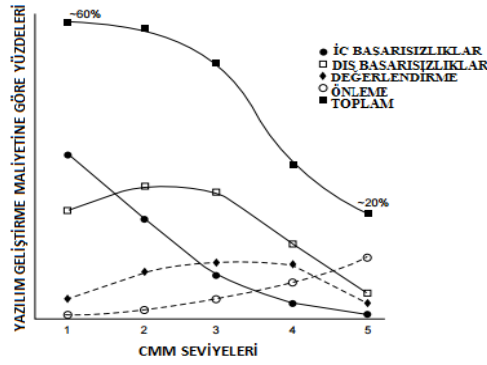
Sre maliyet modeli, rn ya da servis yerine sre zerine odaklanan bir kalite maliyet modelidir. Bu modelde kalite maliyeti, rnleri ya da servisleri retme ařamasındaki belirli bir srecin gerekli standartları ilk anda saęladıęı maliyetler (conformance costs) ve o srecin başarısızlık maliyetlerinin (non-conformance costs) toplamıdır.

Aktivite-tabanlı model, aslında bir kalite maliyet modeli olmayıp rnler arasındaki kalite maliyetlerini tanımlamak, lmek ve btce tahsis etmek amacıyla kullanılan alternatif bir yaklařımdır [3].

4.2 Araştırma Sorusu 2: Vaka Çalışmaları

Kalite maliyet modellerinin üretim, elektronik yazılım, bilgi teknolojileri vb. birçok endüstri alanındaki kullanımları ve bunların sağladıkları kazançlar farklı kaynaklarda bulunabilir (örneğin [3]). Bu bildiri yazılım maliyetinin hesaplanması alanında endüstri ile işbirliği içerisinde yapılan ve en çok atıf alan beş tane vaka çalışması incelenmiş ve sonuçları sentezlenmiştir:

- Bu vaka çalışmalarından ilki Knox'un çalışmasıdır. Knox, yazılım kalite maliyeti konusunda literatürdeki verilerin/bulguların çok sınırlı olması sebebiyle, yazılım kalite maliyetinin hesaplandığı revize edilmiş yazılım kalite maliyet modelini (Şekil 2) Yazılım Mühendisliği Enstitüsünün Kapasite Olgunluk Modeli (Capability Maturity Model, CMM) etrafında genişleterek beş CMM düzeyine uygulamış ve Şekil 3'de gösterilen teorik yazılım kalite maliyet modelini [10] elde etmiştir.



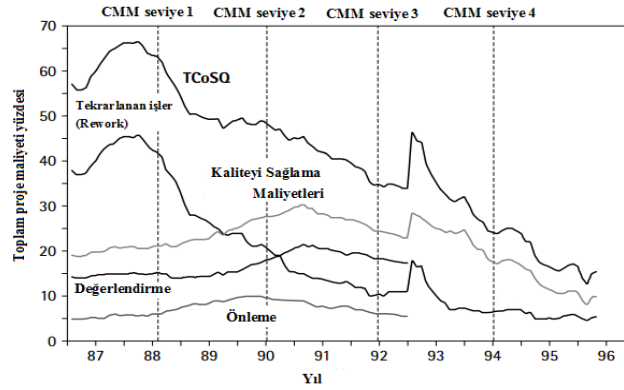
Şekil 3: Knox'un teorik yazılım kalite maliyeti modeli [2]

Knox yaptığı inceleme sonucu, CMM 1 düzeyindeki organizasyonların toplam kalite maliyetinin yazılım geliştirme maliyetinin %60'ını oluşturduğunu, bu maliyetlerin ise kalitenin yokluğundan meydana gelen iç başarısızlık ve dış başarısızlık maliyetleri olduğunu ortaya koymuştur. Knox, üretim sektöründeki kalite maliyet modeli üzerindeki deneyimlerinden yararlanarak CMM 5 düzeyindeki organizasyonların yazılım kalite maliyetlerini yaklaşık olarak %67 oranında azaltabilecekleri varsayımında bulunmuştur. Bu model, CMM 3 düzeyindeki organizasyonlar için yazılım kalite maliyetinin, toplam yazılım geliştirme maliyetinin yaklaşık olarak yarısı kadar olacağını ve kaliteyi sağlamak için gerekli maliyetlerin başarısızlık maliyetlerine oranının 0,5 olacağını önermektedir [2].

- Daha sonra 1988 yılında RES firması bünyesindeki projelerde yazılım kalite maliyetlerinin hesaplamasını sağlamak amacıyla bir süreç iyileştirme programı başlatmıştır [3]. Bu programı 15 proje üzerinde uygulamış ve 3 yıl içerisinde CMM 1 düzeyinden CMM 3 düzeyine ulaşmıştır. Bu uygulamanın sonuçlarının Knox'un teorik yazılım kalite maliyeti modelinin (Şekil 3) varsayımlarına uygun olduğu gözlemlenmiştir. RES firmasının, kalite hesaplama modellerini kullandıkları zaman elde ettikleri kazançlar şu şekildedir: (1) Firma CMM 1 düzeyinde iken kalite

maliyetleri, yazılım geliştirme maliyetlerinin %55-%67'si oranında hesaplanmış, firma CMM 3 düzeyine ulaştığında ise bu oran yaklaşık %40 civarına düşmüş ve kaliteyi sağlamak için gerekli maliyetlerin başarısızlık maliyetlerine oranı 1,5 olarak ölçülmüştür, (2) Yazılım kalite maliyetleri 8 yıl içerisinde %65'den %15'e düşmüştür, (3) Tekrar yapılan işlerin (rework) maliyeti %40'dan %6'ya düşmüştür, (4) Yazılım üretkenliği %170 oranında artmıştır.

Şekil 4'de RES'in 15 proje için uyguladığı süreç iyileştirme programının yazılım kalite maliyetleri üzerindeki etkisi grafiksel olarak gösterilmektedir.



Şekil 4. RES'in Yazılım Kalite Maliyeti ile ilgili verileri [2, 9]

- Price Waterhouse firması İngiltere'de bulunan CMM 3 düzeyindeki 19 yazılım firmasının katıldığı bir anket yaparak, yazılım kalite standartlarının maliyetleriyle bu maliyetlerden elde edilen kazançları analiz etmiştir. Bu çalışmanın sonuçları kaliteyi sağlamak için gerekli maliyetlerin (önleme ve değerlendirme maliyetleri) toplam yazılım maliyetininin %23-%34'ünü oluşturduğunu, kalitesizlik maliyetlerinin (iç ve dış başarısızlık maliyetleri) ise toplam yazılım maliyetlerininin %15'ini oluşturduğunu göstermektedir. Ancak bu çalışmaya gözden geçirme (rework) ve birim testi maliyetleri dâhil edilmemiştir [2].
- Bombardier Taşımacılık şirketinin yürüttüğü büyük bir demiryolu projesinde, yazılım kalite maliyetlerinin hesaplandığı bir vaka çalışması yapılmıştır [9]. Bu çalışmada, 88,000 adam-saat süren bu projede 1,100 den fazla yazılım parçasını geliştirmek ve kalitesini sağlamak için görevler belirlenmiş ve bu görevler analiz edilerek maliyet hesapları yapılmıştır. Yapılan hesaplamalar yazılım kalite maliyetininin toplam proje maliyetininin %33'ünü oluşturduğunu, bu maliyetin %2'sinin önleme maliyetleri, %21'nin değerlendirme maliyetleri ve %10'unun başarısızlık maliyetleri olduğunu göstermektedir.
- RBCS yazılım test, eğitim ve danışmanlık şirketi yazılımın test edilmesi için yapılan yatırımın geri dönüşünü analiz etmek amacıyla ortak olarak çalıştıkları bir şirketle işbirliği içerisinde bir vaka çalışması yayınlamıştır [1]. Bu vaka çalışmasının sonucu olarak, sistem test edilmediği zaman elde edilen yazılım kalite maliyetininin, sistem test edildiğinde elde edilen yazılım kalite maliyetininin hemen hemen 2 katı

olduğu belirtilmiştir. Bu sonuçlara göre yazılımı teslim etmeden önce hataların test edilerek bulunup onarılmasının yazılım kalite maliyetlerini büyük oranda azalttığını ve yazılım testi için yapılan yatırımdan %850 oranında bir geri dönüş elde edildiğini kaydetmişlerdir. Bu vaka çalışması da, yazılım kalite faaliyeti olarak yazılım testinin yapılarak hataların önceden bulup onarılacak hata önleme maliyetlerine ayrılacak bütçenin, yazılım değerlendirme ve başarısızlık maliyetlerini büyük oranda azalttığını göstermektedir.

Bu çalışmalardan elde edilen sonuçlara göre, kalite sistemleri CMM 3 seviyesinde olan yazılım organizasyonlarının toplam yazılım kalite maliyetleri toplam proje bütçesinin %40-%55'ini oluşturmalı ve kaliteyi sağlamak için gerekli maliyetlerin başarısızlık maliyetlerine oranı 1,5 ile 2 arasında olmalıdır [4].

4.3 Araştırma Sorusu 3: Uygulamalar

Akademik alanda kalite maliyet modellerine ilgi ne kadar yoğun olursa olsun endüstriyel alanda bu modeller yaygın olarak kullanılmamaktadır. Küçük firmalar, kalite için yeterli bütçe ayıramamakta ve kalite maliyetlerini gözlemlemek için herhangi bir girişimde bulunmamaktadırlar. Büyük firmalar ise kalitenin en önemli öncelikleri olduğunu iddia etmelerine rağmen, çok sınırlı sayıda firma kalite iyileştirme programlarının sonuçlarını incelemekte ve kaliteye yeterli bütçeyi ayırmaktadırlar. Kalite özellikle son zamanlarda firmalar arasındaki en önemli rekabet aracı olmasına rağmen, yöneticiler tarafından kalite için ayrılacak bütçenin firmaya sağlayacağı faydalar yeterince anlaşılmamaktadır [3]. Bu sebeple, bu bölümde özellikle yazılım firmalarında yönetimin kalite faaliyetlerine yönelik yatırım kararlarını desteklemek için yazılım kalite maliyet modellerinden hangisini kullanarak işe başlamalarının daha uygun olacağı ve bunları nasıl kullanacakları anlatılacaktır.

Birçok (büyük) firmanın kalite maliyet modeli olarak P-A-F kalite modelini tercih ettiği ve bu modeli uygulamaları sonucu ciddi geri kazanımlar elde ettikleri gözlenmiştir [3]. Yapılan vaka çalışmaları ve büyük şirketlerin uygulama sonuçları göz önüne alınarak, Türkiye'deki yazılım firmalarının da başlangıçta P-A-F kalite maliyet modelini uygulamaları, hem daha anlaşılır olması hem de sağladığı kazançlar sebebiyle tercih edilmelidir.

Kalite maliyet modeli belirlendikten sonra proje kapsamında yazılım ürününün sağlanması beklenen kalite özelliklerinin gerçekleştirilebilmesi için gerekli yazılım kalite güvence faaliyetleri belirlenmeli, yazılım kalite süreçleri tanımlanmalı ve firmalar bunları kendi şirketleri bünyesinde uygulamalıdır.

Sınırlı bir bütçeye sahip olan yazılım firmaları, yazılım ürününün beklenen kaliteyi sağlanması için gerekli kalite güvence faaliyetlerinden hangilerine öncelik vererek yatırım yapmaları gerektiğini nasıl anlayabilirler? Bunu anlamak için öncelikle yazılım ürününün beklenen kaliteyi sağlanması için gerekli bütün yazılım kalite güvence faaliyetleri, bu faaliyetleri başlatmak ve sürdürmek amacıyla yapılan yatırımlar hesaplanmalıdır. Bu faaliyetleri başlatmak ve sürdürmek amacıyla yapılacak muhtemel yatırımlar Tablo 3'de gösterilmiştir.

Tablo 2. Yazılım kalite güvence faaliyetlerinin maliyetleri

Yazılım Kalite Maliyetleri (SQI-initial investments)	Kalite güvence faaliyetlerini başlatmak amacıyla yapılan yatırımlar	Kalite eğitimleri, kullanılacak araçlar, materyaller vb.
Yazılım Bakım Maliyetleri (SQM-ongoing investments)	Yazılımın kalitesinin devamlılığını sağlamak amacıyla yapılan yatırımlar	Daha sonra yapılacak toplantılar, araçların yeni versiyon güncellemeleri, eğitimler vb.

Yazılım kalite güvence faaliyetleri için yapılan harcamalar yıllık bir gelirle sonuçlanmalıdır. Bu gelir, projenin satışında oluşan artış sonucu elde edilen kar ve yazılım kalite iyileştirme faaliyetleri sonucu elde edilen maliyet tasarruflarını içeren yazılım kalite gelirleridir (Software Quality Revenues, SQR).

Yazılım firmaları belirledikleri kalite güvence faaliyetlerinin her biri için gerekli kalite maliyet metriklerini belirlemelidirler. Bu metrikler Tablo 3’de gösterilmektedir. Daha sonra Denklem 3 ve Denklem 4 kullanılarak kalite güvence faaliyetlerinin karlılık indeksleri belirlenmelidir. Bu kalite güvence faaliyetlerinden, SQPI değeri 1’den büyük olanlar arasından firma bütçesi dâhilinde SQPI değeri en büyük olanları seçerek uygulamalıdır [4].

Tablo 3. Yazılım Kalite Maliyet Hesaplama Metrikleri

Metrik	Tanımı
SQPI	Yazılım kalite karlılık indeksi (Software quality probability index): Yazılım kalite gelirleri ile giderleri arasındaki farkın yazılım kalite giderlerine oranı
ROSQ	Yazılım Kalite Güvence Faaliyetlerinin Gelirleri (Return on Software Quality): Yazılım kalitesi için yapılan harcamalar ve gelirler arasındaki nakit akışı değerinin (the net present value of the SW quality cash flows, NPVCF) yazılım kalite güvence faaliyetleri için yapılan yazılım kalite maliyetleri ve yazılım bakım maliyetlerine (the net present value of investment costs, NPVIC) oranı
NPVCF	Nakit akışının net mevcut değeri (Net present value of cash flows): Yazılım kalitesi için yapılan harcamalar ve gelirler arasındaki nakit akışı
NPVIC	Yatırım maliyetlerinin net mevcut değeri (Net present value of investment costs): Yazılım kalite güvence faaliyetleri için yapılan yazılım kalite maliyetleri ve yazılım bakım maliyetleri
NPVBRC	Yazılım Kalite Gelirleri ve giderleri arasındaki fark (Net present value between return and cost)
SQI	Yazılım Kalite Yatırımları (Software Quality Investments)

$$ROSQ = NPVCF / NPVIC \quad (\text{Yazılım kalite güvence faaliyetlerinin gelirleri}) \quad (3)$$

$$SQPI = NPVBRC / SQI \quad (\text{Yazılım kalite karlılık indeksi}) \quad (4)$$

Hangi yazılım kalite güvence faaliyetlerini projeye uygulayacaklarını belirledikten sonra, firmalardaki test mühendisleri ve kalite mühendisleri bir arada çalışarak ve benzer projelerin ilgili dokümanlarını inceleyerek yazılımın kaliteyi sağlaması için

gerekli diğer maliyetleri ve kalitenin yokluğundan dolayı meydana gelebilecek başarısızlık maliyetlerini hesaplarlar. Yazılım kalite maliyetlerini Tablo 1’de gösterilen ilgili yazılım kategorilerine göre ayırarak Denklem 5’i kullanıp yazılım kalite maliyetini hesaplayabilirler [1, 3].

$$\begin{aligned} C_{\text{kalite}} &= C_{\text{kaliteyisağlama}} + C_{\text{kalitesizlik}} \\ C_{\text{kalite}} &= C_{\text{önleme}} + C_{\text{denetleme}} + C_{\text{başarısızlık}} \end{aligned} \quad (5)$$

Yazılım kalite maliyeti, yazılımın kalitesini sağlamak için yapılacak önleme ve denetleme faaliyetleri ile yazılımda hata çıkması durumunda o hataların düzeltilmesi sırasında yapılan faaliyetler ve hatalar düzeltildikten sonra tekrar yapılan kodlama, gözden geçirme, test vb. gibi faaliyetler için harcanacak iş gücü ve zaman esas alınarak adam/ay hesabı üzerinden yapılır.

Yazılım kalitesine belirli bir bütçe ayırmanın amacı yazılım ürününü teslim etmeden önce yazılımdaki bozuklukların oranını optimum düzeyde azaltarak başarısızlık maliyetlerini mümkün olduğu kadar en aza indirmektir. Bu sebeple, şirketler yazılım ürününe ayıracakları bütçeyi tahmin edebilme kabiliyetlerini arttırmalıdır. Bunu yapmanın yollarından biri eğer proje uzun soluklu bir proje ise projenin gerçek geçmiş bozukluk ve işgücü verilerini kullanmak, diğeri ise benzer projelerdeki bozukluk ve işgücü verilerini kullanmaktır.

Yazılım proje yöneticileri daha önceki projelerden elde ettikleri tahmini başarısızlık maliyetleri ve yazılım kalite maliyetleri arasındaki ters ilişkiyi Şekil 2’de gösterilen Revize Kalite Maliyet Modeli aracılığıyla yakaladıkları zaman yazılım kalite güvence faaliyetlerine yapacakları yatırım kararlarının faydalarını göreceklerdir.

Özetle, yazılım kalite maliyetlerinin artması başarısızlık maliyetlerini çok büyük bir oranda azaltmaktadır. Yazılımın beklenen kalite düzeyini sağlaması adına yapılan bu yüksek yatırımlar, değerlendirme maliyetleri ile neredeyse bütçenin %80’ini içeren başarısızlık maliyetlerinin büyük bir oranda azalmasını sağlamaktadır. Bu sebeple, yazılım için ayrılan bütçenin aşılmasına sebep olan başarısızlık maliyetlerini optimum seviyede azaltmak için yazılımın kalite gereksinimlerini sağlayacak yazılım kalite güvence maliyetlerine yapılan yatırımlar arttırılmalıdır. Sonuç olarak, yazılım proje yöneticileri kalite faaliyetlerine yapacakları yatırım kararlarını bu yazılım kalite maliyet hesaplama modelleri ile destekleyebilirler.

5 Sonuç ve gelecek çalışmalar

Türkiye’deki küçük ya da büyük ölçekli olmasından bağımsız olarak yazılım firmalarının çoğu ya kaliteye yeterince önem vermemekte ya da kalite maliyetinin nasıl hesaplanacağını bilmemektedir. Bu sebeple, kalite maliyet hesabının nasıl yapılacağı ve hangi kalite maliyet modellerini kullanmaları gerektiği ile ilgili bir çalışmaya ihtiyaç duyulmuştur. Bu sebeple, kalite maliyeti alanında bir literatür tarama (survey) çalışması yapılmıştır.

Kalite maliyet modelini kullanan firmaların rapor ettikleri kazançlar incelendiğinde yazılım kalite maliyetlerinin ve harcanan iş gücünün zaman içerisinde hemen hemen

%50 oranında bir düşüş sağladığı gözlemlenmiştir [2, 3]. Bu da yazılım kalite maliyetlerinin toplam bütçede sağladığı düşüş ve gelir karlılığını göstermektedir. Kalite maliyeti hesabı görüldüğü gibi zor ve pahalı bir işlem olmayıp aksine şirkete büyük bir kar ve fayda sağlayan bir işlemdir. Bu sebeple, yazılım proje yöneticileri proje maliyeti hesaplarında yazılım kalite maliyetlerini de katmalıdırlar.

İncelenen vaka çalışmalarından ve gözlemlenen gözden geçirme çalışmasından elde edilen bulgulara göre, P-A-F kalite maliyet modelini kullanmaları hem kolay olması hem de sağladığı faydalar açısından uygundur. Şirketler başlangıç düzeyinde bu modeli kendi projelerine ve kalite gereksinimlerine göre uyarlayıp kullanabilirler.

Bu çalışmanın devamında literatürde bu alanda yapılmış tüm birincil çalışmalarını inceleyerek bir sistematik literatür araştırma (systematic literature review, SLR) çalışması ve endüstri partnerlerimizle beraber ortak yürütebileceğimiz bir veya birkaç vaka çalışması yapmayı planlamaktayız.

Teşekkür. Hacettepe Üniversitesi Bilgisayar Mühendisliği Bölümü öğretim üyesi Doç.Dr.Vahid Garousi'ye bu çalışmanın başlangıcında yaptığı katkılardan ötürü teşekkürlerimizi sunarız.

Kaynaklar

1. RBCS | resources | articles, URL <http://rbc-us.com/resources/articles/basic-library-articles-testing-roi-what-it-managers-should-know/>
2. Houston, D. and J. Bert Keats. "Cost of software quality: a means of promoting software process improvement." *Quality Engineering* **10**(3): 563-573. (1998)
3. Schiffauerova, A. and V. Thomson. ". International Journal of Quality & Reliability Management **23**(6): 647-669. (2006)
4. Slaughter, S. A., et al. (1998). "Evaluating the cost of software quality." *Communications of the ACM* **41**(8): 67-73
5. Garousi, V., et al. "A survey of software engineering practices in Turkey." *Journal of Systems and Software* **108**: 148-177 (2015)
6. Garousi, V., Ahmet Coşkunçay, A. Can, A.B., Demirörs, O. Türkiye'deki Yazılım Test Uygulamaları Anketi. Proceedings of the 7th Turkish National Software Engineering Symposium (UYMS2013). İzmir (2013) <http://ceur-ws.org/Vol-1072/>
7. Dinçer, K., Garousi, V. Yazılım Projelerinde Başarısızlık: Kritik Başarı Faktörlerine Dayalı bir Vaka Çalışması. Proceedings of the 9th Turkish National Software Engineering Symposium (UYMS 2015), s. 59–71. İzmir (2015) <http://ceur-ws.org/Vol-1483/>
8. Cruzes, D. S. and T. Dybå (2010). Synthesizing evidence in software engineering research. Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ACM.
9. Laporte, C. Y., et al. (2012). "Measuring the Cost of Software Quality of a Large Software Project at Bombardier Transportation: A Case Study." *Software Qual. Manage* **14**(3): 14-31.
10. Knox, S. T. (1993). "Modeling the cost of software quality." *Digital Technical Journal* **5**: 9-9.