

Yazılım Projelerinde Başarısızlık: Kritik Başarı Faktörlerine Dayalı bir Vaka Çalışması

Kıvanç DİNÇER, Vahid GAROUSI

Yazılım Mühendisliği Araştırma Grubu (HUSE)
Bilgisayar Mühendisliği Bölümü, Hacettepe Üniversitesi, Ankara, Türkiye

{kivanc.dincer, vahid.garousi}@hacettepe.edu.tr

Özet: Tüm dünyada olduğu gibi ülkemizde de çok sayıda yazılım projesi ya öngörülen bütçe ve süre sınırları aşılarak veya kullanıcı beklentileri tam olarak karşılanmadan tamamlanabilmekte ya da tamamen başarısız bir şekilde sonlanmaktadır. Bu konuyu daha sistematik bir şekilde analiz etmek için, literatürde Kritik Başarı Faktörleri (KBF) adında bir tanım ortaya çıkmıştır. Bu bildiri, bir kamu kurumu tarafından ihaleyle küçük bir yazılım şirketine verilen ve sonuçta başarısız bir şekilde neticelenen bir otomasyon projesi vaka çalışması olarak ele alınmakta, projenin başarısız olmasının kök nedenleri tanımlı KBF'lere dayalı olarak analiz edilerek önemli noktalar vurgulanmaktadır. Buradaki amacımız bu analizin sonuçlarını ve tecrübelerimizi diğer yazılım mühendisleri ile paylaşarak, başka projelerin de benzer sebeplerle başarısız olmasının önüne geçilmesini sağlamaktır.

Anahtar kelimeler: Yazılım mühendisliği; proje yönetimi; yazılım projelerinde kritik başarı faktörleri; vaka çalışması / analizi

Abstract: A large number of software projects in Turkey, just like anywhere else in the world, are completed by exceeding the specified budget and time constraints and/or only partially meeting the user expectations (i.e., challenged), or fails completely. A notion called Critical Success Factors (CSF) has emerged in the literature to address the causes of project failures systematically. In this paper, we present is a web-based software automation project of a public institution, which was undertaken by a small software house through a public tender. We analyze the root causes of the failure in this software project based on CSF's. We share our experience and the results of our analysis with other software engineers through this paper, and we believe that this will help other software engineers in preventing failures in other projects due to similar reasons.

Keywords: Software engineering; project management; critical success factors in software projects; case study

1 Giriş

Yazılım projelerinin başarılı bir şekilde tamamlanmaları tüm proje paydaşları için kritik önem taşır. Maalesef dünyada olduğu gibi ülkemizde de çok sayıda yazılım projesi geç, öngörülen bütçeyi aşarak ve/veya kullanıcının beklentilerini tam olarak sağlamadan tamamlanmakta (*challenged projects*;) ya da tamamlanmadan iptal edilmekte veya teslim edildiği halde kullanılamamaktadır (*failed projects*) [1, 2].

1985'ten beri periyodik olarak yazılım projelerine ilişkin kapsamlı analiz raporları yayımlayan Standish Group'un 2001 CHAOS raporuna [1] göre, projelerin sadece %28'i başarılıydı yani istenen özellikler ve işlevlerle birlikte zamanında ve bütçesini aşmadan teslim edildi. Projelerin %23'ü başarısız oldu yani ya tamamlanamadan iptal edildi veya teslim edilmesine rağmen hiç kullanılmadı. %49'u ise zorlandı yani, öngörülen zamandan geç, planlanan bütçesinin üstünde ve/veya gerekli özellik ve işlevlerin daha aزیyla teslim edilebildi. Yine CHAOS araştırmalarına göre [2] 2012'de bu durum çok az iyileşmiştir: yazılım projelerinin sadece %39'u başarılı iken, %18'i başarısız olmuş ve %43'ü ise zorlanarak tamamlanabilmiştir.

Türkiye'de 2011 itibariyle yaklaşık 1,600 yazılım geliştirme firması bulunmaktadır [3]. Türkiye yazılım sektöründe kısa sürede yüksek bir kapasiteye ulaşmış olsa da [4-7], yazılım projelerinde dünyanın diğer yerlerindeki gibi tipik sıkıntılar ve başarısızlıklar görülmekte, projeler genelde zorlanarak bitirilebilmektedir. Yazarların tecrübelerine göre, özellikle kamu kurumları profesyonel destek almadan yaptıkları özelleştirilmiş yazılım (*custom software*) hizmet alımlarında büyük sıkıntı yaşamaktadırlar. Bu sıkıntılar teknik şartnamenin hazırlanmasın aşamasında başlamakta, projenin icra edilmesi sırasında devam etmekte, projenin kabul veya ret edilip kapanışına kadar sürmektedir.

Bu çalışmada, küçük bir yazılım firması tarafından üstlenilen bir kurumsal otomasyon yazılımının hikâyesi ele alınmıştır. Süreci ve sonuçları sistematik ve objektif bir şekilde analiz etmek için literatürde önerilen 'kritik başarı faktörleri' [8-11] adlı çalışma esas alınmıştır. Amacımız bu tecrübemizi ve analizimizin sonuçlarını diğer yazılım mühendisleri ile paylaşarak, benzer semptomları gösteren, yani başarısızlığa doğru giden, projelerin önceden tespit edilebilmesine yardımcı olmaktır.

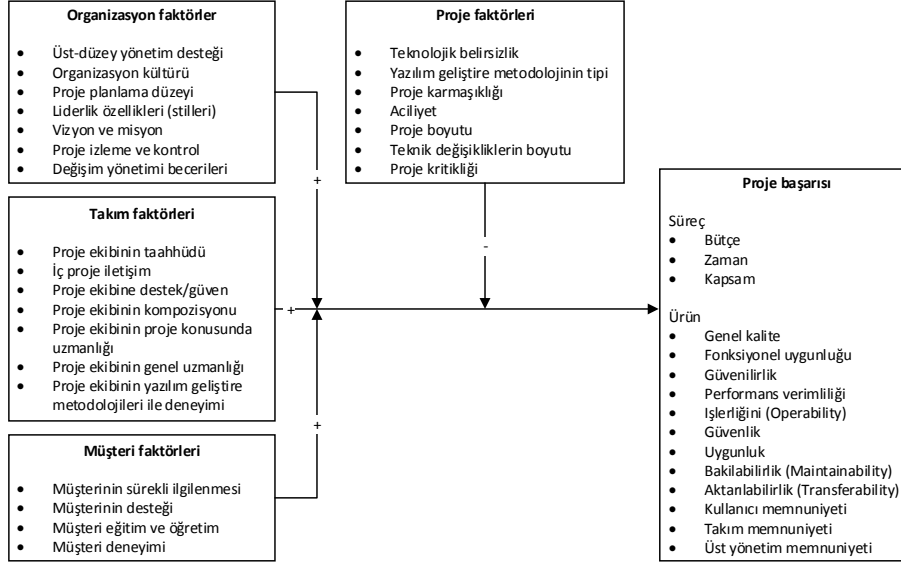
Bu bildirinin devamı şu şekilde yapılandırılmıştır. İkinci bölümde ilgili çalışmalar sunulmuştur. Üçüncü bölüm araştırma yönteminin detaylarını ve tasarımı içermektedir. Dördüncü bölümde vaka çalışmasının analiz ve sonuçları, beşinci bölümde ise bulguların özeti ve geçerliliğe tehditler sunulmaktadır.

2 İlgili Çalışmalar

Literatürde, yazılım projelerinin başarı ve başarısızlığı konuları birçok araştırmannın konusu olmuştur. Bu konuda analizleri daha sistematik kılmak için, Kritik Başarı Faktörleri (KBF) (*Critical Success Factors*) adlı bir tanım ortaya atılmıştır [8-11].

Reel [8], yazılım projelerinin başarı olması için önemli noktalar ve KBF'ler önermiştir ve çalışmasında vurguladığı iki husus özellikle önem arz etmektedir:

- “Proje başarısızlıklarının 10 işaretinden en az 7’si, tasarımın bir parçası dahi geliştirilmeden veya bir satır dahi kod yazılmadan önce belirlenebilir.”
- “Bir kalite sorununun var olduğunu anladığınız zaman, muhtemelen o sorunu düzeltmek için artık çok geç kalınmıştır.”



Şekil 1. Yazılım geliştirme projeleri için kritik başarı faktörleri tabanlı bir olasılık uyum modeli (contingency fit model). Kaynak: [9]

Ahimbisibwe ve arkadaşları [9] literatürde bulunan 148 makaleyi sistematik bir şekilde analiz ederek, yazılım geliştirme projeleri için toplam 37 KBF tespit edip, Şekil 1’de gösterildiđi gibi bunları 3+1 sınıfa ayırmışlardır: Müşteri Organizasyonu ile ilişkili faktörler, Geliştirici Ekiyle ilişkili faktörler, Kullanıcıyla ilişkili faktörler. Sonradan Proje Özellikleri ile ilişkili faktörleri de bunlara eklemişlerdir. Ayrıca KBF’ler ile proje başarısını eşleştirmek için bir olasılık uyum modeli (contingency fit model) önermişlerdir.

Akkermans ve van Helden [10] yazılım projelerinde kısır ve besleyen döngülere (vicious and virtuous cycles) odaklanmışlar ve örnek olarak bir Kurumsal Kaynak Planlama (Enterprise resource planning, ERP) Sistemi’nin üzerinde KBF’lere dayalı bir vaka çalışması raporlamışlardır. Bir çalışma da, Chow ve Cao’nun çevik yazılım geliştirme modeline (*agile methods*) özel olarak bir takım KBF’ler önerdikleri ve onları bir vaka çalışması çerçevesinde deđerlendirdikleri çalışmadır [11].

3 Araştırma Yöntemi ve Tasarımı

3.1 Çalışma Kapsamındaki Proje ve Yüklenici Firma Özellikleri

Çalışma kapsamındaki proje, bir kamu kurumu tarafından¹ ihale yoluyla bir yazılım firmasına verilen kurumsal otomasyona yönelik bir yeni sistem geliştirme (*green-field*) projesidir. Hedeflenen sistem, kurumun sayıları binlere varan iç paydaşlarının kullanımına yönelik web-tabanlı bir yazılımdır.

Müşteri kurum, yeni sisteme geçişle ilgili üst yönetim süre baskısı yaptığından dolayı projenin azami 6 ayda tamamlanmasını ve bütçe kısıtlarından dolayı da projede asgari 3 kişilik bir ekibin çalışmasını öngörmüştü. İstekli firmaların daha önce benzer bir kurumun eşdeğer bir otomasyon projesini tamamlamış olması şartını getirerek süre ve işgücü konusundaki risklerini gidermeye çalışmıştır. Kurum iki sene kadar önce ihaleye çıkmış, proje en düşük teklifi veren bu bir yazılım firmasına verilmiştir.

Projenin üç safhası vardı: (1) Firmanın elinde mevcut hazır sistemin kurulması ve kuruma adaptasyonu, (2) Kurumun farklı veritabanlarında ve Excel/Access ortamında tutulan verilerinin yeni sistemin veritabanına aktarımı, ve (3) Sistemin müşteri kurumun tüm gereksinimlerini karşılayacak şekilde genişletilmesi. Bunlar aynı zamanda yüklenici firma için hak ediş aşamalarıydılar.

Toplam süresi 6 ay olan bu projede birinci safhanın ilk ay içinde, ikinci safhanın 2 ay içinde ve üçüncü safhanın ise son 3-4 ayda icra edilmesi öngörülmüştü. Yüklenici birinci aşamayı zamanında tamamladı ve birinci hak edişi aldı. Ama veri aktarımında takıldı ve bu safha 6 ayın geri kalanı boyunca sürdü. Firmanın talebi üzerine firmaya 6 ay ek süre verildi. Bu süre de yazılım altyapısının (*framework*) tekmili, veri aktarma çalışmaları ve biraz da arayüz tasarımları ile geçti. Firma üçüncü bir 6 ay süre istedi ama müşteri kurum bu sefer sadece 3 ay ek süre verip kabule girdi.

Kabul muayene komisyonu 5 tecrübeli kişiden oluşuyordu, birisi yazılım mühendisliği ve proje yönetimi pratikleri konusunda tecrübeli bir profesyonel, diğer dördü ise ikisi yazılımcı temelli olmak üzere alan uzmanıydı. Gereken hallerde kabul muayenesine ilgili birimlerden diğer alan uzmanları da davet edildi.

Kabul muayenesi için ilk aşamada bir değerlendirme tablosu (Tablo 1) yapıldı. Kabul muayene raporunda, yapılacak değerlendirmeler dört bölüme ayrıldı:

- Fonksiyonel maddelerin değerlendirilmesi
- Fonksiyonel olmayan (non-functional) maddelerin değerlendirilmesi
- Dokümantasyon / plan ve geliştirme aşamalarının değerlendirilmesi
- GUI özelliklerinin değerlendirilmesi

¹ Gizlilik sebeplerinden dolayı, yazarların bu bildiriye proje hakkında teşhis edilebilir detaylar vermeleri mümkün değildir.

Tablo 1. Kabul-muayenesinde kullanılan değerlendirme tablosu örneği

Md. No	Ref. No	Madde Başlığı	Sınanacak Hususlar / Örnek Sınama Soruları	Kabul Puanı (1..5)	Tespit Edilen Eksikler / Hatalar
9.1		Veri Sözlüğü – Kodlu Alanlar		1	Firmaya teslim edilen ve EK-2’de işaretlenen dosyalardaki içeriğin yazılımda kodlu alanların tasarımında kullanılmadığı anlaşılmıştır.
	4.1	Veri Sözlüğü	Kodlu Alanlar ile ilgili veri dosyaları firmaya teslim edilmiştir. Bunların aktarımı kontrol edilecektir.		
	14.4	Kısaltmalar Tablosu			
	16.1	Veri Gruplama Çalışmaları			

Her bir teknik şartname maddesi için, maddenin Komisyon tarafından değerlendirilip değerlendirilmeyeceği (idari içerikli maddeler değerlendirilmemiştir), değerlendirilecekse sınamanın nasıl yapılabileceğine ilişkin notlar önceden hazırlandı. Puanlamada şu kriterler kullanıldı: 1 (tam başarısız), 2 (oldukça başarısız), 3 (kabul edilebilir ama majör düzeltmeler lazım), 4 (kabul ama minör düzeltmeler lazım), 5 (tam kabul). Bu şekilde Komisyon üyeleri kendi değerlendirmelerini sayısal bir değer üzerinden sunabildiler. Komisyon raporunda her bir madde için son puan belirlenirken, üyelerin verdiği puanların ortalamaları alındı ya da konuşularak birlikte konsolide edildi.

Kabul muayenesi sırasında alan uzmanlarının ve potansiyel kullanıcıların Yüklenici’ye sorduğu veya var olup olmadığını kontrol ettiği birçok hususun sistemde mevcut olmadığı anlaşıldı. Sonuçta Yüklenici kabul muayenesini geçemedi, harcanan süre planlananın 2.5 katına (planlanan: 6 ay, uzatmalarla birlikte gerçekleşen süre: 15 ay) ulaşmasına rağmen, şartname maddelerinin üçte birinin hiç, üçte birinin ise büyük ölçüde karşılanmadığı tespit edildi. Müşteri Kurum maalesef projeyi bu şekilde sonlandırmak zorunda kaldı ve firmaya ödenen hakkedişin faiziyle geri alınması için hukuki işlem başlatıldı.

Bir vaka çalışması olarak, bu makalenin devamı sistematik bir yöntem kullanarak bu projenin başarısızlık sebeplerini bulmayı hedeflemektedir.

3.2 Araştırma Yöntemi

Araştırma yöntemi olarak deneysel (*empirical*) yazılım mühendisliği yöntemlerinden olan keşif ve açıklayıcı vaka çalışması (*exploratory and explanatory case study*) [12] uygulanmıştır. Yazarlarımız bu konuda uluslararası birikime sahiptir. Bu tür vaka çalışmalarında amaç bir durum ya da bir sorun için çoğunlukla nedensel ilişki şeklinde açıklama aramak ve yeni araştırmalar için fikir ve hipotezler üretmektir [12].

Araştırmamızın amacı Ahimbisibwe ve arkadaşlarının [9] Kritik Başarı Faktörleri (KBF) sınıflandırmasını kullanarak, çalışma kapsamındaki projenin başarısızlık sebeplerini sistematik şekilde bulmaktır. Bu hedef kapsamında, iki araştırma sorusu ortaya konulmuştur:

- Soru 1: KBF esas alındığında bu projenin başarısızlık sebepleri nelerdir?
- Soru 2: Yapılacak çıkarımlarla benzer bir başarısızlığı diğer projelerde önlemek nasıl mümkün olur?

4 Vaka Çalışma Analiz ve Sonuçları

4.1 Projenin Kritik Başarı Faktörlerle Değerlendirilmesi

Çalışma kapsamındaki proje Şekil 1'deki KBF modelini kullanarak yazarlar tarafından dikkatle değerlendirilerek ve elde edilen sonuçlar Tablo 2, 3, 4 ve 5'te gösterilmiştir. Orijinal çalışmada [9] önerilen 37 KBF'den, bu proje için uygun gördüğümüz 23 faktör bu tabloda 0...4 arası (5-noktalı) bir Likert ölçeği ile değerlendirilmiştir. Açıklamalar değerlendirme notlarının dayandığı gerekçeleri içermektedir. Değerlendirmeler yazılım mühendisliği nicel ve nitel yöntemleri [12] kullanılmak suretiyle, her iki araştırmacı (yazarlar) tarafından bağımsız olarak yapılmış ve tartışılmak suretiyle konsolide edilerek not değerlendirmesinde öznellik asgariye indirgenmiştir.

Ayrıca, aynı kaynakta [9] sunulan proje başarı ölçütlerinin bu proje için değerlendirmeleri Tablo 6'da gösterilmektedir. Yukarıda Bölüm 3.1'de proje ve firma hakkında verilen bilgiler bu değerlendirmelerde göz önüne alınmıştır.

Tablo 2. Müşteri organizasyonu ile ilişkili KBF'lere göre değerlendirme sonuçları

Faktörler	Not	Açıklama
Üst-düzye yönetim desteği	4	<i>Çok yüksek</i> / Kamu Kurumunun en üst düzey yöneticisi iki aylık periyotlarla, görevlendirdiği yardımcısı ise haftalık periyotlarla yapılan proje ilerleme toplantılarına aktif katılım sağlamışlardır.
Organizasyon kültürü	4	<i>Adaptif / esnek</i> / Yüklenici küçük bir şirket olduğundan dolayı, şirket sahibi ve çalışanların kendi aralarındaki ilişki arkadaş gibiydi. İlişkilerde formalite ve bir alt/üst ilişkisi gözlenmemiştir.
Vizyon ve misyon	4	<i>Önceden proje için oldukça iyi tanımlanmış vizyon ve misyon</i> / Projenin başından itibaren yeni otomasyon sisteminin kurumda yürütülen proje kapsamındaki tüm iş ve işlevleri desteklemesi öngörülmüş idi. Üst yönetim tarafından bu otomasyon sisteminin Türkiye'deki benzer kurumlar arasındaki en gelişmiş sistemlerden birisi olması hedefleniyordu.
Proje planlama düzeyi	1	<i>Proje başlangıcında çok az planlama</i> / Müşteri Kurum yeterli düzeyde proje planlama yetenek ve alışkanlıklarına sahip değildi. Projeyi üç safhaya ayırarak ihaleye çıktı. Yüklenicinin sadece sözleşme yükümlülüklerini yerine getirmek amacıyla yaptığı az sayıda (<10) aktiviteden oluşan çok üst düzey bir proje planı yeterli gördü.
Proje izleme ve kontrol	1	<i>Az (zayıf)</i> / Proje planı az sayıda üst düzey aktiviteden oluştuğundan dolayı izleme ve kontrol noktasında da zafiyet oluşturdu. Proje ilerleme toplantıları detaylı plan olmayınca daha çok bazı teknik problemlerin tartışıldığı yapısal olmayan (unstructured/unsystematic) beyin fırtınası (brainstorming) toplantılarına dönüşüyordu.
Değişim yönetimi becerileri	1	<i>Değişim yönetimi planlı ve sistematik değil (ad-hoc)</i> / Yüklenici toplantılarda ve çalışmalar sırasında kullanıcı taleplerini dinliyor ve yazılı talepleri alıyordu. Ancak bunları iş planına ekleyip eklenmediği veya eklendiyse yapılmı durumu kontrol altında tutulmadı, gözden geçirme toplantılarında irdelenmedi. Kurumun yazılı olarak Yüklenici'ye ilettiği taleplerin analiz edilmeden ham bir şekilde tutanaklarda kaldığı, kabul aşamasında ortaya çıktı. Diğer yandan projenin ilk uzatmasının ortasında iki personelden biri ayrılmış yerine yeni bir personel başlamış idi ama sistematik bir iş devri yapılmadı.
Toplam	15 (kazanılmış not toplamı) / 24 (azami not toplamı) = %63	

Tablo 3. Geliştirici ekiple ilişkili KBF'lere göre değerlendirme sonuçları

Faktörler	Not	Açıklama
Proje ekibinin taahhüdü	2	<i>Orta</i> / Yüklenicinin proje ekibi 3 kişiden oluşuyordu. Firmanın merkezi farklı bir şehirdeydi. Şirket yöneticisi proje dışında genel şirket yönetimi ve iş geliştirmeden sorumluydu, haftada bir kaç gün gelip müşterinin verdiği ofiste çalışıyordu, vaktinin sadece bir kısmını projeye ayırabilirdi, tam zamanlı konsantre olamadı. İkincisi şirket merkezinde çalışıyordu, vaktini daha çok yeni sistem altyapısını geliştirme üzerinde harcadığı söylendi, kabul sırasında sahadaki projelerle ilgili sistem desteğinden de onun sorumlu olduğu anlaşıldı. Üçüncü personel ise tam zamanlı olarak Kurum personeliyle birlikte sistemin adaptasyonu üzerinde çalıştı.
İç proje iletişim	3	<i>Yüksek</i> / Küçük bir takım olmasından dolayı ekip üyeleri birbirleriyle yakın çalışıyorlar ve iyi sözlü iletişim kuruyorlardı. Ancak bu iletişim hiçbir şekilde sistematik veya kurumsal nitelikte değildi, bazı talepler kaybolabiliyordu.
Proje ekibine destek/güven	3	<i>Orta</i> / İki ekip üyesinin de iş tanımı net idi, dolayısıyla proje gerekleri doğrultusunda kendi iş kalemleriyle ilgili tam yetki ve sorumluluk sahibi idiler.
Proje ekibinin kompozisyon uyumluluğu	1	<i>Ekip oluşumu geçmişteki tecrübelerle dayanarak yapıldı, ama yine de iyi yapılmadı</i> Proje şartnamesine göre projenin 1 + 2 kişilik bir ekip ile yürütülmesi öngörülmüştü. Yüklenici asgari elemanla işi bitirmeye çalıştı. Şirketin yöneticisi zaman zaman aktif olarak veri aktarımı ile ilgili kod yazıyordu. Personelden birisi sistem altyapısını (framework) geliştirmeye çalışıyordu. Diğeri ise firma sahibi ile birlikte yazılım arayüzleri ve iş kurallarının gerçekleştirimi üzerinde çalışıyordu.
Ekibin proje konusunda uzmanlığı	2	<i>Orta</i> / Şirket sahibi ve çalışanlardan birisi alana vakıf idiler, daha önce bu alanda farklı projelerde çalışmışlardı. Diğer personel ise böyle bir işi ilk kez yapıyordu.
Ekibin yazılım mühendisliği bilgisi	1	<i>Çok az</i> / Firmanın hiçbir personeli sistematik yazılım mühendisliği bilgisine sahip değildi. Proje boyunca bu konuda telkin edilen iyi pratikleri de uygula(ya)madılar.
Toplam	12 (kazanılmış not toplamı) / 24 (azami not toplamı) = %50	

Tablo 4. Kullanıcıyla ilişkili KBF'lere göre değerlendirme sonuçları

Faktörler	Not	Açıklama
Müşterinin sürekli ilgilenmesi	4	<i>Çok yüksek</i> / Kullanıcılar haftada 1-2 iş gününü Yüklenici personeli ile çalışarak ve mevcut sistemin ve verilerinin analizi ile ilgili Yüklenici'nin yapması gereken çalışmaları kendileri yaparak geçirdiler.
Müşterinin desteği	4	<i>Çok yüksek</i> / Ortak toplantıların yanı sıra, yapılacak işlerin belirlenmesi, nasıl yapılacağı belirlenmesi, vb. gibi konuların firmaya aktarımı için Kullanıcıların ayrıca çalışma yapmaları gerekti. Gereken tüm çabayı gösterdiler.
Müşteri BT'de bilgi ve deneyimi	3	<i>Yüksek</i> / Yüklenici'ye destek olan ekipten bazıları alan uzmanı idiler fakat BT konusundaki tecrübeleri kısıtlı idi. Ancak bir de kurumun BT departmanından olup, mevcut otomasyon projelerine katılmış olan çok yetkin kişiler de vardı.
Müşterinin kendi alanında deneyimi	4	<i>Çok yüksek</i> / Bazı alan uzmanları 20 yıldan fazla tecrübeye sahipti, işe liderlik yapan müşteri temsilcilerinin ortalama tecrübesi ortalama 10 yıl civarındaydı.
Toplam	15 (kazanılmış not toplamı) / 16 (azami not toplamı) = %94	

Tablo 5. Proje özellikleriyle ilişkili KBF'lere göre değerlendirme sonuçları

Faktörler	Not	Açıklama
Teknolojik belirsizlik	1	<i>Yüksek</i> / Yüklenici daha önce benzer projeler gerçekleştirmişti. Ancak bu projede denenmiş yazılım altyapısını kullanmadı. Her bir müşteri için ayrı yazılım geliştirmek için yeni geliştirmekte olduğu jenerik ve özelleştirilebilen sistem altyapısını (framework) kullanmaya karar vermişti. Muhtemelen internetten açık-kaynak bir altyapıyı aldı ve uyarlamaya çalıştı. Bu altyapıyı kullanıcıların isteklerine göre uyarlamaya çalışırken bu altyapının temel teknolojisine hakim olmadığından dolayı bazı belirsizliklerle mücadele etmek zorunda kalmıştı. Kurum da bunun getirdiği riskleri fark edemedi. Yüklenicinin harcadığı efor ve

		zamandan bu teknoloji ve altyapıyı istediği gibi kullanamadığı sonradan anlaşıldı.
Yazılım geliştire metodolojinin tipi	1	<i>Hiç</i> / Şirket hiçbir tanımlı biçimsel (formal) metodolojiye göre çalışmamıştı. Şartname'ye göre firmanın gereksinim dokümanı ve tasarım dokümanı hazırlaması ve testleri biçimsel olarak icra etmesi gerekliliği vardı. Ancak bunların hiçbirini yerine getirmemişti. Müşteri de bunlara gereken önemi vermemiş, teslim edilmediği halde bunların arkasını aramamıştı. Kabul aşamasında istendiğinde tasarım dokümanı olarak sadece veritabanının mevcut halinden tersine mühendislik yoluyla üretilmiş E-R diyagramlarını teslim etti. Bunlar alt düzey detay içeren tasarımlardı. Bu diyagramlar otomatik üretildiği için gerçek alanları ve gerçek tabloları görebilmek mümkün değildi. Firma teslim aşamasına kadar hep tasarsız/plansız (ad-hoc) test yapmıştı. Test senaryoları ve test senaryo (case)'leri yoktu. Aslında doğal olarak gereksinimlerin toplanması ve analizi aşamasında iş süreçleri çıkarılmadığı, kullanıcı senaryoları veya kullanıcı durumları (use cases) yazılmadığı için, gereksinimleri belli olmayan böyle bir yazılımın test senaryoları de yazıl(a)mamıştı. Dolayısıyla firma yazılımın ne kadar çalışıp çalışmadığını kendisi de bilmiyordu.
Proje karmaşıklığı	1	<i>Yüksek</i> / Projede hem veriler hem de sürekli değişen ve seneden seneye veya kurumun bölümünden bölümüne farklı şekilde yorumlanıp uygulanan mevzuatın tanımlı iş kurallarına dönüştürülmesinin zorluğu projenin karmaşıklığını artırıyordu. Projede uzunca bir süre Yüklenici ve Müşteri Kurum tarafından veri aktarımının kapsamı konusunda tartışılmıştı. Kurumun elinde geçmiş 40 yıla varan geçmişe sahip verileri mevcut idi. Verilerin son 10 yılı aktif, diğerleri pasif olarak kabul edilmesi gerektiği görülmüyordu. Ancak Kurum zaman zaman pasif veriye de ihtiyaç duyuluyor olduğundan ve istatistiklerin tam olmasını istediğini belirterek bu verilerin hepsinin aktarılmasını istemişti. Aktarım sırasında verilerin eskiye doğru gittikçe çok sayıda eksik alan bulunduğu anlaşılmış, anlaşılmayan bir sebepten dolayı Yüklenici tarafından diğer veriler analiz edilerek eksik alanlar doldurulmaya (!) çalışılmıştı. Bu sırada hem veriler bozulmuş hem de çoğu yerde veritabanı tablolarının birincil/ikincil anahtar ilişkilerinde ve zorunlu/tercihe bağlı alanlarda sorunlara yol açmıştı. Daha doğrusu tüm alanlar tercihe bağlı alan olmuş denilebilirdi. İkinci olarak Kurumun verileri birbirinden bağımsız en az 4 veritabanına, hatta bunlar da kendi içinde ikiye bölünebilecek olmasına rağmen, verilerin hepsi birlikte/toplu olarak aktarılmaya çalışılmıştı. Eğer her bir bölüm üzerinde ayrı ayrı çalışılsa idi, en küçükten başlamak üzere adım adım gidilebilir ve veri aktarımı 4-8 farklı aktiviteye ayrılmış olarak planlanıp takip edilebilirdi. Veri aktarımıyla ilgili faaliyetler her zaman tek bir aktivite gibi görülüp bu şekilde çalışıldı. Dolayısıyla bu aktivite 2.aydan 15.ay sonuna kadar devam etmişti.
Aciliyet	2	<i>Oldukça acil</i> / Özellikle üst yönetim ilk başta 6 ayda biteceğini umarak başladığı projenin bir an önce tamamlanmasını ve kullanıma girmesini istiyordu. Muhtemelen ihaleye çıkarken de – belki isteklilerle de görüşerek - en kısa süre verilmek suretiyle bu amaç gözlenmişti. Birkaç küçük firma kabul edilebilir olmamasına rağmen bu süreyi kabul ederek projeye teklif vermişti. Bu da yönetimi 6 aylık sürenin makul olduğu konusunda daha da ikna etmişti.
Proje boyutu	2	<i>Orta boyut (20 kişi)</i> / Projede Kurum tarafındaki katılımcıları da göz önüne alırsak yaklaşık 20 kişi katkı sağlıyordu. Bunlardan sadece 3'ü Yüklenici tarafındaydı.
Teknik gereksinim değişikliklerin boyutu	1	<i>Gereksinimler çok sık değişiyor (% 80'dan fazla)</i> / Kullanıcı gereksinimlerini ifade eden şartnameden sistemin gereksinimleri çıkartılmamıştı. Gereksinim analizi yapılmadığı için Kullanıcıların beklentileri önceliklendirilmemiş ve belki bazıları kapsamdan çıkartılmamıştı. Kullanıcıların beklentileri detaylandırılmadığı için kabul muayenesi sırasında kullanıcıların sorduğu veya var olup olmadığını kontrol ettiği birçok husus yazılımda mevcut değildi. İlginç bir şekilde Yüklenici elinde mevcut altyapıyı – eski tecrübelerine dayalı olarak – az bir eforla işi bitirebileceğini düşünmüştü. Müşteriye özel ciddi bir uyarılama yapmayı planlamamıştı. Yazılımın arayüzlerinde kullandığı terminoloji Kurumunkinden farklı idi, iş ve işlem sıraları kullanıcının alıştığından farklı idi. Uzun yıllardır belli bir terminoloji ve oturmuş bir süreç olduğu için Kurum kullanıcıları doğal olarak bundan vazgeçmek istemedi.

Proje kritikliği	1	<i>Kritik /</i> Mevcut otomasyon sistemi sınırlarına geldiği için bu projenin başarılı bir şekilde bitmesi müşteri için çok kritik idi.
Toplam	9 (kazanılmış not toplamı) / 28 (azami not toplamı) = %32	

Tablo 6. Proje başarı ölçütlerinin değerlendirilmesi

Faktörler	Not	Açıklama
Bütçe	0	<i>Tamamen bütçesi üzerinde (birkaç kat ek maliyet) /</i> Başta 6 aylık bir bütçe ve 6 x 3 = 18 adam aylık bir bütçe ile başlanmıştı. Müşteri Kurum aradaki farkı ödemek zorunda kalmasa da Yüklenici proje için 6x15 = 90 adam aylık iş yapmak zorunda kaldı.
Zaman	0	<i>Tamamen geç (birkaç aylar) /</i> Yine zaman olarak 6 aylık bir öngörü ile başlayan proje, 15 ay yani öngörülenden 2.5 kat daha uzun sürdü. Ve başarısız oldu.
Kapsam	1	<i>Anlaşılan kapsamım dışında (%25'ten fazla, %75'den az) /</i> Kabul muayene sonuçlarına göre yazılım şartname maddelerinin ancak %35'ini karşılayabildi.
Genel kalite	2	<i>Orta /</i> Komisyon yazılımın genel kalitesini 5 üzerinden 2-3 olarak değerlendirdi.
Fonksiyonel uygunluğu	2	<i>Orta /</i> Kabul muayene komisyonu yazılımın gerçekleştirilen fonksiyonlarının uygunluğunu 5 üzerinden 3 olarak değerlendirdi.
Güvenilirlik	1	<i>Az /</i> Kabul muayene sırasında gözlenen hatalar, yanlış hesaplamalar, ekrana çıkan hata izleri, vb. yazılımın güvenilirliğinin oldukça düşük olduğunu gösterdi.
Toplam	6 (kazanılmış not toplamı) / 24 (azami not toplamı) = %25	

4.2 Kök-Neden Analizi (Root-Cause Analysis) ve Öğrenilen Dersler

Bölüm 3.2'de sorduğumuz Araştırma Sorusu 1'i bu bölümde cevaplamak için bu projenin başarısızlık sebepleri belirlemeye çalışacağız. Bir önceki bölümde, projenin durumunu kritik başarı faktörlerine (KBF) göre değerlendirdik. Tecrübelerimizi ve analiz sonuçlarını diğer yazılım mühendisleri ile paylaşmak için, literatürdeki benzer çalışmalar gibi (örneğin: [13]) kısa bir kök-neden analizi yapılacaktır.

Çalışma kapsamındaki projede, Tablo 2, 3, 4 ve 5'te gösterildiği üzere, farklı KBF'lerin başarı veya başarısızlığına etkisinin aynı olmadığı anlaşılmaktadır. Başarısızlığa en çok etkisi olan faktörler, bu tablolarda 0 veya 1 notu alan KBF'lerdir. O faktörleri Tablo 7'de listeleyip ve bunların üzerinde, 'sebe-sonuç haritalama' (cause mapping) [14] metodunu kullanarak, kök-neden analizi yapılmıştır. Dolayısıyla, Tablo 7'de yer alan faktörleri kritik başarısızlık faktörleri olarak adlandırabiliriz.

Tablo 7. Projenin başarısızlık sebeplerinin kök-neden analizi

Projenin başarısızlık sebepleri	Kök-neden analizi
Proje başlangıcında yetersiz planlama yapılmıştı	Yüklenici Firma başlangıçta çok az planlama yapmıştır. Müşteri bunu ihmal etmiştir. Firma yöneticisi, 20+ senelik yazılım tecrübesine rağmen, ekibinin plan-odaklı (plan-driven) bir yöntem izlenmesini sağlayamamıştır. Özellikle, Proje-Yönetimi Bilgi Tabanı Kılavuzu (Project Management Body of Knowledge, PMBOK Guide) [15] sunulan planlama yöntemlerinden "en iyi uygulamalar (best practices)" kullanılmamıştır.
Proje izleme ve kontrolü zayıftı	Proje planı az sayıda üst düzey aktiviteden oluştuğundan dolayı izleme ve kontrol noktasında da zafiyet oluşmuştur. Müşteri yeterli takip yapmamıştır. Proje ilerleme toplantılarında Yüklenici bir şekilde Kurum temsilcilerini çok az işi kaldığına, altyapıdaki bazı problemleri aşarsa, hemen hemen tüm işin haftalar mertebesinde biteceğine – hemen her toplantıda - inandırmayı başarmıştır. Proje izleme ve kontrolü için PMBOK'ta [15] önerilen yöntemler sistematik bir

	şekilde kullanılmamıştır.
Değişim yönetimi tasarsız/plansız (ad-hoc) şekilde yapıldı	Bu konuda, üç maddeyi tespit edilmiştir: <ul style="list-style-type: none"> • Personel devir yönetimi (<i>staff turnover management</i>) • Gereksinimleri değişim yönetimi (<i>requirements change management</i>) • Kaynak kod değişim yönetimi Değişim yönetimi tasarsız yapıldığı için, değişiklikler projeye yüksek negatif etki (change impact) yapmıştır. Örneğin, Yüklenici'nin tecrübeli bir personeli proje ortasında projeden ayrılmış ve iş tecrübesi olmayan yeni bir eleman alınmıştır. Bu değişimde, eski elemanın edindiği bilgiler yeni elemana çok az seviyede aktarılmış ve neticede yeni elemanın projeye faydalı iş çıkarması zaman almıştır.
Proje ekip oluşumu iyi yapılmamıştı	Mevcut Kamu İhale Kanunu yazılım alımı ihalelerinde çalışan sayı ve özelliklerinin nitelendirilmesini / tanımlanmasını öngörmekle birlikte, bunların sayısal olarak yeterli olup olmadığını değerlendirilmesi için bir enstrüman içermemektedir. Buna ek olarak ayrılan ekip de tam zamanlı olarak çalıştırılmayınca projeyi bitirmeye kafi gelmemiştir. Bu projenin başarısızlığındaki en önemli faktörlerden birisi proje için öngörülen kaynakların yetersizliğidir.
Proje ekibinin genel yazılım mühendisliği bilgisi çok azdı	Proje ekibinin genel yazılım mühendisliği ve iyi pratikler konusunda bilgili olmaları gerekiyordu. Bunlara önem verilmemiş, Kurum da gereken kontrolleri yapmamış ve işler tamamen <i>ad hoc</i> bir şekilde yürütülmeye çalışılmıştır.
Hiçbir yazılım geliştire metodolojisi kullanılmamıştı	Projede tanımlı bir biçimsel yazılım geliştirme metodolojisine göre geliştirilmemişti. Bu da çoğunlukla, proje ekibinin genel yazılım mühendisliği bilgisinin çok az olmasından kaynaklandı. Ayrıca, müşteri de bu konuda öyle bir biçimsel yöntemin kullanılmasını zorlaması gerekiyordu.
Yüksek proje karmaşıklığına karşı tedbir alınmamıştı	Projede yüksek karmaşıklık seviyesini kontrol etmek için karmaşıklık yönetimi (<i>complexity management</i>) yöntemlerinin kullanılması gerekiyordu. Sürenin yeterinden az olması karmaşıklığın yönetilemeyişi daha da artırdı.
Gereksinimler tanımlanmamış ve değişimi yönetilememişti	Başlangıçta gereksinim analizi yapılmadığı ve gereksinimler önceliklendirilmediği için müşterinin birçok talebinin dikkate alınmadığı kabul aşamasında anlaşılmıştır. Bunların birçoğu sonraki süreçte toplantılar ve çalışmalar sırasında kullanıcılardan sözlü veya yazılı şekilde iletilmiştir. Bunlara yönelik planlama yapılmadığı için firmayı zorlamıştır. Bu sorunu çözmek için, gereksinim belirleme sürecinin iyi işletilmesi ve biçimsel değişim etki analizi (<i>change impact analysis</i>) [16] yöntemlerinin kullanılması gerekiyordu.

5 Tartışma: Bulguların Özeti ve Tavsiyeler

Bölüm 4'de sunulan vaka çalışma analizlerimize dayalı olarak, bu bölümde başka projelerin de benzer sebeplerle başarısız olmasının önüne geçilebilmesi için bulgularımızın özetlerini ve tecrübelerimizi diğer yazılım mühendisleri ile paylaşacağız.

Şekil 2 bulguları özetleyerek projenin başarısızlık sebepleri ve ölçütlerini bir sebep-sonuç (cause-effect) diyagramı olarak göstermektedir. Bölüm 4'de bahsedildiği gibi, değerlendirmemize göre KBF'lere dayalı toplam not $51/92 = \%55$ hesaplandı. Hâlbuki proje başarı ölçütlerinin toplam $6/24 = \%25$ hesaplandı. Bu durum projenin başarı seviyesinin KBF'lere göre yapılan kestirimden daha düşük olduğunu göstermektedir. Diğer bir deyişle, bağımlı değişken (dependent variable) olarak, bu projenin başarı durumu, bağımsız değişkenler (independent variable) olan projenin geliştirme KBF'lerine göre daha düşük olmuştur. Daha detaylı analiz edersek, Şekil 2'nin sol tarafında yer alan bağımsız değişkenlerin yarısı 3 veya 4 notu (iyi seviyede) almış olmasına rağmen, sağ taraftaki bağımlı değişken (projenin başarı durumu) düşüktür.

Diğer bir deyişle, proje altyapı ve ortamına ilişkin faktörler arasında birçok faktör (örneğin: müşteriye ve misyona ait faktörler) iyi seviyede olsa bile, görüyoruz ki proje sonucu başarısız olmuştur. Bu bir meşhur atasözünü hatırlatmaktadır: “Kurunun yanında, yaş da yanar”. Yani KBF’lerden bir kısmının bile kötü olması projenin başarısız sonuçlanmasına yol açabilmektedir.

Proje kritik başarı faktörlerine göre değerlendirme
Bağımsız değişkenler (independent variables)

<ul style="list-style-type: none"> • Üst-düzyer yönetim desteđi • Organizasyon kültürü • Vizyon ve misyon • Proje ekibinin taahhüdü • İç proje iletişim • Proje ekibine destek/güven • Müşterinin sürekli ilgilenmesi • Müşterinin desteđi • Müşterinin kendi alanında deneyimi 	4
<ul style="list-style-type: none"> • Aciliyet • Müşteri BT’de bilgi ve deneyimi 	3
<ul style="list-style-type: none"> • Proje ekibinin proje konusunda uzmanlığı • Teknolojik belirsizlik • Proje boyutu 	2
<ul style="list-style-type: none"> • Proje ekibinin genel yazılım mühendisliği bilgisi • Proje karmaşıklığı • Proje ekibinin kompozisyon uyumluluđu • Teknik gereksinim deđişikliklerin boyutu • Proje kritikliği • Proje planlama düzeyi • Proje izleme ve kontrol • Deđişim yönetimi becerileri • Yazılım geliştire metodolojinin tipi 	1
	0

Proje başarı ölçütlerinin değerlendirme
Bağımlı deđişkenler (dependent variables)

	4
	3
<ul style="list-style-type: none"> • Genel kalite • Fonksiyonel uygunluğu 	2
<ul style="list-style-type: none"> • Güvenilirlik • Kapsam 	1
<ul style="list-style-type: none"> • Bütçe • Zaman 	0

Şekil. 2. Projenin başarısızlık sebepleri ve ölçütlerine ait neden-etki (cause-effect) diyagramı.

Araştırma Sorusu 2’yi cevaplamak üzere, bu projenin başarısızlık sebeplerini diğer projelerde önleyebilmek için aşağıdaki tavsiyeleri önermek istiyoruz:

- Bir projenin başlangıcında tüm ekibin (hem müşteri hem de firma) KBF’lerin hepsine dikkat etmesi gerekir. Çünkü KBF’lerin çođu iyi seviyede olsa bile, birkaç tane düşük seviyede KBF, kolayca projenin başarısını riske atabilmektedir.
- Proje başarısızlığında sadece geliştiren firma ve ekibin sorumlu olmadığına ilişkin bir örneđi bu projede de gördük. Müşteri bu projede tüm süreçte çok ilgilenmiş olduğuna rağmen, yazılım süreçlerine ilişkin disiplinle ilişkili birçok şartname maddesini yeterince zorlamamıştır. Ayrıca, müşteri proje izleme ve kontrolünün asıl işlevini yeterince ciddiye almamıştır. Dolayısıyla, yazılım geliştiren firma ile beraber müşteri de tüm sorumluluklarını, KBF’leri göz önüne alarak, yerine getirmelidir.

6 Sonuç ve Gelecek Çalışmalar

Bu bildiriye, bir kamu kurumu tarafından ihaleyle bir yazılım şirketine verilen ve sonuçta başarısız bir şekilde neticelenen bir otomasyon projesi bir vaka çalışması olarak ele alınmış, literatürde tanımlı Kritik Başarı Faktörleri'ne (KBF) dayalı olarak projenin başarısız olmasının kök-nedenleri analiz edilerek önemli noktalar vurgulanmıştır. Umarız bu analizin sonuçları ve tecrübelerimiz diğer yazılım mühendislerine, başka projelerin de benzer sebeplerle başarısız olmalarını önceden öngörebilmekte fayda sağlar. Gelecekte yapılması planlanan çalışmalar arasında, bu çalışmanın başka projelere daha uygulanıp, birden fazla sonucun birbiriyle kıyaslanması bulunmaktadır.

Kaynaklar

- [1] The Standish Group, "Extreme CHAOS," http://www.standishgroup.com/sample_research/showfile.php?File=extreme_chaos.pdf, 2001, Last accessed: Oct. 2014.
- [2] The Standish Group, "CHAOS Manifesto 2013: Think Big, Act Small," 2013.
- [3] G. Tirpançeker, "Türkiye Yazılım Sektörü ve Yazılımın Yarattığı Katma Değerler," *Stratejik Düşünce Enstitüsü*, http://www.sde.org.tr/userfiles/file/Gulara_Tirpanceker_SDE_2011Aral%C4%B1k-2.pdf, 2011.
- [4] V. Garousi, A. Coşkunçay, A. B. Can, and O. Demirörs, "A Survey of Software Testing Practices in Turkey," in *Turkish National Software Engineering Symposium (Ulusal Yazılım Mühendisliği Sempozyumu, UYMS)*, 2013.
- [5] N. Sökmen, "Competency level of the software industry in Turkey and guidelines for enhancement of companies and the sector (in Turkish: Türkiye'de Yazılım Üreticilerinin Yetkinlik Düzeyi Firmaların ve Sektörün Gelişimi)," 2010.
- [6] D. U. Güneş, "Software industry in Turkey," *Turkish Software Industry Association (YASAD)*, http://yasad.org.tr/Content/UserFiles/YASAD_Presentation_ENG.pdf, 2010.
- [7] Turkish Software Industry Association (YASAD), "Software: the new strength of the economy (in Turkish: Yazılım: ekonominin yeni kalkınma gücü)," http://www.yasad.org.tr/Content/UserFiles/yasad_rapor.pdf, 2009.
- [8] J. S. Reel, "Critical success factors in software projects," *Software, IEEE*, vol. 16, pp. 18-23, 1999.
- [9] A. Ahimbisibwe, R. Y. Cavana, and U. Daellenbach, "A contingency fit model of critical success factors for software development projects," *Journal of Enterprise Information Management*, vol. 28, pp. 7-33, 2015.
- [10] H. Akkermans and K. van Helden, "Vicious and virtuous cycles in ERP implementation: a case study of interrelations between critical success factors," *Eur J Inf Syst*, vol. 11, pp. 35-46, 03/08/print 2002.
- [11] T. Chow and D.-B. Cao, "A survey study of critical success factors in agile software projects," *Journal of Systems and Software*, vol. 81, pp. 961-971, 6// 2008.
- [12] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, pp. 131-164, 2009.

- [13] K. M. Whitney and C. B. Daniels, "The Root Cause of Failure in Complex IT Projects: Complexity Itself," *Procedia Computer Science*, vol. 20, pp. 325-330, // 2013.
- [14] L. N. V. Heuvel, D. K. Lorenzo, and W. E. Hanson, *Root Cause Analysis Handbook: A Guide to Efficient and Effective Incident Investigation*: Rothstein Associates Inc, 2008.
- [15] Project Management Institute (PMI), *A Guide to the Project Management Body of Knowledge, 5th Ed.*, 5th Ed. ed., 2012.
- [16] S. A. Bohner and R. S. Arnold, 1996: IEEE Computer Society Press, Software change impact analysis.