Kaostan Düzene: Yazılım Projelerinde Hasar Kontrolü

Koray İNÇKİ

TÜBİTAK BİLGEM Bilişim Teknolojileri Enstitüsü (BTE)
P.K. 74, Gebze 41470 Kocaeli
korayincki@alumni.usc.edu

Özet: 1960'lı yılların sonundan bu yana yazılım projeleri yürütülmektedir. Yazılım mühendisliği alanında yapılan araştırmalar bu projelerin başarı ile tamamlanması için alınması gereken önlemleri anlatsa da günümüzde hala pek çok yazılım projesi başarısızlıkla sonuçlanmaktadır. Ayrıca, bu alanda yapılan araştırma sonuçları daha çok proje süreci başlamadan alınması gereken önlemleri belirtir. Biz bu bildiride BTE'de yürütülmüş DO-178B uyumluluğu gerektiren bir gömülü sistem yazılım projesinin başarısızlıktan (göreceli) başarıya dönüşümünde tecrübe edinilen deneyimleri paylaşacağız. Bildiride, örnek projenin başarı ile tamamlanmasında başarı ölçütlerinin yeniden tanımlanması, tüm paydaşların projeye destek olmasının sağlanması, yazılım süreçlerinin çeşitli aşamalarında uygulanan köklü ve zaman alıcı, fakat kaostan düzene geçişi kalıcı hale getiren, bir nevi hasar kontrolü yapmamızı sağlayan hem yazılım mühendisliği ve hem de proje yönetimi en iyi pratiklerimizi paylaşayacağız.

Anahtar Kelimeler: Yazılım Proje Yönetimi, Yazılım Süreçleri, Yazılım Kalite Güvence

1. Giriş

Yazılım Mühendisliği literatüründe yazılım projelerinin başarısızlığını önlemek için çeşitli çalışmalar yapılmaktadır. Paylaşılan çalışmaların büyük çoğunluğu önleyici tedbirler mahiyetindedir. Tüm bu araştırmalara rağmen yazılım projelerinde başarısızlıklar süregelmektedir. Standish Group tarafından 2013 yılında yayınlanan Chaos Manifesto raporunda 2012 yılında yürütülen yazılım projelerinin %39'u başarılıyken, %18'i kapatılmış, %43'ü ise önceden tahmin edilen kısıtların üzerinde tamamlanmıştır [6].

Yazılım geliştirmek masraflı ve çoğu zaman zor bir süreçtir. Yazılım geliştirme işinin başarılı tamamlanabilmesi için pek çok kılavuz ve standart tanımlanmış olmasına rağmen, proje sonrası değerlendirmelere pek rastlanmaz ve dolayısıyla geçmiş projelerden çok az ders çıkartırız [3]. Kılavuz ve standartların projelerde uygulanması öncelikle proje yöneticisinin görevi olmakla birlikte, tüm paydaşların bu en iyi pratiklerin uygulandığını gözlemlemesi ve denetlemesi gereklidir.

Başarısızlık faktörleri olarak pek çok etkenden söz etmek mümkündür [7]. Bu nedenleri i) Teknik, ii) Yönetimsel, ve iii) Sektörel olarak sınıflandırabiliriz. Proje paydaşlarının her üç (3) alanda da proje sonuçlarından etkilenmesi mümkün olduğu için, bu faktörlerde meydana gelen aksaklıklar proje ekibine baskı olarak yansır.

Başarısızlık faktörleri arasında proje ekibinin paydaşlardan kaynaklanan baskı nedeniyle yoğun stres altında çalışması önemli bir paya sahiptir [4].

Chaos raporunda projeler sonuçlarına göre Başarılı, Başarısız ve Sorunlu (Challenged) olarak sınıflandırılmıştır. Sorunlu projeler zaman, maliyet ve/veya kapsam açısından zorluklar yaşamış projelerdir. Yazılım mühendisliği alanında yapılan çalışmalar Başarısız ve Sorunlu projelerden kazanılan derslerin süreçsel olarak değerlendirmesi ile, müteakip projeler için bu projeler başlamadan önce önlemler alınması yönünde yoğunlaşır. Fakat, Sorunlu projelerin Başarısız olarak tamamlanmasını önlemek için yürümekte olan Sorunlu bir projede yapılması gerekenler bu bulgular ve tedbirlerden farklılık gösterebilir. Bu durumun yönetilmesi için tüm paydaşların (müşteri, fon sağlayıcı, sektörel uzmanlar, proje yürütücüsü) aynı bilgi seviyesi ve izleme yetenekleri ile projenin Başarısız olarak tamamlanmasını engelleyecek algı seviyesinde birlikte çalışmasını temin etmek gerekecektir.

Bu bildiride TÜBİTAK BİLGEM Bilişim Teknolojileri Enstitüsü'nde yürütülen, Türkiye'de alanında bir ilk olan gerçek-zamanlı bir sistemin Ar-Ge projesinde, Sorunlu aşamadaki bir projenin Başarısız olarak sonuçlanmaması sürecinde kazanılan yazılım proje yönetim tecrübesi paylaşılacaktır.

Bölüm 2'de projenin Sorunlu durumu tespiti ve iletişimi için yapılanlar anlatılacak olup, Bölüm 3'de durum tespitinin ardından alınan önlemlere değinilecektir. Bölüm 4'te proje yönetiminin bu aşamadaki rolünden bahsederken, Bölüm 5'te genel bir değerlendirme yapılacaktır.

2. Farkındalık Sorumluluk Getirir

Sorunlu duruma gelmiş projelerde zaman yetersizliğinden dolayı proje yöneticisi ve ekip üzerinde fazla mesai ile adeta nefes almadan çalışmaları için yoğun bir baskı vardır; durmaya zaman yoktur. Sorunlu projelerin Başarısızlık ile sonuçlanmasında bu bakış açısının önemli bir katkısı vardır. Bu tür durumlarda öncelikle proje yöneticisinin bir adım geri çekilerek projenin ve proje ekibinin genel resmine bakarak durum tespiti yapması, proje Başarısızlık ile sonuçlanmadan önce bir nevi hasar kontrolü yapabilmek için tüm paydaşlara kabiliyet kazandıracaktır.

(a) Kaostan Düzene İlk Adım

Proje planına göre belli dönemlerde projenin resmini çekmek genelde uygulanan bir tekniktir. Proje resmini çekerken genelde teknik ve takvimsel faktörler değerlendirmeye alınır. Biz, bu projede proje çalışanları ve müşteri tarafı dahil tüm paydaşlardan projenin mevcut durumunu incelemelerini istedik. Bu çalışmada, sadece teknik ve takvimsel faktörler değil bunların yanısıra proje yönetimi, proje ekibinin takım çalışması, çalışma ortamı ve geliştirme ortamı dahil çok çeşitli konularda paydaşların görüşlerini ve düzeltici önerilerini aldık.

Mevcut Durum Analizi (MDA) adını verdiğimiz bu aşamada paydaşlardan durum değerlendirmelerinin yanında düzeltici önerilerini de almamız tüm paydaşların projenin belirlenen hedeflere bağlılığını ve proje çıktılarında pay sahibi olma algılarını pekiştirmiştir. Böylece, projenin kalan süresinde tüm gelişmelerin paydaşlara iletişiminde kolaylık sağlanmıştır.

(b) **Hasar Kontrolü**

Yazılım projeleri başarısızlık faktörleri Chaos raporu yaklaşımından farklı olarak [4]'te incelenmiştir. Yaptığımız MDA raporunda bu faktörlerden (i) risklerin proje yaşam döngüsü süresince kontrol edilmesi, izlenmesi, (ii) personelin projede çalışmaktan hoşnutsuz olması, (iii) değişiklik kontrolünün etkin biçimde yönetilmemesi gibi etkenlerin proje başarısızlığında yüksek tesiri olduğu görülmektedir. 2.a'da anlattığımız MDA'da ortaya çıkan bulgular bize bu üç ana başlıkta aksaklıklar yaşandığını göstermiştir.

3. Yazılım Süreçleri İyileştirme ve Bilgiye Dayalı Yönetişim

Proje başlangıç aşamasında üzerinde yoğun çalışılan İş Kırınım Ağacı (İKA: Work Breakdown Structure) katı bir halde olmak zorunda değildir. İKA, proje planı ve takvimi hazırlanırken temel dayanak noktasıdır. Şu kabul edilen bir gerçektir ki proje planları yaşayan belgelerdir, projenin yaşam döngüsü içinde sürekli izleme ve kontroller ile gerektiğinde değişime açıktır [3]. İKA'lar da projenin ilerleyen safhalarında kapsam değişikliklerine göre özellikle Ar-Ge projelerinde değişiklik yaşayabilir. Özellikle proje takvimindeki kritik yol bu değişiklikten etkilenecektir.

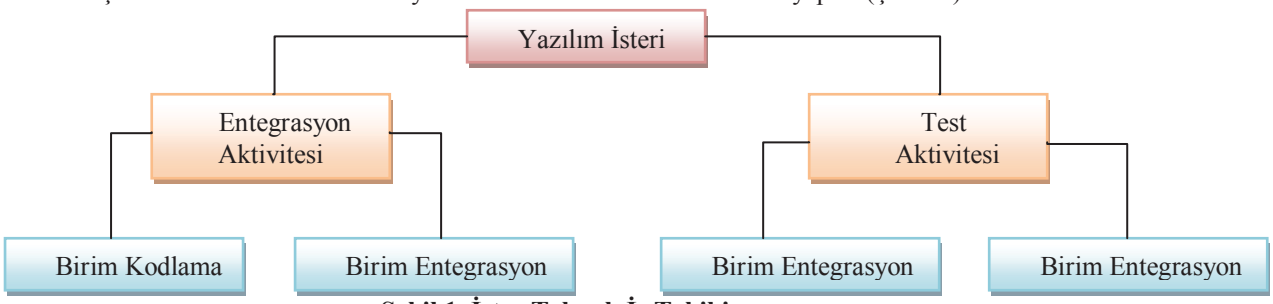
(a) İster Tabanlı İş Takibi

DO-178B Seviye-A yazılım kılavuzu gereği bu yazılım projesinde İster Yönetim Sürecinde müşteri isterlerinden test durumlarına kadar çift taraflı izlenebilirlik sağlanması gerekiyordu. İzlenebilirlik, kullanılan araçlarının uyumlu seçilmesi ile yönetilmiştir.

Tablo 1 Yazılım Süreç Araçları

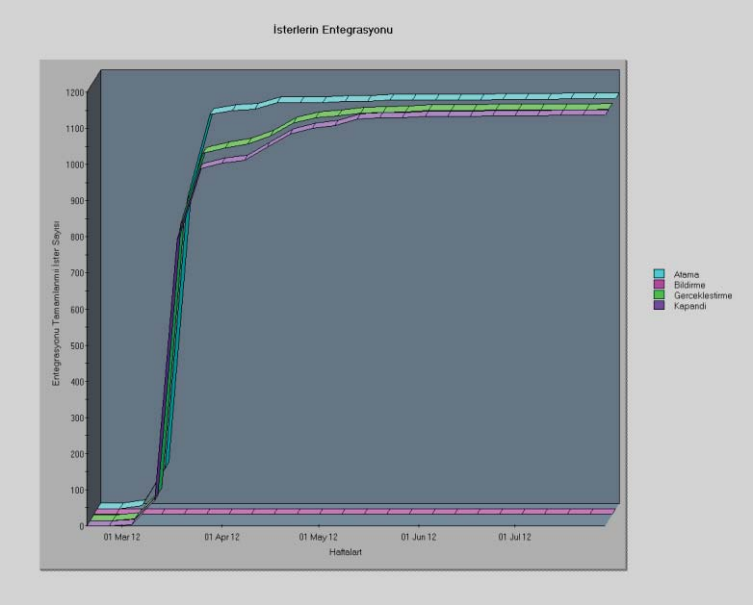
Yazılım Süreç Araçları	
Yazılım Modelleme Aracı	*Rational Rhapsody*
İster Yönetim Aracı	*Telelogic Doors*
Konfigurasyon Yönetim Aracı	*Rational Clearcase*
Test Durumları Tanımlama	*Telelogic Doors*
Aktivite/Hata Yönetim Aracı	*Rational Clearquest*

İster tabanlı yönetimini kolaylaştırmak için yazılım isterlerininin her birini Clearquest aracında "Aktivite„ olarak tanımlayıp, her istere bağlı alt aktiviteler oluşturarak tamamlanma oranını doğru biçimde takip ettik. Her isterin tamamlanması için iki temel durumu tanımlayarak bunların ister tabanlı takibini yaptık (Şekil-1).



Şekil.1. İster Tabanlı İş Takibi

İster tabanlı proje yönetimi sayesinde her personel projeye olan katkısını bireysel olarak takip edebilir duruma gelmiştir. Ayrıca, müşteriye projenin anlık ilerleme aşamasını Clearquest üzerinden aldığımız canlı raporlar ile en doğru ve etkin biçimde aktarabilmiş olduk (Şekil-2). Böylece müşterinin, kendisine sağlanan doğru bilgilere dayanarak ekibe güveni ve inancını artırmış olduk.



Şekil.2. İsterlerin Entegrasyon Durumu

(b) Ölçmeden Yönetemezsiniz

Yazılım Mühendisliğine yaptığı katkılardan dolayı 1999 yılında Stevens ödülüne layık görülen Tom DeMarco, [1]'de yer alan kitabında, "Ölçemediğin şeyi Kontrol Edemezsin!„ sözüyle yazılım projelerinde ölçmenin önemine değinmektedir. Yazılım projelerinde sıklıkla göz ardı edilen ölçme faaliyetleri bir proje yöneticisi ve diğer paydaşlar için projenin sağlıklıklı gidişatını izleyebilecekleri kalp atışlarını dinledikleri algılayıcılar gibidir. Bu süreçte, projenin tüm paydaşlarına projenin kalp atışları hakkında farkındalık oluşturmak ve sürekli güncel tutmak için gerekli ölçüm metriklerini aşağıdaki tabloda olduğu gibi tanımladık:

Tablo 2 Ölçüm Metrikleri

Ölçüm Metrikleri
1. Haftalık Açılan ve Kapatılan Hata Sayıları
2. Yazılım Geliştiricinin Üzerinde Açık Yazılım İsterleri
3. Yazılım Mühendisinin Üzerinde Açık Hata ve Düzeltmeler
4. Yazılım İsterlerinin Gerçekleme ve Doğrulama Oranı

Ölçüm Metrikleri
5. Yazılım Bileşenleri Entegrasyon Oranı
6. Sistem İsterlerinin Gerçekleme ve Doğrulama Oranı
7. Risk Durumları

(c) Araçlar Hayat Kurtarabilir

Proje ekibinin ortak bir dili konuşuyor olması grup etkileşiminde oldukça önemlidir. Ayrıca, proje sürecinde oluşturulan verilerin ortak bir havuzda toplanması, daha sonra bu verilerin derleme ve değerlendirmesi için büyük kolaylık sağlayacaktır.

Projenin tamamlanmasına kısa süre kalmasına rağmen proje ekibinin birlikte çalışabilirliğini ve dolayısıyla proje için belirlenen kalite kriterlerine erişilmesini temin etmek için yazılım mühendisliği geliştirme ortamını tekrar gözden geçirdik. Değerlendirmelerimiz sonucunda projenin tüm çalışanlar tarafından izlenebilirliğini artırmak için 3.b'de belirttiğimiz metrikleri her personele özgü sorgular olacak biçimde Rational Clearquest aracında tanımladık. Clearquest aracında üretilen "kişisel iş maddeleri,,, "kişisel hata durumları,,, "kişisel sorumlu olunan yazılım isterleri durumları,, sorgularının çıktıları, yazılım geliştirme ortamımız olan Eclipse'e entegre edilerek, proje yönetimi ve proje personeli arasında sürekli entegre bir etkileşim kurulmuş oldu. Proje yöneticisinin günlük ve haftalık ekip değerlendirmelerinde her personel özelinde sistemin tamamına yapılan katkının değerlendirmesini niceliksel bilgiye dayalı yapabilmesi, yöneticinin proje ve ekip üzerindeki kontrolünü artırmıştır.

4. Liderlik

Bölüm 2'de tüm paydaşların projenin mevcut durumu hakkında doğru ve zamanında bilgi sahibi olmaları, yani proje hakkında kendi ilgilendikleri çerçevede farkındalıklarının arttırılmasından bahsetmiştik. Proje yöneticisinin "Farkındalık Sorumluluk Getirir!,, yaklaşımını gerek proje ekibi gerekse projenin tamamlanmasına katkı sağlayacak ekip dışındaki paydaşlara benimseterek projenin Sorunlu da olsa tamamlanmasını temin etmesi için çaba göstermesi beklenir.

Kouzes ve Posner "Liderlik Mücadelesi,, [5] isimli kitaplarında kurumlarda sıradışı işler yapabilmek için bir liderin yapması gerekenleri anlatmaktadır. Sorunlu aşamaya gelen bir projede alışılagelmiş pratikler olumsuz sonuç doğurduğu farkındalığı oluşturmak için "Statükoya Karşı Gelmek,, (Challenge the Status Quo) ilk atılması gereken adımdır.

Kaos dönemlerinde proje ekibi belirsizliğe sürüklenen bir gemide gibi hisseder, dolayısıyla takım çalışması ilkeleri zedelenir. Bu durumu düzeltmek ortak hedef birliği oluşturmak (Inspiring a Shared Vision [5]) ile mümkün olacaktır. Bölüm 2'de sözünü ettiğimiz anket çalışmasında elde ettiğimiz kusurlardan birisi de projenin mevcut durumu ve ne zaman biteceğine dair belirsizliklerin olmasıydı. Belirsizlikleri en aza indirerek tüm personeli aynı ortak hedefe inanmasını sağladığınızda ekibiniz sizinle birlikte projeyi kurtarmak için özverili ile çalışacaktır [4,5].

Karmaşa durumlarında liderlik eden kişinin tek başına her alanda başarıya ulaşması mümkün değildir. Bu yüzden ekip arkadaşlarını da mücadeleye katılmalarını sağlayacak motivasyonlar üretmesi gerekir (Enabling Others to Act) [5]. Projenin

ilerlemiş aşaması olmasına rağmen insanların sorumluluk almasını sağlayacak alt ekiplerin yeniden şekillendirilmesi, personelin ufak başarılarını dahi topluluk karşısında taltif etmek, ürünün kalitesinin artımına neden olan faaliyetleri ekibin toplam kalite algısını artırıcı sembolik ödüllerle ödüllendirmek gibi detaylarla ekibimizin takım olarak toplam kaliteye katkılarını artırdık. Yazılımda bulunan hataların ekibin motivasyonunu olumsuz etkilememesi için bunları „İyileştirme Fırsatı" olarak lanse ederek hataların tespitini teşvik edici bir motivasyon oluşturduk. Çünkü, müşterinin kullanımına açılmadan önce üründe tespit edilen her hata, ürünümüz için bir iyileştirme fırsatıdır.

Proje yöneticisi ekibine nasıl çalışmaları gerektiğini en etkin, kendisi uygulayarak anlatabilir (Modeling the Way [5, 8]). Bu projede proje yönetimi olarak projenin ister tanımlama safhasından kalite gereksinimlerinin uygulanmasına, gözden geçirmelerden birim testlere kadar her aşamasında ekip ile birlikte çalışarak hangi adımda hangi kalitede sonuç beklediğimizi ekibimize anlattık. Böylece ekibin kendilerinden talep edilen disiplin ve titizlikte çalışma konusunda kolaştırıcı bir fayda elde ettik.

5. Sonuç

Bu bildiride TÜBİTAK BİLGEM BTE'de yürütülmüş olan, savunma sanayi alanında Türkiye için ilk olan özgün bir gerçek-zamanlı gömülü sistem yazılım projesinde edindiğimiz proje yönetim tecrübesini paylaştık. Bildiride aktarılan yöntemler uygulanarak, Başarısız olarak kapatılma aşamasına gelmiş bu projenin tüm paydaşların sorunların çözümüne katkısı sayesinde zaman aşımı ile de olsa Sorunlu sınıflandırmasında tamamlanmasını gerçekleştirdik. Bu sonucu elde ederken sadece teknik önlemler değil aynı zamanda paydaşların farkındalığını arttırarak ve daha çok katılımını temin edecek insani faktörleri de yoğun olarak kullandık. Bu yüzden, proje yönetiminde insan faktörünün yönetiminin etkisini de tecrübe ettik. Bu nedenle Başarısızlığa giden projelerin Sorunlu proje olarak tamamlanması için daha çok araştırma ve önleyici tedbirler geliştirilmesi gerektiğine inanıyoruz.

6. Referanslar

[1] DeMarco, T., *"Controlling Software Projects: Management, Measurement, and Estimation„*, Book, Prentice Hall PTR, 1986.

[2] *"DO-178B, Software Considerations in Airborne Systems and Equipment Certification„*, RTCA, 1992.

[3] *"Proje Yönetimi Bilgi Birikimi Kılavuzu (PMBOK Kılavuzu)„*, PMI-TR, 2013.

[4] Cerpa, N., Verner, J.M., *"Why Did Your Project Fail?„* Communications of the ACM, December 2009.

[5] Kouzes, J.M., Posner, B.Z., *"The Leadership Challenge„*, Josse-Bass Publishers, 1st Edition, California, 1990.

[6] *"The Chaos Manifesto 2013 – Thing Big Act Small„*, The Standish Group International Inc., 2013.

[7] *"The Chaos Report„* The Standish Group International Inc., 2007.

[8] Papke-Shields, K.E., Beise, C., Quan, J., *"Do project managers practice what they preach, and does it matter to project success?„*, Intl. Jrnl. of Prj. Mang., 2009.