

Yazılım Projelerindeki Başarı / Başarısızlık Sebepleri

İçerik ;

- ❖Yazılımın günümüzdeki yeri ve önemi.
- ❖Yazılım projeleri neden başarısız olur?
- ❖Başarılı yazılım proje yönetimi?
- ❖İstatistikler ne gösteriyor?
- ❖Sonuç.

Yazılımın Önemi

"Teknoloji tarihinde, hiçbir şey yazılımlar kadar çabuk insan hayatına girip, bu kadar etkili hale gelmemiştir."

Fishman (1997)

Yazılım her şeydir.

Yazılım her yerdedir.

Problemin Tanımı

Yazılım geliştirme yaratıcılık, mühendislik, akılcı yaklaşım ve liderlik gerektiren; tanımlı bir yaşam döngüsü içinde ilerleyen karmaşık ve yönetilmesi zor bir süreçtir.

Birçok yazılım projesi,

- *Hedeflenen süreden çok daha uzun sürmekte,*
- *Ayrılan bütçeyi aşmakta,*
- *Kullanıcı ihtiyaçlarını karşılamamaktadır.*

Yazılım Projeleri Neden Başarısız Olur?

IT uzmanlarının hemen hepsi, yazılım projelerinde meydana gelen hataların olması gerekenden çok daha fazla olduğunu kabul etmektedir.

Başarısızlık;

Vaktinde tamamlanmamış,
Ayrılan bütçeyi aşmış,
Kullanıcı gereksinimlerini karşılamamış proje.

Yazılım Projeleri Neden Başarısız Olur?

"Başarısızlık sebeplerinin hemen hepsi teknolojiden ziyade, yönetim ve personel kaynaklıdır."

Morgan, Soden (1973)

"Bazı projeler teknoloji veya dizayn problemleri sebebiyle başarısız olmuşsa da, asıl problemler proje yöneticilerinden kaynaklanmaktadır."

Keider (1984)

Yazılım Projeleri Neden Başarısız Olur?

Block 1983 yılındaki çalışmasında, başarısızlık kriterlerini sınıflandırılıp, sebeplerini belirtmiştir.

(22 yıl önce yapılan bu çalışma, günümüzde de oldukça sık kullanılmaktadır.)

"Başarısızlık faktörlerinin, birbirleri üzerinde güçlü etkileri bulunmaktadır. Bir alanda yapılan hata, diğer alanlarda da hataların oluşmasına sebep olmaktadır."

Lyytinen ve Hirschheim (1988)

Yazılım Projeleri Neden Başarısız Olur?

Gelişim süreçlerinin özensiz, yarım yamalak uygulanması.

Gerçekçi ve açık proje hedeflerinin belirlenmemesi.

Müşteri isteklerinin tam olarak anlaşılmadan geliştirme sürecine başlanması.

Yazılımın büyüklüğüne göre değil, yönetimin beklentilerine göre belirlenmiş gerçekçi olmayan son tarihler.

- Risk yönetiminde yapılan hatalar.
- Müşteri, geliştirici ve kullanıcılar arasındaki yetersiz iletişim.
- Olgunlaşmamış teknoloji kullanımı.
- Ticari baskılar.

Yazılım Projeleri Neden Başarısız Olur?

Yazılım sistemleri, doğası gereği, hata oluşumuna çok açık, kolay zarar görebilecek sistemlerdir.

"Telefon ağının çalışamaz hale gelip, 12 milyon üyenin servis dışı kalmasının sebebi: *Kod içindeki tek 1 satırda yer alan tek bir karakterin yanlış tipte tanımlanmasıdır.*"

AT&T (1991)

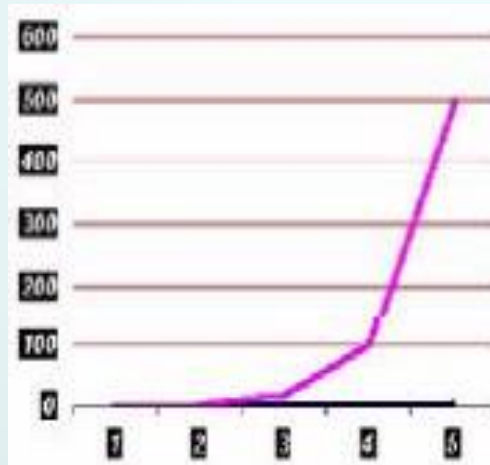
Yazılım Projeleri Neden Başarısız Olur?

Yazılım sürecindeki tek bir hata, zincirleme bir seriyle, tüm sürecin işleyişini bozabilir.

"Yazılım geliştirme, bir kazağın örülmesi gibidir." Unutulan eksik bir ilmik;

- Erken fark edilirse, 1-2 adım geriye dönülüp saniyeler içerisinde düzeltilebilir.
- Geç fark edilirse, atılmamış tek bir ilmik yüzünden tüm kazağın sökülüp, baştan örülmesi gerekebilir.

Yazılım Projeleri Neden Başarısız Olur?



X eksen: Proje Safhası
Y eksen: Hatayı Düzeltmenin Etkisi

- 1: Analiz
- 2: Tasarım
- 3: Kodlama
- 4: Test
- 5: Teslim Sonrası

Geliştirme sürecinde, bir hatayı düzeltmenin proje safhasına bağlı olarak etkisi.

Yazılım Projeleri Neden Başarısız Olur?

Yetersiz planlama.

Gereksinimlerin tam olarak belirlenememesi.

"Sydney Su Mirketi 33.2 milyon dolarlık harcama yaptıktan sonra, gelen çok sayıdaki değişiklik talebi sebebiyle, kullanıma yeni açılan projeyi iptal etmiştir."

(2002, Avustralya)

Yazılım Projeleri Neden Başarısız Olur?

Risk yönetimi:

Neler yanlış gidebilir?

Hatalar niçin oluşuyor?

Tehlikeleri, riskleri elemek ya da en azından etkilerini azaltmak için neler yapılabilir?

"Endüstriyel sektörler içindeki proje yönetim alanlarının hepsinde, en az uygulanan çalışma risk yönetimidir."

(Proje Yönetim Enstitüsü, Amerika)

YAZILIM PROJE YÖNETİMİ

Proje yönetimi, "sponsorun ve müşterilerin beklentilerinin karşılanması amacıyla, bilgi, yetenek, araç ve tekniklerin proje faaliyetlerine uygulanması" olarak tanımlanabilir.

Yazılım proje yönetimi açısından bakıldığında, bu tanım doğru ancak yazılım ürünlerinin doğal yapısı nedeniyle tam değildir.

Yazılım projeleri, diğer mühendislik dallarında yürütülen projelerden farklılıklar taşımaktadır.

Yazılım proje yönetimi neden farklıdır?

Mühendislik, programlama, tasarım, pazarlama gibi farklı bölümlerden insanların bir araya gelerek çalıştığı, belirsizliklerin fazla olduğu bir süreçtir.

Ürün odaklı değil, **süreç odaklı** yönetim kullanılır.
Çalışanlarının yaratıcılığına, ekonomik kısıtlara ve platform performansına bağlıdır.

Üretimde tekrar azdır.

Herhangi bir anda hemen her şey için değişiklik kararı alınabilir: planlar, çalışanlar, bütçe, kontrol noktaları, gereksinimler, dizaynlar ve testler.

Yazılım ürünleri için kullanılan metrikler ve ölçümler atomik olmadığı gibi oldukça sübjektiftir.

Yazılım Proje Yöneticisinin, Başarı Üzerine Etkisi

Yazılım projesini, tek başına başarısız kılacak yegane nokta proje yöneticisi tarafından alınmış kötü kararlardır.

Proje yöneticisinin görevleri:

- ✓ Proje planlama,
- ✓ Zaman, kaynak ve bütçe tahmini,
- ✓ Proje organizasyonu,
- ✓ Kontrat yönetimi,
- ✓ Kalite yönetimi,
- ✓ Risk yönetimi,
- ✓ Kurum içi ve dışı iletişimler,
- ✓ Çalışanların yönetimi.

Yazılım Proje Yöneticisinin, Başarı Üzerine Etkisi

Yazılım projelerinin başarılı olarak sonlandırmada en önemli oyuncu, hiç şüphe yok ki, tüm başarı kriterlerinin bağlı olduğu proje yöneticisidir.

Proje yöneticisi, “*yönetici*” vasfını ön plana çıkartarak, yönetsel faaliyetlerde özenli davranmak zorundadır.

Başarılı Yazılım Proje Yönetimi

Başarılı yazılım projelerinin yönetiminde etkili olan 5 temel faktör şu şekilde sıralanabilir:

- ✓ Doğru başlangıç yapılması.
- ✓ Momentumun oluşturulması ve korunması.
- ✓ Sürecin izlenmesi.
- ✓ Akıllı kararlar alınması.
- ✓ Proje sonu değerlendirmelerinin yapılması. (Post-Mortem Analizi)

Başlangıcın doğru yapılması

Proje doğru adımlarla başlanması, diğer başarı faktörlerinin durumu üzerinde önemli etkiye sahiptir.

"Başarısızlık faktörlerinin yaklaşık %70'nin, henüz projenin yapımına başlanmadan olduğu söylenebilir."

(Tom Field)

Başlangıcın doğru yapılması

Gerçekçi hedef ve beklentilerin tanımlanması

Yöneticiler, kullanıcılar, tasarımcılar ve geliştiricilerin hepsi gerçekçi beklentiler içinde olmalıdır.

Proje içindeki her bir üye, birlikte çalışmaya istekli olarak, takım halinde hareket etmeyi bilmelidir.

Başlangıcın doğru yapılması

Doğru takımın kurulması

Proje yöneticisi, en iyi tasarımcıların ve geliştiricilerin hepsine birden sahip olmak zorunda olmadığını bilmelidir.

Proje takımının yaklaşık %20'sinin üst düzey kalitede olması, oldukça iyi sonuçlar vermektedir.

Takım üyelerinin kendilerini geliştirebilecekleri bir ortam oluşturarak, takım içi dinamiklerden çok teknik ve yönetsel konular üzerine yoğunlaşabilecektir.

Başlangıcın doğru yapılması

İhtiyaç duyulan ortam ve araçların sağlanması

Proje başarısızlığına en direkt etkiyi, uygun seçilmemiş araçların yapacağı bir gerçektir.

- Araçtan kasıt yalnızca derleyici değildir.
- Analiz, dizayn, konfigürasyon, yönetim, test, yedekleme, doküman hazırlama ve problem giderme için de araçlara ihtiyaç duyulmaktadır.

Momentumun oluşturulması ve korunması

Başlangıçta momentumun oluşturulması kolaydır ancak süreç ilerledikçe momentumun korunması oldukça zor bir iştir.

Momentumun korunması için **3 temel** nokta üzerine yoğunlaşılmalıdır :

- **Sürtüşme/Yıpranma:** Düşük seviyede tutulmalıdır...
- **Kalite:** Geriye dönülüp, yazılıma kalite eklenmez...
- **Yönetim:** Çalışanlardan çok yazılım yönetilmelidir.

Her şey bittikten sonra, satışa sunulacak olan tek şey yazılımın kendisidir.

Sürecin İzlenmesi

Teknik anlamda, proje başarısı üzerine etki eden en önemli faktör süreç yönetiminin izlenmesidir.

Süreç yönetimindeki en büyük problem, planlanan zaman çizelgesinin neresinde bulunduğunun belirlenmesidir.

"Kavramsal bir modelin fiziki tezahürü ile, yani görülebilir, dokunulabilir, somut bir yapı ile sonuçlanır. (inşaat projesi)

“Kavramsal bir model ile başlar ancak ortaya çıkan ürün dokunulabilir, objektif yöntemlerle ölçülebilir, fiziksel varlığa sahip değildir. (yazılım projesi)

Akılcı Kararlar Almak

Akıllı kararlar verebilmek, başarılı proje yöneticilerini ortaya çıkaran en önemli unsurdur.

Akıllica kararlar almaktan kasıt "çözümü zor olan problemlere cevap bulmak" demek değildir.

Projenin, başlanılan platform üzerinde tamamlanabilmesi.

Teknik değerlendirilme ile müşteri gereksinimlerinin doğru ve tam olarak belirlenmesi.

Değerlendirme Çalışması (Post-Mortem Analizi)

Postmortem kelimesinin sözlük anlamı "otopsi"dir.

Proje yönetimi açısından, "bir olayın ya da sürecin sonrasında yapılan durum değerlendirme çalışması" olarak tanımlanabilir.

Post-mortem analizi ile başarı ve/veya başarısızlık üzerine etkide bulunmuş tüm faktörler belirlenebilir.

Değerlendirme Çalışması (Post-Mortem Analizi)

Şirketlerin kullandıkları yazılım metotları ve alışkanlıkları, yazılı olmasa da kendiliğinden gelişen bir sistem oluşturmaktadır.

Post-mortem analizi ile, oluşan bu sistemin olumlu yönde gelişmesi ve bir sonraki “üretim yarışında” daha faydalı olması sağlanabilir.

Sağladığı faydalara rağmen, şirketlerin çok azı, yaptıkları hatalardan dersler çıkararak, süreç yönetimi için kurumsallaşmaya gitmektedirler.

İstatistikler ne gösteriyor?

Başarısızlıklar, büyük bir oranda, uygulanmaya çalışılan yazılım proje yönetim sürecindeki temel noktalardan kaynaklanmaktadır.

2005 yılı içinde, yaklaşık 2000 ticari ve hükümet organizasyonun gönüllü olarak katıldıkları CMM değerlendirme çalışmalarının sonuçları aşağıda gösterilmektedir.

CMM 1,2: %53

CMM 3 : %30

CMM 4,5: %17

İstatistikler ne gösteriyor?

Yapılan istatistik temelli bir çalışmada, yazılım projelerinde kullanılan baskın proje yönetim çalışmalarının tespiti için, Avustralyalı yazılım geliştiricilerle birlikte çalışılmıştır.

Elde edilen istatistiksel sonuçlar ile yazılım projelerinin başarısı üzerine etkiyen faktörlerin belirlenmesi amaçlanmıştır.

Anket soruları hazırlanırken, proje yönetiminin 5 temel faktörü üzerinde yoğunlaşmıştır: Proje yöneticisi (M), gereksinim analizi (R), bütçe/zaman tahmini (C), risk değerlendirmesi (A) ve post-mortem analizi (P).

İstatistikler ne gösteriyor?

Çalışmanın sonuçları:

Aynı organizasyon tarafından yürütülen farklı yazılım projelerinde, tutarlı yöntemler kullanılmamaktadır.

Projelerin %50'si, gereksinim çalışması tamamlanmadan başlamaktadır. Risk değerlendirmesi, geliştirme sürecinin normal bir parçası olarak görülmemektedir.

Organizasyonlar, post mortem analizini zaman kaybı olarak gördüğünden, yaptıkları hatalardan gerekli dersleri almamaktadırlar.

İstatistikler ne gösteriyor?

"Hangi yazılım geliştirme metodolojisi kullanıldı?"

%20; Yaşam döngü metodolojisi tanımlanmamıştır.

%31; Waterfall veya türevi bir metodu kullanılmıştır.

%2 ; Prototip yaklaşımı kullanılmıştır.

%2 ; Spiral model kullanılmıştır.

%26; Soruya cevap verememiştir.

%10; "Yazılım geliştirme yaşam döngüsü" kavramının ne olduğunu bilmemektedir.

İstatistikler ne gösteriyor?

"Kullanılan yazılım geliştirme metodolojisi açıklar mısınız?"

GUI tabanlı,
Prosedürlere dayalı yöntem,
Gereksinimlerin belirlenmesi için basit bir metot,
Kullanıcılar ve sponsor ortaklıkları,
Kullanıcı işyerleri,
Yapılması gerekenlerin listelendiği bir “to do list”
gibi ilginç (!) cevaplar verilmiştir.

BAŞARILI BİR PROJENİN ORTAK YÖNLERİ

Harvard Business School tarafından yapılan ve yaklaşık 2 yıl süren bir araştırma [MacCormack01] sonucu başarılı yazılım projelerinin aşağıdaki 4 özelliğe sahip oldukları görülmüştür.

BAŞARILI BİR PROJENİN ORTAK YÖNLERİ

- 1) Başarılı projeler **yinelemeli** (iterative) şekilde yazılım geliştirirler. Yazılımın müşteri için anlamlı bir parçası erken bir yayımla (release) teslim edilir ve yazılım teslimi diğer yinelemeler ile devam eder. Müşteriden yayımlar sonrası sürekli geri beslenim alınır. Yazılım bir **evrim süreci** sonucunda oluşur.
- 2) Yapılan değişiklikler sonrası günlük tümleştirme (daily integration) yapılır. Tümleştirme sonucu yazılımın durumu hakkında **bağlanım (regression) testleri sayesinde hızlı bir şekilde geri beslenim** alınır.
- 3) Yazılım geliştirme ekibi **deneyimlidir**.
- 4) Projenin başından itibaren yazılım mimarisine ve sistemin **birbirinden bağımsız bileşenlerden** oluşturulmasına dikkat edilir.

BAŞARILI BİR PROJENİN ORTAK YÖNLERİ

Benzer bir araştırmada Bell Labs tarafından yapılmış ve aşağıdaki özellikler saptanmıştır.

- 1)Yinelemeli(iterative) yazılım geliştirme ve yinelemelerin sonunda müşterinin geri beslenimi.
- 2)Basit organizasyon yapısı ve rol tanımlarının gereksizce çoğaltılmaması.
- 3)Ekip içi iletişim.

BAŞARILI BİR PROJENİN ORTAK YÖNLERİ

- ☐ Bu özellikler yinelemeli yazılım süreçlerinin özelliklerine uyuyor.
- ☐ Özellikle çevik yazım süreçlerinin prensipleri ile birebir uygunluk sözkonusu.

BAŞARILI BİR PROJENİN ORTAK YÖNLERİ

İlk araştırma sonuçlarının birinci maddesinde yinelemeli şekilde yazılım geliştirmeden ve yazılımın çalışan bir parçasını erken teslim etmenin(early release) yararından bahsediliyor.

Yazılımda çıkan hatalara hangi faktörlerin etki ettiği konusunda yapılan başka bir araştırmada birinci maddeyi doğrular nitelikte.

Sonuçlara göre yazılımın %20 lik bölümünü erken teslim eden projenin ,yazılımın %40 'lık bölümünü daha geç teslim eden projeye oranla 10 faktör **az hata** içerdiği ortaya çıkıyor .

BAŞARILI BİR PROJENİN ORTAK YÖNLERİ

Yinelemeli(iterative) yazılım süreçlerinin başarılı olmasının nedenleri

- 1)Yinelemeli yazılım geliştirme süreçlerini kullanmak daha az risklidir.
- 2)**Riskler önceden tespit edilebilir ve çözülebilir.**
- 3)Proje süresince meydana gelebilecek değişikliklere çabuk tepki verebilir.
- 4)**Projenin durumu hakkında daha fazla bilgi sağlar ve tekrarlar arttıkça tahminler daha sağlıklı, kesin hale gelir.**
- 5)Hatalar daha çabuk bulunur, kalite seviyesi yüksektir.
- 6)**Sonuçta üretilen yazılım müşteri isteklerini daha iyi şekilde karşılar.**
- 7)Erken ve sürekli süreç iyileştirmesine olanak verir.
- 8)**İletişim ve koordinasyonu zorunlu kılar.**
- 9)“Gördüğüm zaman anlarım” anlayışına ters düşmez.
Müşteriler ne istediklerini anlamak için bazen yazılımı görmeleri gerekir.
- 10)**Tekrar kullanılabilirlik (reuse) için elverişli ortam yaratır.**
- 11)Proje yöneticileri yerinde taktik kararlar alabilirler, insan kaynağı daha yerinde kullanılır.
- 12)**Ekip üyeleri tekrarlar boyunca hatalarından dersler alır ve kendilerini geliştirir.**

BAŞARILI BİR PROJENİN ORTAK Y

Standish grubunun 1998 de 23000 projeyi kapsayan yaptığı bir araştırmada [Standish98] projelerin başarısının en çok aşağıdaki 5 faktöre bağlı olduğu sonucuna varıyor.

Faktör	Etki
1-) Kullanıcı/müşteri katılımı%30
2-) Üst Yönetici desteği%15
3-) Açık iş hedefleri(business objectives)%15
4-) Deneyimli proje yöneticisi%15
5-) Kısa aralıklı kilometre taşları%25

Sonuç

Bu çalışmada, tasarım/geliştirme metodolojisinden, programlama dilinden ve uygulama alanından bağımsız olarak, genel başarı kuralları aktarılmıştır.

Başarısızlık faktörleri, yazılım geliştirme sürecinin her zaman içinde yer almaya devam edecektir.

“Üzücü ve düşündürücü olan, günümüzde yaşanan başarısızlıkların büyük bir bölümünün önlenabilir sebeplerden kaynaklanmasıdır.”

Yakın gelecekte, geniş ölçekli IT projelerinin başarısızlığı yalnızca maddi anlamda büyük kayıplara sebep olmayacak, aynı zamanda insanların günlük yaşamları üzerine büyük etkilerde bulunabilecektir.

Bu nedenle, daha ciddi çalışmaların yapılması kaçınılmaz hale gelmektedir.

KAYNAKLAR

Yazılım Süreç İyileştirme Modelleri ve Proje Yönetimi

Meriç AYKOL; TBD - Sstanbul

Australian Software Development: What Software Project Management Practices Lead to Success?

J. M. Verner, N. Cerpa; (ASWEC'05)

Why software fails?

Charette, R.N., ITABHI Corp., USA; Spectrum, IEEE

Critical success factors in software projects.

Reel, J.S., Software, IEEE

Successful software management style: steering and balance

Royce, W., IBM Software Group, USA; Software, IEEE