

Yazılım Projelerinde Başarısızlık

Devrim Rehber

ODTÜ Enformatik Enstitüsü, Bilişim Sistemleri Bölümü, 06531, Ankara
drehber@ii.metu.edu.tr

Özet. İnsanlar, bilgi gereksinimlerini karşılamak ve hedeflerine ulaşmak için, itici güç olma potansiyelini keşfettiler yazılım sistemlerini, yaşamlarının her alanında kullanır olmuşlardır. Bu geniş kullanım, yazılım ağırlıklı sistemler ve çözmeye çalıştığı problemleri giderek karmaşıklaştırmaktadır. Bu sistemlerin, başarılı bir şekilde gereksinimlerinin belirtimi, tasarımı, geliştirilmesi, kurulum ve desteği zorlu bir entelektüel meydan okumadır. Sonuçta, yazılım uygulamalarının geliştirilmesi son derece dertli, pahalı ve hataya yatkın bir süreç olup en emek-yoğun çalışmaların başında gelmektedir. “Yazılım projelerinde başarısızlık”; planlı bütçenin veya zamanın aşılması, müşteri isteklerinin karşılanamaması, projenin iptali, ya da kullanımının birinci yılında düşük güvenilirlik ve kalite sorunları nedeniyle hizmet dışı bırakılmasıdır. Yazılım mühendisliği disiplini altında; çeşitli yazılım süreç modelleri, araç ve metodları geliştirilmesine rağmen, yüksek kaliteli yazılım sistemlerinin üretimi için gereksinim duyulan beceri düzeyi ve uygunluk seviyesine ulaşıldığını söyleyebilmek güçtür. Yazılım mühendisliğinin teknik cephesindeki ilerlemelere karşın, yönetim tarafındaki metodolojilerin iyileştirilmesi yeterli seviyede olmamıştır. Bu çalışmada, yazılım projelerindeki başarısızlık nedenleri; teknik, yönetsel ve sosyal başlıkları altında incelenecektir.

1 Giriş

1960’ların sonlarından bu yana, yazılım projelerinin, geliştirilen sistemlerin tümünün gecikmesine, planlı bütçeyi aşmasına, kullanıcı istekleriyle örtüşmemesine, kullanım, sürdürülebilirlik ve genişletilebilirlik sorunlarına yineleyen bir biçimde yol açması nedeniyle, “Yazılım Krizi” terimi kullanılmaktadır. Yazılımın elle tutulur olmaması, doğanın alışlagelen fizik kurallarıyla hükmedilmemesi, üretim biçiminin farklılığı gibi nedenlerle, günümüz toplumunda bile, birçok kişi, yönetici ve müşteri tarafından yazılıma hala kuşkuyla bakılmaktadır [1].

Bununla birlikte, insanlar işlerinde, kararlarında, eğlence, eğitim, finans, iletişim, sağlık, güvenlik, ulaşım, v.b., gibi yaşamın hemen her alanında yazılımlara güvenir olmuşlardır. Bu geniş kullanım, yazılım ağırlıklı sistemleri ve çözmeye çalıştığı problemleri de giderek karmaşıklaştırmaktadır. Akademisyenler, yöneticiler ve yazılım uzmanları anılan bu problemlere daha disiplinli ve yöntemsel bir yaklaşımın gerekli olduğu düşüncesiyle “yazılım mühendisliği” disiplinini geliştirmişlerdir. Teknik cephesindeki ilerlemelere karşın, hala gerekli uygunluk seviyesine ulaşamamıştır. Buna bağlı olarak, yüksek kaliteli yazılım sistemlerinin geliştirilmesi ve ilgili projelerinin yönetilmesi için gerekli olan becerilerde ustalık kazanılamamıştır [2], [3]. Kötü stratejik yönetim ve buna bağlı insan etkenleri başarısızlıkların en önemli nedenlerinden biri olarak gösterilmiştir. Anılan sorunların kaynağı araştırıldığında; politikalar, bütçeleme ve diğer dışsal kısıtlayıcılar gibi yazılım dışı konular yüzünden problemler yaşandığı görülmüştür [4]. Ancak, yazılım projelerinin başarılı olması isteniyorsa; yazılım mühendisliğiyle ilgisi olmayan bu konuların da yazılım uzmanlarınca bilinmesi ve yöneticilerce bu sorunlarla başa çıkma yöntemlerinin geliştirilmesi gerekmektedir. Çünkü kötü yönetim yazılım maliyetlerini diğer tüm etkenlerden daha hızlı bir şekilde artırmaktadır [5].

Sonuç olarak, modern dünyamızın en zorlu alanlarından biri olan yazılımda başarıyı başarısızlıktan ayıran üç temel etken mevcuttur. Bunlar; **teknik**, **yönetsel** ve **sosyal** nedenler olarak sınıflandırılabilir.

2 Başarısızlık Nedir?

Yazılım projesi

- Tamamlanmadan iptal edilirse,
- Tasarlanmış bütçesini aşarsa,
- Öngörülen tamamlanma süresinden daha uzun sürerse,
- Önceden belirlenen özellik ve işlevlerin daha azını karşılarsa

başarısız olmuş demektir. Yukarıdaki ilk maddenin gerçekleşmesi mutlak başarısızlık (*absolute failure*), diğerleri ise göreceli başarısızlık (*relative failure*) olarak da adlandırılır.

Yazılım dünyasınca küçük farklılıklarla birlikte benzer anlamlarda kullanılmakta olan diğer terimler şunlardır: Son derece sıkı programa sahip projeleri tanımlamak için Çatırtı Modu (*Crunch Mode*) deyimi kullanılır. Projede yer alanlar tarafından hissedilen baskıları dile getirir [6]. Ölüm Marşı (*Death March*) deyimi neredeyse olanaksız programa sahip projeler için kullanılır. Proje ekibini saran potansiyel başarısızlık hissini tanımlar [7]. Kaçak Proje (*Runaway Project*) deyimi ise iptal olmuş ya da iptale yakın projeleri tanımlamak için kullanılır [8], [9]. Tipik bir projeyi bu terimler cinsinden ifade etmek gerekirse: Yönetim, proje sonuçlarına ilişkin çok kısa bir süre için çok fazla vaatlerde bulunmuşsa, çatırtı moduna girilmiştir. Proje devam ederken, kısa bir süre sonra proje ekibi gerçekleşmesi imkansız bu

hedefe ulaşmak uğruna, kendisini ölüm marşında bulur. Projenin başarılı olamayacağı apaçık olduğunda proje artık kaçak bir projedir.

Başarı ve başarısızlık örüntüleri (*pattern*) cinsinden, yazılım dünyasını ilgilendiren en önemli unsurlar aşağıdadır [10]:

- Yazılım projelerinin öngörülen teslim tarihleri ve zamanında pazara yetiştirme düşüncesi veya kaygısı,
- Yazılım uygulamasını geliştirme maliyeti ve gereksinim duyulan kaynaklar,
- Yazılımın teslimindeki kalite ve güvenilirlik düzeyi,
- Yazılımın işleyişindeki öğrenme ve kullanım kolaylığı,
- Problem olduğunda müşteri desteği ve hizmet seviyesi,
- Uygulamanın olgunluk kazanması sırasında değişiklik ve sürdürülebilirlik kolaylıkları.

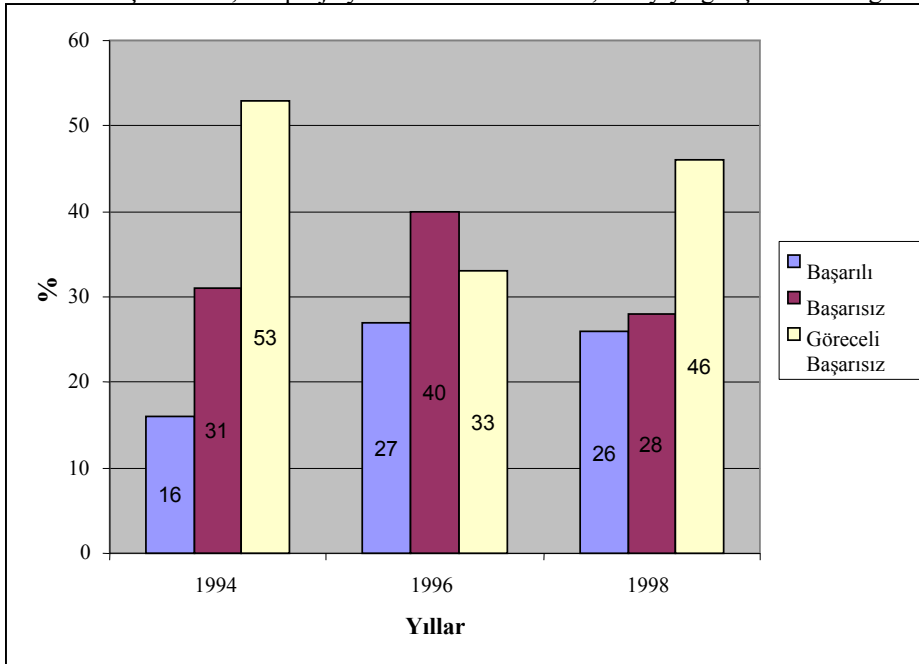
Bir projenin başarılı sayılması için yukarıdaki kıstasların tümünü sağlaması gerekmesine karşın, ilk üçünü yerine getirememesi başarısız olması için yeterlidir.

2.1 CHAOS Çalışması

Massachusetts, ABD’de konuşlu bir araştırma şirketi olan *The Standish Group*, 1993 yılından beri yazılım ve bilgi teknolojileri (BT) alanında meydana gelen başarısızlıkları inceleme çalışmasını sürdürmektedir. Anılan çalışma, yazılım proje başarısızlıklarının kapsamını, temel etkenlerini ve başa çıkma yollarını belirlemeye çalışmaktadır [11]. 1999 yılında yayımladıkları *CHAOS Chronicles* raporunda yer alan bazı sayısal değerler aşağıdadır:

- 23.000 uygulama incelemeye alınmıştır.
- Başarısız projelerin maliyeti 75 Milyar ABD \$’ıdır.
- Yazılım projelerinin sadece %26’sı başarılı bir şekilde tamamlanmıştır.
- Bütçesini veya planlı zamanını aşan (*challenged*) proje oranı %46’dır.
- İptal edilen proje oranı %28’dir.
- Toplam maliyet aşımı 22 Milyar ABD \$’ıdır.

CHAOS araştırmaları, BT proje yönetiminde az da olsa, bir iyiyeye gidişin var olduğunu göstermektedir (Şekil-1).



Şekil 1. Proje başarı oranları

3 Yazılım Felaketlerinin Temel Nedenleri

Araştırmanın izleyen bölümlerinde, yazılım projelerindeki başarısızlık nedenleri; teknik, yönetsel ve sosyal başlıkları altında incelenecektir.

3.1 Teknik Nedenler

Yazılım projelerinin başarısızlığına neden olan pekçok etken bulunmaktadır. Bu etkenler arasında, yazılımın doğasından kaynaklanan ve yazılımın büyüklüğü ile doğrudan ilintili karmaşıklık (*complexity*) ön plana çıkmaktadır. Tüm yazılım

alt endüstrilerinde, büyük sistemlerin iptal edilme ya da gecikmiş bir şekilde tamamlanabilme olasılıkları küçük uygulamalara oranla daha büyüktür. Tablo-1, projelerin zaman planlamaları açısından sonuçlanma olasılıklarını göstermektedir [10], [11].

Tablo 1. Projelerin zaman planlamaları açısından sonuçlanma olasılıkları

Function Points	Erken	Zamanında	Gecikmiş	İptal Edilmiş
1 FP	14.68%	83.16%	1.92%	0.25%
10 FP	11.08%	81.25%	5.67%	2.00%
100 FP	6.06%	74.77%	11.83%	7.33%
1,000 FP	1.24%	60.76%	17.67%	20.33%
10,000 FP	0.14%	28.03%	23.83%	48.00%
100,000 FP	0.00%	13.67%	21.33%	65.00%
<i>Ortalama</i>	5.53%	56.94%	13.71%	23.82%

Yazılım projelerinin başarısızlığında sıkça karşılaşılan teknik nedenler aşağıda sıralanmıştır:

- Etkili olmayan yazılım teknolojilerinin kullanılması
- Uygun olmayan yazılım araçlarının seçimi
- İş süreçlerinin belirlenmemesi
- Çözümlemenin (*analysis*) uygun bir şekilde gerçekleştirilmemesi
- Geçmişe yönelik yazılım ölçüm verilerinin bulunmaması
- Etkin bir mimarının seçilmemesi
- Etkin geliştirme yöntemlerinin kullanılmaması
- Tasarım gözden geçirmelerinin (*design review*) yapılmaması
- Kod denetimlerinin (*code inspection*) yapılmaması
- Uygun olmayan, disiplinsiz sına (testing) yöntemlerinin uygulanması
- Belirtim (*specification*) ve tasarımın el ile yapılması
- Düzenleşim (*configuration*) denetiminin uygulanmaması
- Kullanıcı gereksinimlerinin belirsizliği veya %30'dan daha fazla değişmesi
- Uygun programlama dillerinin kullanılmaması
- Aşırı ve ölçülmemiş karmaşıklık düzeyi
- Yeniden kullanılabilir bileşenlerin kullanılmaması
- Uygun veritabanı planlamalarının ve tasarımının yapılmaması
- Yeni teknolojileri kullanmaya geçişin birdenbire olması, deneyim eksikliği

3.2 Yönetimsel Nedenler

Yazılım mühendisliğinin teknik alanlarında pek çok süreç modeli, araç ve yöntemin geliştirilmiş olmasına karşın, yönetim alanında aynı düzeye erişildiği söylenemez. Yazılım projeleri sadece kötü hatalar, kötü şans, kötü programcılar ya da kötü test personeli yüzünden değil çoğunlukla yönetimsel nedenlerden başarısızlığa uğramaktadır [11]. Üst yönetimin genellikle yazılım konularına uzak oluşu ve kendi uzmanlık alanlarında göstermiş oldukları yetkeyi (otorite) bu alanda yeteri kadar uygulayamamaları yönetimsel sorunların başında gelmektedir. Başarılı yazılım projeleri ile başarısız projeler arasındaki en önemli fark iki sözcük ile özetlenebilir: "Sürpriz yok!" Başarılı projelerde tahmin, planlama ve kalite kontrol tüm proje süresi boyunca gözönünde bulundurulur. Bu nedenle, beklenmedik gecikmelere rastlanmaz. Yazılım projelerini başarısızlığa iten diğer yönetimsel nedenlerden başlıcaları aşağıdadır:

- Üst yönetim, pazarlama bölümü veya proje yöneticisi tarafından saflıkla verilen sözler ve diğer politik baskılar
- Gerçekçi olmayan kestirimlere bağlı olarak çok yoğun bir zaman baskısı
- Büyüklük, işgücü ve maliyet kestirimlerinin üst yönetimde geri çevrilmesi
- Üst yönetimin bilgisizliği ve ilgisizliği
- Proje yönetimi uygulamalarındaki yanlışlıklar
- Proje amacının ve kapsamının açık bir şekilde belirlenmemesi
- Proje yöneticisinin yetersizliği
- Nitelsiz teknik personelin çalıştırılması
- Etkin olmayan geliştirme süreçlerinin bulunması
- Geliştirme ekibinde deneyimli personelin yer almaması

- Otomatik kestirim araçlarının kullanılmaması
- Otomatik planlama araçlarının kullanılmaması
- Otomatik proje gelişim izleme yöntemlerinin kullanılmaması
- Etkin olmayan proje değişim yönetimi usullerinin kullanılması
- Kalite kontrol yaklaşımlarının az ya da hiç kullanılmaması
- İşe uygun personel, personele uygun iş ölçütünün gözardı edilmesi
- Yapılan yazılım proje yönetim planına uyulmaması
- Uygun olmayan metriklerin toplanması veya hiç toplanmaması
- Bazı kritik görevlerde (kalite güvence, sınama, planlama, kestirim, veritabanı yöneticisi, tasarım, değişim kontrolü, hata ayıklama, v.b.) uzmanlar yerine sıradan teknik personelin kullanılması
- Başarısızlık uyarı imlemlerinin bulunmaması ya da geç fark edilmesi
- Belgelemenin (*documentation*) uygun bir şekilde yapılmaması
- Geliştirme ekibinin iş süreçleri ve sorun sahası üzerine bilgilendirilmemeleri
- Pazarın küreselleşmesinden kaynaklanan yoğun çekişme ortamı
- Yeni teknolojilerin üretilmesinden kaynaklanan çekişme ortamı
- Yazılım/donanım sağlayıcılarının zayıf performans sergilemeleri
- Beklenmedik ve öngörülemeyen bunalımlar

3.3 Sosyal Nedenler

Yazılım projelerinin başarısızlık örüntüleri arasında bir de kültürel ve sosyal etkenler bulunmaktadır. Hem alıcı hem de sağlayıcı açısından, kurumun kültürü ve sosyal yapısı projenin başarısına doğrudan etki yapmaktadır. Müşteri tarafı genellikle yoğun bir zaman baskısına yönelir. Kuruluşların kültür ve yapılarına bağlı olarak, üst yönetimin projeye yerinde olmayan tavsiyeleri, çoğunlukla projelerin başarısızlığında etkindir. Bilişim projelerinin çoğu beraberinde bazı köklü değişiklikleri de getirir. Ancak, müşterinin bu değişime tepkisi genellikle olumsuz yönde gerçekleşir. Bunda; alışlagelmiş yöntemlerin dışına çıkmanın getireceği tedirginlik, mevcut sistemde ellerinde bulundurdukları güçten ödün verme zorunda kalma duygusu ve işini yitirme korkusu büyük rol oynamaktadır. Buna bağlı olarak, yeni sistemin geliştirilmesi için yürütülen projeye destek olmak yerine, açıkça veya dolaylı bir biçimde karşı çıktığı da çeşitli projelerde gözlemlenmiştir [7], [8], [9], [10], [12], [13], [14], [15], [16]. Başarısız projelerde sıklıkla görülen sosyal nedenlerin en önemlileri şunlardır:

- Takım içi iletişim ortamının zayıf olması
- Geliştirici personelin çalışma ortamının uygunsuz olması
- Alıcı ile zayıf bağlantı kurulması sonucu girdilerinin alınmaması
- Alıcıların gerçekçi olmayan beklentileri
- Alıcı ve diğer paydaşlar arasında şiddetli sürtüşmelerin yaşanması
- Deneyimsizlikten kaynaklanan iyimserlik
- Deneyimsiz yönetim ekibi
- Deneyimsiz teknik ekip
- Deneyimsiz müşteri ile çalışılması
- Teknolojik gelişmelerden habersiz olunması
- Şirket politikalarında ayrımcı uygulamaların bulunması
- Yönetimin soğukluğundan kaynaklanan mesafeli iletişim
- Zayıf rapor verme yapısı
- Teknik personelin sürekli zaman baskısı altında çalışması
- Bölüm ve birimler arası iletişim ve desteğin olmaması
- Karmaşık bağımlılıkların mevcut olması
- Hırsa bağlı yeni yeni hedeflerin ortaya konması
- İşsiz kalma korkusu
- Öç alma duygusu ve düşmanca tavır

3.4 Diğer Nedenler

Yukarıda belirtilen nedenlerin dışında, sıklıkla bütün projelerde görülmeyen, ancak projelerin başarısızlığına neden olabilecek diğer etkenler de aşağıda sıralanmıştır:

- Coğrafi olarak birbirinden uzak ve birden fazla şehir/ülke için geliştirilecek projeler
- Proje sözleşmesinin dağıtık ve birden fazla alt-yüklenicinin sorumlu olduğu projeler
- Hem yazılım hem de donanımın eşzamanlı olarak birlikte geliştirildiği hibrid projeler
- Gerçek zamanlılık kısıtlayıcısının en yoğun olduğu projeler (uzay, havacılık, güdümlü mermi, v.b.)
- Yasal kısıtlayıcılar altında geliştirilen projeler

- İhaleyi alma kriterinin en düşük fiyat teklifi olduğu sözleşmeli projeler
- Parasal olarak güçsüz, riskli fona sahip şirketlerin projeleri
- Köklü personel indirimine giden veya kilit personelini işten çıkaran şirketlerin yönettiği projeler
- Personel aşınma oranı %40'ı aşan projeler
- Personel artışı ayda %10'dan fazla olan projeler
- Teknik personelin genel becerilerinin dışındaki alanlarda yürütülen projeler
- Yönetim ekibinin genel deneyimlerinin dışındaki alanlarda yürütülen projeler
- Teknik personel değişim hızının yüksek olduğu projeler

4 Sonuç ve Öneriler

Yazılım projeleri doğasından kaynaklanan bir karmaşıklığa sahiptir ve karmaşık projeler de dikkatli bir **planlama** gerektirir. İyi planlanmış projeler etkinlikle denetlenebilir, gelişimi izlenebilir ve çalışanlarına gereksinim duyacakları desteği sağlar. Planlamada açık ve belirgin amaçlar, kararlı, tam ve belirgin gereksinimler, doğru kestirimler başarı için anahtardır. Yazılım projeleri doğası gereği risklidir ve etkin bir **risk yönetimi** olmaksızın başarılı olamazlar. **Kullanıcıların** projede erken ve sürekli bir biçimde **yer almaları** risklerin karşılanmasında kilit rol oynar. Proje, zaman, bütçe ve diğer hedeflere erişim yönünden **denetlenmelidir**. Bu denetimin gerçekleştirilebilmesi; projenin gerçek durumunun belirlenebilmesi olarak tanımlanan **görünürlük** yardımıyla olur. Belirlenmiş metrikler, geçmişteki verilerle karşılaştırmalı **ölçümler** ve kilometre taşlarına göre ilerleme durumunun gözlenmesi görünürlüğü sağlayan yardımcılarıdır. Doğru mimarinin, geliştirme araç, dil ve yöntemlerinin seçimi, tasarım gözden geçirmelerinin ve kod denetimlerinin uygulanması, uygun test yöntemlerinin kullanımı, düzenleme denetiminin araç yardımıyla yapılması ve etkin belgeleme başarı etkenleridir. **Yahınlık** tüm geliştirme evrelerinde vurgulanmalıdır. Yazılım geliştirme yaratıcılık, akıl, girişimcilik, ısrarcılık ve güdülenme (motivasyon) gerektirir. Bu özelliklerin tümü **insan odaklıdır**. Eğitim, moral, takım iletişimi ve çalışma ortamının da önemi çok büyüktür. Proje yöneticisi ve teknik ekibin **yetenekli ve deneyimli** olduğu, kritik görevlerde uzman personelin yer aldığı ve tüm bu sayılanları **kalite** şemsiyesi altında gerçekleştirebilen proje gruplarının başarı şansı daha yüksektir.

Yazılım projelerinde başarısızlığın nedenlerinin anlaşılması ve bunlardan dersler çıkarılması, gelecekteki projelerin başarısını etkiler. Biten her projenin ardından nelerin iyi, nelerin kötü gittiğinin çözümlenmesinin yapılacağı toplantı ve çalışmalar (*post-mortem*), projede kullanılan ölçün (*standard*), yöntem, ölçüm ve verilerin kaydedilerek geleceğe ışık tutacak bilgilere dönüştürülmesi ve bu bilginin kurumsallaştırılması için en iyi fırsattır.

Kaynakça

1. Dorfman, M. ve Thayer, R. H., Software Engineering, IEEE Computer Society Press, Los Alamitos, CA, 1996, ISBN: 0818676094.
2. Pressman, R. S., Software Engineering, A Practitioner's Approach, 4th Ed., McGraw-Hill, 1997, ISBN: 0071146032.
3. Abdel-Hamid, Tarek K. ve Madnick, Stuart E., Software Project Dynamics: An Integrated Approach, Prentice-Hall, New Jersey, NJ, 1991, ISBN: 0138220409.
4. Thomsett, R., People Project Management, Yourdon Press, Inc., New York, NY, 1980, ASIN: 0136557473.
5. Boehm, B., Software Engineering Economics, Prentice-Hall, New Jersey, NJ, 1981, ISBN: 0138221227.
6. Boddie, John, Crunch Mode: Building Effective Systems on a Tight Schedule, Yourdon Press, New York, NY, 1987, ASIN: 0131949608.
7. Yourdon, Ed, Death March: The Complete Software Developer's Guide to Surviving 'Mission Impossible' Projects, Prentice-Hall, Englewood Cliffs, NJ, 1997, ISBN: 0130146595.
8. Glass, R. L., Software Runaways, Prentice-Hall PTR, Upper Saddle River, NJ, 1998, ISBN: 013673443X.
9. Cole, A., "Runaway Projects-Cause and Effects," Software World (UK), Vol.26, No.3, 1995, s. 3-5.
10. Jones, Capers, Patterns of Software System Failure and Success, International Thomson Computer Press, Boston, MA, 1996, ISBN: 1850328048.
11. Jones, Capers, "Project Management Tools and Software Failures and Successes," CrossTalk, July 1998, s. 13-17, (<http://www.stsc.hill.af.mil./crosstalk/1998/07/tools.pdf>), 2003
12. Weinberg, G. M., Quality Software Management, Vol.1, Systems Thinking, Dorset House Publishing, 1997, ISBN: 0932633226.
13. The Standish Group International, Inc, "CHAOS: A Recipe for Success", <http://www.standishgroup.com>, 2003.
14. Flowers, S., Software Failure: Management Failure, Amazing Stories and Cautionary Tales, Wiley, 1996, ISBN: 0471951137.
15. Brooks Jr., F. P., The Mythical Man-Month (20th Anniversary Edition), Addison-Wesley, MA, 1995, ISBN: 0201835959.
16. DeMarco, T. ve Lister, T. R., Peopleware: Productive Projects and Teams, 2nd Ed., Dorset House Publishing, NY, 1999, ISBN: 0932633439.

17. May, L. J., "Major Causes of Software Project Failures," CrossTalk, July 1998, s. 9-12, (<http://www.stsc.hill.af.mil./crosstalk/1998/07/causes.pdf>), 2003.