

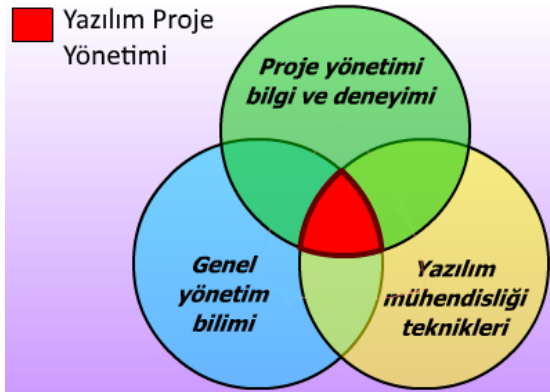
BÖLÜM 2. YAZILIM PROJE YÖNETİMİ

1

2.1. YAZILIM PROJE YÖNETİMİ BİLEŞENLERİ

2.1.0. GENEL BİLGİLER

- Yazılım proje yönetimi; yazılım mühendisliği teknikleri, genel yönetim bilimi ile proje yönetimi bilgi ve deneyiminin ara kesitidir.



2

2.1. YAZILIM PROJE YÖNETİMİ BİLEŞENLERİ

2.1.0. GENEL BİLGİLER

- Yazılım projesinin büyüklüğüne göre, uygulanacak yönetim ve denetiminin düzeyi, kullanılacak araç tipleri ve teknoloji gereği farklılık göstermektedir.
 - Küçük ölçekli projeler:
 - Çok küçük: 1-2 kişi içeren çok kısa süreli (1-2 ay) projeler
 - Küçük: 2-3 kişi içeren kısa süreli (2-3 ay) projeler
 - Basit bir kullanıcı uygulama programı tekniği ile düzenlenebilmektedir.
 - Büyük ölçekli projeler:
 - 5-20 kişilik bir ekip tarafından, 2-3 yılda tamamlanabilmektedir.
 - Çok sayıda alt sistemden oluşmaktadır.
 - Başka sistemlerle tümleştirme gereksinimleri mevcuttur.
 - İş planlama ve kod yönetim uygulamalarının kullanılması zorunludur.

3

2.2. YAZILIM ÖLÇÜMÜ

2.2.1. ÖLÇÜM BİÇİMLERİ

- İki temel ölçüm biçimi vardır:
 - Doğrudan (Ör. gider, süre, komut satırı, hız, bellek genişliği, hata miktarı)
 - Dolaylı (Ör. işlev, nitelik, karmaşıklık, etkinlik, güvenilirlik, dayanıklılık)
- Ortak amaç; verimlilik, kalite, gider ve belgelemenin hesaplanmasıdır
- Doğrudan ölçülen niceliklerden tanımlanan bazı ölçütler:
 - LOC = Line of Code (Kod satırı sayısı)
 - KLOC = 1000 * LOC (K:Kilo)
 - Verimlilik = KLOC/(kişi * ay)
 - Kalite = Hata/KLOC
 - Gider = Toplam gider/KLOC
 - Belgeleme = Belge sayfası/KLOC

4

2.2. YAZILIM ÖLÇÜMÜ

2.2.1. ÖLÇÜM BİÇİMLERİ

- Bir dolaylı ölçüm örneği: Fonksiyon noktası ölçümü (FP: Function Point)
 - Tablo 2.1, 2.2 ve ardındaki eşitlikler kullanılacaktır:

Tablo 2.1. Fonksiyon noktası bileşenleri

i	Nicelik	Sayısı (S)	Ağırlık Faktörü (AF)			F _{Pi}
			Basit	Orta	Karmaşık	
1	Kullanıcının yazılıma giriş sayısı (user input)		3	4	6	S * AF
2	Kullanıcının aldığı çıktı sayısı (output)		4	5	7	
3	Kullanıcının sorgulama sayısı (query)		3	4	6	
4	Kütük sayısı (record)		7	10	15	
5	Dış arabirim sayısı		5	7	10	
					ΣF _{Pi} :	Hesapla

5

2.2. YAZILIM ÖLÇÜMÜ

2.2.1. ÖLÇÜM BİÇİMLERİ

Tablo 2.2. Yazılım parametreleri

i	Parametre	Fi (0-5 arası)
1	Güvenli yedekleme ve geri yükleme gerekli mi?	
2	İletişim altyapısı gerekli mi?	
3	Dağıtılmış işleme fonksiyonları var mı?	
4	Performans kritik mi?	
5	Sistem yükü fazla mı?	
6	Çevrimiçi veri girişi var mı?	
7	Çok ekranlı hareket girişleri var mı?	
8	Ana dosyalar çevrimiçi güncelleniyor mu?	
9	Giriş, Çıkış ve Sorgular karmaşık mı?	
10	İçsel işlemler karmaşık mı?	
11	Yeniden kullanılabilirlik var mı?	
12	Yükleme tasarıma dahil mi?	
13	Farklı şirketlerde de çalışması söz konusu mu?	
14	Uygulama kullanıcı tarafından kolayca değiştirilebilir mi?	
		ΣFi: Hesapla

6

2.2. YAZILIM ÖLÇÜMÜ

2.2.1. ÖLÇÜM BİÇİMLERİ

- FP hesabı:
 - Tablo 2.1 her satırı için $FP_i = \text{Sayısı} * \text{Ağırlık Faktörü}$
 - Sayı toplamı: $\sum FP_i$ hesaplanır.
 - Yazılımın basit-orta-karmaşık oluşu tahmin yoluyla kestirilmektedir
 - Tablo 2.2 yazılımın 14 özelliğine göre "karmaşıklık düzeltme değeri" $\sum F_i$ hesaplanır
 - Her özellik 0 (Uygulanamaz/geçersiz) ile 5 (mutlaka gerekli) arasında değerlendirilerek toplama işlemi yapılır
 - $FP = \sum FP_i (0.65 + 0.01 \sum F_i)$
- FP hesabı üzerinden tanımlanan bazı ölçütler:
 - Verimlilik = $FP / (\text{kişi} * \text{ay})$
 - Kalite = Hata / FP
 - Gider = $\text{Toplam gider} / FP$
 - Belgeleme = $\text{Belge sayfası} / FP$

7

2.2. YAZILIM ÖLÇÜMÜ

2.2.2. ÖLÇÜM KIYASLAMASI

- Eleştiriler:

LOC ÖLÇEĞİ	FP ÖLÇEĞİ
– Programlama diline bağımlı (kısmen)	+ Programlama dilinden bağımsız
– İyi tasarlanmış ama kısa yazılımları yeterince değerlendirememekte	– Sübjektiftir
– İşlemsel olmayan dillerdeki yazılımlara kolayca uyarlanamaz	– Doğrudan fiziksel bir ölçüt değildir
	– Veri toplaması güçtür

8

2.3. YAZILIM PROJE MALİYET TAHMİN YÖNTEMLERİ

2.3.0. GENEL BİLGİLER

- Yazılım geliştirme planı aşamasında proje hesaplarını yapabilmek için, daha önce tamamlanmış projelerin kesin hesaplarını örnek almak gerekmektedir.
- Planlama aşamasında yapılan proje tahminleri %100 tutarlı ve güvenli olamamaktadır.
- Buna karşın yine de, eski bilgi ve deneyim sonuçlarına dayanarak, bir tahminde bulunmak gerekmektedir.
- Bu amaçla geliştirilen çeşitli tahmin yöntemlerinden bazıları bu bölümde incelenecektir.

9

2.3. YAZILIM PROJE MALİYET TAHMİN YÖNTEMLERİ

2.3.1. BİLİRKİŞİ TAKDİRİ

- Bilirkişi takdiri; daha önce tamamlanmış ve maliyeti ile bitirme süresi bilinen projeleri karşılaştırmak yoluyla yapılmaktadır.
- Karşılaştırmada eski ve yeni projenin maliyet kalemleri arasında görülen farklar yüzde olarak belirtilmektedir.
- Bu farklılıklara göre, toptan bir gider ve süre tahmini yapılmaktadır.
- Tutarlı bir tahmin yapılabilmesi için, bilirkişilerin deneyim sahibi olmaları ve iki proje arasındaki farklılıkları gerçekçi olarak saptamaları gerekmektedir.

10

2.3. YAZILIM PROJE MALİYET TAHMİN YÖNTEMLERİ

2.3.2. DELPHİ YÖNTEMİ

- Bir koordinatör yönetiminde bilirkişi ekibi tarafından birkaç kez yinelenen yazılı takdir miktarlarına göre ortak bir tahmin yapılmaktadır.
- Bu yöntem aslında sosyal bilimlerde kullanılan bir anket şeklinin proje takdiri için uyarlanmış halidir.
- Yöntem:
 - Koordinatör bilirkişilere "sistemi tanımlama" belgelerini ve birer "tahmin formu" verir.
 - Bilirkişiler birbirinden habersiz, nedenleri ile birlikte tahminlerini yazılı olarak koordinatöre bildirir.
 - Koordinatör ortanca ve aykırı tahmin sonuçlarını bilirkişilere vererek, yeni tahminde bulunmalarını ister.
 - Ortak bir değere yaklaşıncaya kadar bu işlem yinelenmektedir.

11

2.3. YAZILIM PROJE MALİYET TAHMİN YÖNTEMLERİ

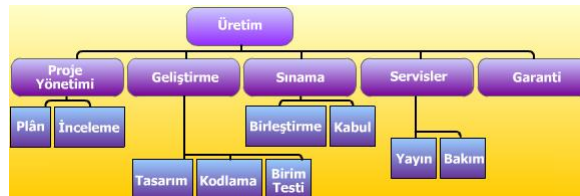
2.3.3. ANALİZ YÖNTEMİ

- Analiz yöntemi; ürün veya işlemi hiyerarşik olarak öğelerine ayırarak, en alt öğelerden başlayıp yukarı doğru her öğe için gider takdir etmeye dayanır.
- Bu amaçla önce, ürüne ve/veya üretime dayalı analiz kartı düzenlenir.



Ürün analiz kartı

Üretim analiz kartı



12