



ANKARA ÜNİVERSİTESİ ENFORMATİK BÖLÜMÜ
TEZSİZ YÜKSEK LİSANS PROGRAMI

Yazılım Mühendisliği Temel Süreçler - PLANLAMA

Doç. Dr. Recep ERYİĞİT



HEDEFLER

- ✓ Proje kaynakları
İnsan, donanım ve yazılım kaynakları



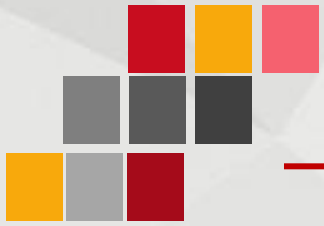
Planlama

Proje planlama aşamasında yapılan işlemler

- ❖ Kaynakların Belirlenmesi
- ❖ Maliyetlerin Kestirilmesi
- ❖ Proje Ekip Yapısının Oluşturulması
- ❖ Ayrıntılı Proje Planının Yapılması
- ❖ Projenin İzlenme Yönteminin Belirlenmesi

Çıktı: Proje Planı

Proje planı tüm proje süresince sürekli olarak kullanılacak, güncellenecek ve gözden geçirilecek bir belgedir



Proje Kaynakları

- ✓ İnsan Kaynakları
- ✓ Donanım Kaynakları
- ✓ Yazılım Kaynakları

Planlama; bu kaynakların tanımını yapar ve zaman kullanımı, görev süreleri, edinilme zamanlarını planlar



İnsan Kaynakları

- ✓ Planlama; **hangi tür elemanların**, **hangi süre ile** ve **projenin hangi aşamalarında** yer alacağını belirler

Proje Yöneticisi	Donanım Ekip Lideri
Yazılım Ekip Lideri	Donanım Mühendisi
Web Tasarımcısı	Ağ Uzmanı
Sistem Tasarımcısı	Yazılım Destek Elemanı
Programcı	Donanım Destek Elemanı
Sistem Yöneticisi	Eğitmen
Veri Tabanı Yöneticisi	Denetleyici
Kalite Sağlama Yöneticisi	Çağrı Merkezi Elemanı



Donanım Kaynakları

✓ Donanım Kaynakları:

- Ana Bilgisayarlar
- Sunucular (Web, E-posta, Veri Tabanı)
- Kullanıcı Bilgisayarları (PC)
- Yerel Alan Ağı (LAN) Alt Yapısı
- Geniş Alan Ağı (WAN) Alt Yapısı

✓ Yazılımın geliştirileceği ortam, gerçek kullanım ortamı dışında olmalıdır.

✓ Öte yandan, geliştirme ve uygulama ortamlarının aynı konfigürasyonda olmaları, ileride kurulum sırasında ortaya çıkabilecek taşıma sorunlarını büyük ölçüde giderecektir.



Yazılım Kaynakları

- ✓ Büyük ölçekte otomatik hale getirilmiş ve bilgisayar destekli olarak kullanılmaktadır.
- ✓ **Bilgisayar Destekli Tasarım** (CAD) ve **Bilgisayar Destekli Mühendislik** (CASE) araçları olarak bilinmektedirler



✓ Test araçları

- Yazılımı doğrulama ve geçerleme işlemlerinde kullanılır. Test verisi üreticiler, otomatik test yordamları, ...

✓ Prototipleme ve simülasyon araçları

- Geliştirmenin erken aşamalarında kullanıcıya, sonuç ürünün çalışması ile ilgili fikir veren ve yönlendiren araçlar.

✓ Bakım araçları

- Programın bakımını kolaylaştıran, bir kaynak koddan program şemalarının üretilmesini, veri yapısının ortaya çıkarılmasını sağlayan araçlar.

✓ Destek araçları

- İşletim sistemleri, ağ yazılımları, e-posta ve ortam yönetim araçları.



Proje Maliyetleri

✓ **Maliyet kestirimi;** bir bilgi sistemi ya da yazılım için gerekebilecek iş gücü ve zaman maliyetlerinin üretimden önce belirlenebilmesi için yapılan işlemlerdir.

✓ **Kullanılan Unsurlar**

- Geçmiş projelere ilişkin bilgiler
- Proje ekibinin deneyimleri
- İzlenen geliştirme modeli

birden çok kez uygulanabilir



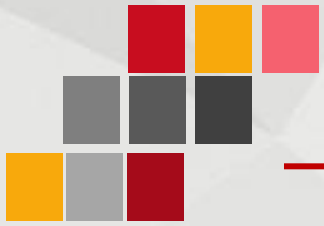
Maliyet yönetimi sayesinde;

- ✓ Gecikmeler önlenir
- ✓ Bilgi sistemi geliştirme süreci kolaylaştırılır
- ✓ Daha etkin kaynak kullanımı sağlanır
- ✓ İş zaman planı etkin olarak gerçekleştirilir
- ✓ Ürün sağlıklı olarak fiyatlandırılır
- ✓ Ürün zamanında ve hedeflenen bütçe sınırları içerisinde bitirilir



Gözlemlenebilecek değerler

- ✓ Projenin toplam süresi
- ✓ Projenin toplam maliyeti
- ✓ Projede çalışan eleman sayısı, niteliği, çalışma süresi
- ✓ Toplam satır sayısı
- ✓ Bir satırın maliyeti (ortalama)
- ✓ Bir kişi/ay'da gerçekleştirilen satır sayısı
- ✓ Toplam işlev sayısı
- ✓ Bir işlevin maliyeti
- ✓ Bir kişi/ay'da gerçekleştirilen işlev sayısı
- ✓ Bir kişi/ay'da maliyeti



Maliyet Kestirim Yöntemleri

1. Projenin boyut türüne göre

- Proje büyüklüğünü kestiren yöntemler
- Proje zaman ve işgücünü kestiren yöntemler

2. Projelerin büyüklüğüne göre

- Makro yöntemler (büyük boyutlu projeler 30 kişi-yıl)
- Mikro Yöntemler (orta ve küçük boyutlu projeler)

3. Uygulanış biçimlerine göre

- Çok yalın düzeyde
- Orta ayrıntılı düzeyde
- Çok ayrıntılı düzeyde



Maliyet Kestirim Yöntemleri

. Değişik aşamalarda kullanılabilirlik

- Planlama ve analiz aşamasında kullanılabilen
- Tasarım aşamasında kullanılabilen
- Gerçekleştirim aşamasında kullanılabilen yöntemler

5. Yöntemlerin yapılarına göre

- Uzman deneyimine gereksinim duyan
- Önceki projelerdeki bilgileri kullanan yöntemler



İşlev Noktaları Yöntemi

- ✓ İşlev noktaları geliştirmenin erken aşamalarında (analiz aşamasında) saptanan bir değerdir.
- ✓ Sistemin oluşturulduğu ortamdan bağımsız elde edilir.
- ✓ Problem tanımı girdi olarak alınarak üç temel adım izlenir:
 - Problemin bilgi ortamının incelenmesi
 - Problemin teknik karmaşıklığının incelenmesi
 - İşlev noktası hesaplama



Problemin bilgi ortamının incelenmesi

- ✓ **Kullanıcı Girdileri:** personel sicil bilgileri, personel izin bilgileri gibi
- ✓ **Kullanıcı Çıktıları:** her türlü mantıksal çıktı; raporlar, ekran çıktıları, hata iletileri,...
- ✓ **Kullanıcı Sorguları:** personel sicil bilgilerinin sorgulaması, personel maaş bilgilerinin sorgulaması
- ✓ **Dosyalar:** Her türlü mantıksal bilgi yığını, tablolar, veri tabanları
- ✓ **Dışsal arayüzler:** Başka programlarla veri iletimi. import/export

Bunların ağırlık faktörleriyle çarpımları toplanarak, **Ayarlanmamış İşlev Nokta (AİN) sayısı** hesaplanır.



Problem Bilgi Ortamı Bileşenleri

Ölçüm Parametresi	Sayı	Ağırlık Faktörü				
		Yalın	Ortalama	Karmaşık		
Kullanıcı Girdi sayısı	?	3	4	6	=	
Kullanıcı Çıktı sayısı	?	4	5	7	=	
Kullanıcı Sorgu Sayısı	?	3	4	6	=	
Kütük Sayısı	?	7	10	15	=	
Dışsal Araryüz Sayısı	?	5	7	10	=	
Toplam Sayı					=	



Problemin teknik karmaşıklığının incelenmesi

1. Uygulama, güvenilir yedekleme ve kurtarma gerektiriyor mu?
2. Veri iletişimi gerektiriyor mu?
3. Dağıtılmış İşlemler var mı?
4. Performans kritik mi?
5. Girdiler, çıktılar, dosyalar ya da sorgular karmaşık mı?
6. İçsel işlemler karmaşık mı?
7. Tasarlanacak kod yeniden kullanılabilir mi?
8. Dönüştürme ve kurulun tasarımı dikkate alınacak mı?

Cevaplar 0 ile 5 arasında puanlandırılır

Bunlar hesaplanıp toplanarak **Teknik Karmaşıklık Faktörü (TKF)** elde edilir.



İşlev noktası sayısı hesaplama

✓ $\dot{I}N = A\dot{I}N * (0,65 * 0,01 * TKF)$

Değişik amaçlarla kullanılabilir

- **Üretkenlik** = $\dot{I}N / \text{Kişi-Ay}$
- **Kalite** = $\text{Hatalar} / \dot{I}N$
- **Maliyet** = $\$ / \dot{I}N$



Satır Sayısı Kestirimi

Assembly	300
Cobol	100
Fortran	100
Pascal	90
C	90
Ada	70
Nesne Kökenli Diller	30
4. Kuşak Dilleri	20
Kod Üreticiler	15

İN=300 ise ve Nesne
Tabanlı bir dil (SmaTalk)
kullanılıyor ise

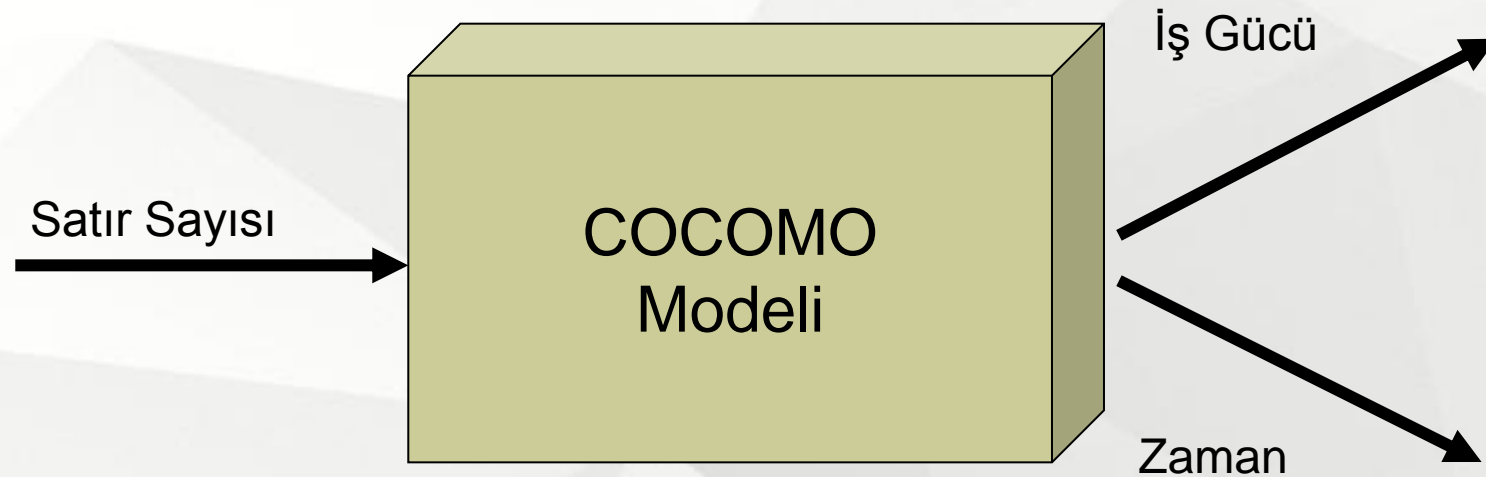
Satır Sayısı=300*30

olarak hesaplanır



Etkin Maliyet Modeli

- ✓ COCOMO 1981 Boehm
- ✓ Mikro maliyet kestirim modeline örnektir.
- ✓ Kullanılacak ayrıntı düzeyine göre üç ayrı model biçiminde yapılabilir:
 - Temel Model
 - Ara Model
 - Ayrıntı Model





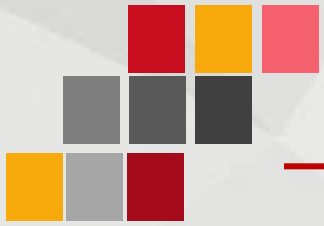
COCOMO formülleri

✓ İş Gücü (K) $K=a*S^b$

✓ Zaman (T) $T=c*K^d$

a,b,c,d : her bir model için farklı katsayılar

S : bin türünden satır sayısı



Proje Sınıfları

✓ Ayrık Projeler:

- Boyutları küçük,
- Deneyimli personel tarafından gerçekleştirilmiş
- LAN üzerinde çalışan insan kaynakları yönetim sistemi gibi

✓ Yarı Gömülü:

Hem bilgi boyutu hem donanım sürme boyutu olan projeler

✓ Gömülü Projeler:

Donanım sürmeyi hedefleyen projeler (pilotsuz uçağı süren yazılım - donanım kısıtları yüksek)



Temel Model

- ✓ Küçük-orta boy projeler için hızlı kestirim yapmak amacıyla kullanılır
- ✓ **Dezavantajı:** Yazılım projesinin geliştirileceği ortam ve yazılımı geliştirecek ekibin özelliklerini dikkate almaz
- ✓ **Avantajı:** Hesap makinesi ile kolaylıkla uygulanabilir



■ Ayrık Projeler

- İş Gücü $K=2.4*S^{1,05}$
- Zaman $T=2.5*K^{0,38}$

■ Yarı Gömülü Projeler

- İş Gücü $K=3,0*S^{1,12}$
- Zaman $T=2.5*K^{0,35}$

■ Gömülü Projeler

- İş Gücü $K=3,6*S^{1,20}$
- Zaman $T=2.5*K^{0,32}$



Ara Model

- ✓ Temel modelin eksikliğini gidermek amacıyla oluşturulmuştur.
- ✓ Bir yazılım projesinin zaman ve iş gücü maliyetlerinin kestiriminde;
 - Proje ekibinin özelliklerini,
 - Proje geliştirmede kullanılacak araçları, yöntem ve ortamı dikkate alır.
- ✓ Üç Aşamadan oluşur:
 - İş gücü hesaplama
 - Maliyet çarpanı hesaplama
 - İlk iş gücü değerini düzeltme



İş Gücü Hesaplama

✓ Ayrık Projeler

$$K=3.2*S^{1,05}$$

✓ Yarı Gömülü Projeler

$$K=3,0*S^{1,12}$$

✓ Gömülü Projeler

$$K=2.8*S^{1,20}$$



Maliyet Çarpanı Hesaplama

- ✓ Maliyet Çarpanı 15 maliyet etmeninin çarpımı sonucudur.

$$C = C1 * C2 * C3 * ... * C15$$

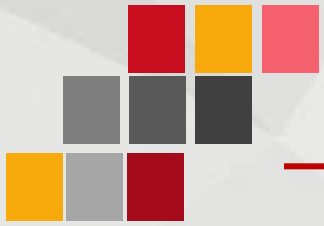
Maliyet Etmenleri

Maliyet etmeni		Seenekler					
		ok Düşük	Düşük	Normal	Yüksek	ok Yüksek	Olduka Yüksek
Ürün Özellikleri	RELY	0,75	0,88	1,00	1,15	1,40	-
	DATA	-	0,94	1,00	1,08	1,16	-
	CPLX	0,70	0,85	1,00	1,15	1,30	1,65
Bilgisayar Özellikleri	TIME	-	-	1,00	1,11	1,30	1,66
	STOR	-	-	1,00	1,06	1,21	1,56
	VIRT	-	0,87	1,00	1,15	1,30	-
	TURN	-	0,87	1,00	1,07	1,15	-
Personel Özellikleri	ACAP	1,46	1,19	1,00	0,86	0,71	-
	AEXP	1,29	1,13	1,00	0,91	0,82	-
	PCAP	1,42	1,17	1,00	0,86	0,70	-
	VEXP	1,21	1,10	1,00	0,90	-	-
	LEXP	1,14	1,07	1,00	0,95	-	-
Proje Özellikleri	MODP	1,24	1,10	1,00	0,91	0,82	-
	TOOL	1,24	1,10	1,00	0,91	0,83	-
	SCED	1,23	1,08	1,00	1,04	1,10	-



Ürün Özellikleri

- ✓ Rely: Yazılımın güvenilirliği
- ✓ Data: Veri Tabanının Büyüklüğü.
Burada program büyüklüğüne oranı dikkate alınır.
- ✓ Cplx: Karmaşıklığı.



Bilgisayar Özellikleri

- ✓ **Time:** İşletim zamanı kısıtı
- ✓ **Stor:** Ana Bellek Kısıtı
- ✓ **Virt:** Bilgisayar Platform Değişim Olasılığı.
Bellek ve Disk kapasitesi artırımı,
CPU Upgrade
- ✓ **Turn:** Bilgisayar İş Geri Dönüş Zamanı.
Hata düzeltme süresi.



Personel Özellikleri

- ✓ **Acap:** Analist Yeteneği:
Deneyim, Birlikte çalışabilirlik.
- ✓ **Aexp:** Uygulama Deneyimi.
Proje ekibinin ortalama tecrübesi.
- ✓ **Pcap:** Programcı Yeteneği.
- ✓ **Vexp:** Bilgisayar Platformu Deneyimi.
Proje ekibinin geliştirilecek platformu tanıma oranı.
- ✓ **Lexp:** Programlama dili deneyimi.



Proje Özellikleri

- ✓ **Modp:** Modern Programlama Teknikleri.
 - Yapısal programlama,
 - Görsel programlama,
 - Yeniden kullanılabilirlik.
- ✓ **Tool:** Yazılım Geliştirme araçları kullanımı.
 - CASE araçları
 - Metin düzenleyiciler
 - Ortam yönetim araçları
- ✓ **Sced:** Zaman Kısıtı.



İlk İşgücü değerini Düzeltme

✓ $K_d = K * C$

$K_d =$ Düzeltilmiş
İşgücü

* Temel Formüldeki Zamanla formülü kullanılarak zaman maliyeti hesaplanır.



Ayrıntı modeli

Temel ve ara modele ek olarak iki özellik taşır.

- ✓ Aşama ile ilgili işgücü katsayıları: her aşama için (planlama, analiz, tasarım, geliştirme, test etme) farklı katsayılar, karmaşıklık belirler
- ✓ Üç düzey ürün sıra düzeni: yazılım maliyet kestiriminde
 - Modül
 - Altsistem
 - Sistem

Sıra düzenini dikkate alır



Proje Ekip Yapısı Oluşturma

✓ PANDA proje Ekip yapısı temel olarak her proje biriminin doğrudan proje yönetimine bağlı olarak çalışması ve işlevsel bölümlenme esasına göre oluşturulur. Temel bileşenler

- Proje Denetim Birimi
- Proje Yönetim Birimi
- Kalite Yönetim Birimi
- Proje Ofisi
- Teknik Destek Birimi
- Yazılım Üretim Eşgüdüm Birimi
- Eğitim Birimi
- Uygulama Destek Birimi



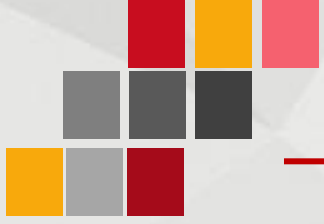
Yüklenici Proje Ekip Yapısı

- ✓ Proje Denetim Birimi: En üst düzey yönetimlerin proje ile ilgisinin sürekli sıcak tutulması ve onların projeye dahil edilmesi
- ✓ Proje Yönetim Birimi: Proje yönetiminden en üst düzeyde sorumlu birim.proje boyutuna göre bir yada daha çok yöneticiden oluşur.
- ✓ Kalite Yönetim Birimi: Projenin amacına uygunluğunu üretim süreci boyunca denetler ve onaylar
- ✓ Proje Ofisi: Her türlü yönetsel işlerden(yazışma, personel izleme) sorumlu birimdir.



Yüklenici Proje Ekip Yapısı

- ✓ Teknik Destek Birimi: Donanım, İşletim sistemi, Veri tabanı gibi teknik destek
- ✓ Yazılım Üretim Eşgüdüm Birimi: Yazılım Üretim Ekiplerinden oluşur(4-7 kişilik sayı fazla artmaz). Eğer birden fazla yazılım Üretim Ekibi varsa Ortak uygulama yazılım parçalarının geliştirilmesinden sorumlu Yazılım Destek Ekibi de olur.
- ✓ Eğitim Birimi: Proje ile ilgili her türlü eğitimden sorumludur.
- ✓ Uygulama Destek Birimi: Uygulama anında destek. (mesela telefonla)



İş Sahibi Proje Ekip Yapısı

- ✓ Proje Eşgüdüm Birimi
- ✓ Kalite Yönetim Birimi
- ✓ Proje Ofisi
- ✓ Teknik Altyapı izleme birimi
- ✓ Yazılım Üretim İzleme Birimi
- ✓ Eğitim İzleme Birimi
- ✓ Kullanıcı Eşgüdüm Birimi