



# YMT 312-Yazılım Tasarım Ve Mimarisi

## Maliyet Planlama-Gereksinim

**Dr. Öğr. Üyesi Fatih ÖZYURT**

Fırat Üniversitesi Yazılım Mühendisliği Bölümü

**Bölüm-4**

---

## Bu Haftaki Konular

Proje Planlama Aşamaları için Gerekli Koşullar .....4

Proje Maliyet Kestirim Yöntemleri .....14

Gereksinim Çözümleme Çalışması .....44

Gereksinim Veri Toplama Yöntemleri .....55

# Amaçlar



- Yazılımda Planlamanın **nasıl yapılacağını** öğrenmek,
- Planlama aşamasında gerekli olan **yöntemleri** öğrenmek,
- Planlama aşamasında **kaynakların** kullanımını görmek,
- Proje yapımında **maliyet kestirimlerini** öğrenip projede uygulamak,
- **COCOMO** Modeli hakkında bilgi sahibi olmak,
- Sistem Çözümleme **aşamasını** öğrenmek,
- Sistem çözümleme aşamasında **kullanılan yapıları** kavramak,
- Ayrıntılı bir sistemi çözümleyerek yeni bir sistem oluşturmak,
- Proje için gereksinimlerin ne anlama geldiğini öğrenmek.
- Veri akış diyagramlarının ne **tür amaca** göre oluşturulduğunu görmek,
- Genel olarak projemizde Tasarım aşamasından önceki Planlama ve Sistem Çözümleme aşamalarını kavramak.

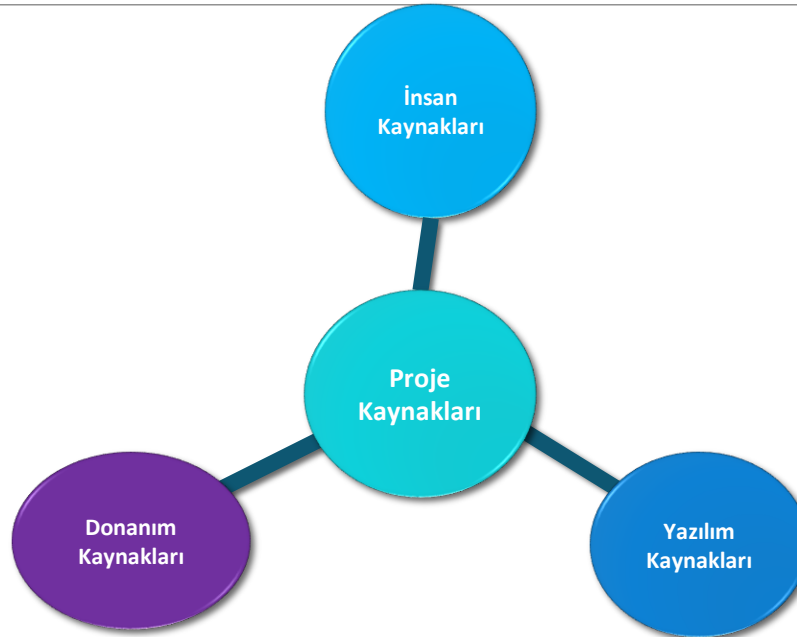
# Planlama

- Yazılım geliştirme sürecinin ilk aşaması,
- Başarılı bir proje geliştirebilmek için projenin tüm resminin çıkarılması işlemi,
- Proje planlama aşamasında yapılan işlemler
  - Proje Kaynaklarının Belirlenmesi
  - Proje Maliyetlerinin Kestirilmesi
  - Proje Ekip Yapısının Oluşturulması
  - Ayrıntılı Proje Planının Yapılması
  - Projenin İzlenmesi
- Proje planı tüm proje süresince sürekli olarak kullanılacak, güncellenecek ve gözden geçirilecek bir belgedir.



# Proje 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

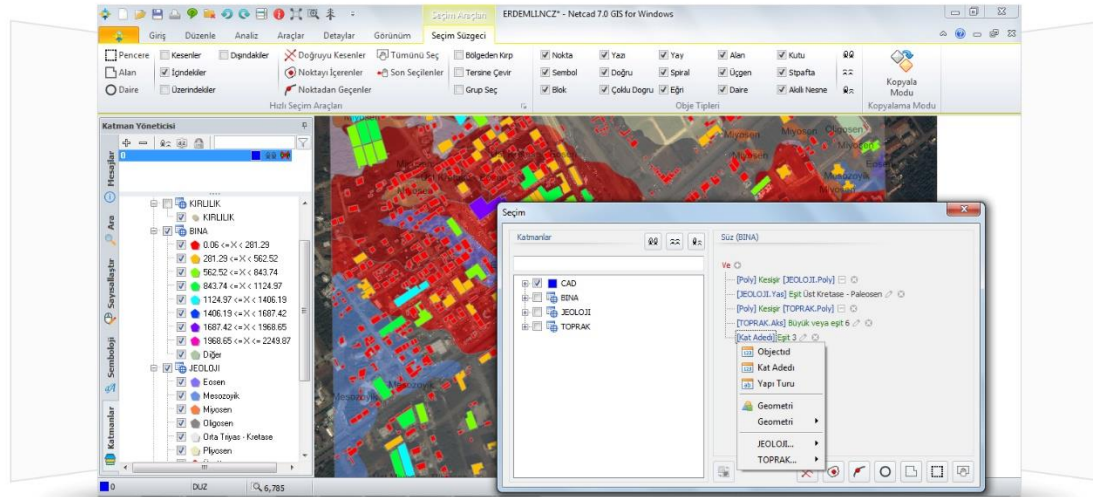
# Donanım Kaynakları

---

- Günümüzde giderek açık sistem mimarisine dönüşmektedir.
- 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.





# Yazılım Kaynakları

---

## ■ İş sistemleri planlama araçları

- İş akış yapısının üst modelinin üretilmesinde kullanılır.
- Bilgi akışı, bilgi yapısı iş birimlerindeki tıkanıklıklar bu araçlar kanalıyla ortaya çıkarılır.

## ■ Proje yönetim araçları

- Yönetici tarafından, projede yapılan işlerin izlenmesi, kaynak ataması, proje iş yapısının üretilmesi, gözlenen değerlerin işlenmesini sağlayan araçlar.

## ■ Analiz ve tasarım araçları

- Kullanılan modelleme tekniklerini ayrı ayrı ya da bütünleşik olarak uygulayan araçlar. Üretilen modelin kalitesinin ölçülmesi

## ■ Programlama araçları

- Derleyiciler, nesne-tabanlı programlama araçları, görsel programlama platformları.

# Yazılım Kaynakları

---

## ■ 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



# Proje Maliyetleri

---

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

## 4. 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 ara yüzler:** 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	=	
<b>Toplam Sayı</b>					=	

# Problemin teknik karmaşıklığının incelenmesi

1. Uygulama, güvenilir yedekleme ve kurtarma gerektiriyor mu?
2. Veri iletişimi gerektiriyor mu?
3. Dağıtılmış işlemler 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

- $İN = AİN * (0,65 * 0,01 * TKF)$

Değişik amaçlarla kullanılabilir

- **Üretkenlik** =  $İN / \text{Kişi-Ay}$
- **Kalite** =  $\text{Hatalar} / İN$
- **Maliyet** =  $\text{\$/İN}$

# 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 (SmallTalk) kullanılıyor ise

Satır Sayısı=300\*30

olarak hesaplanır.

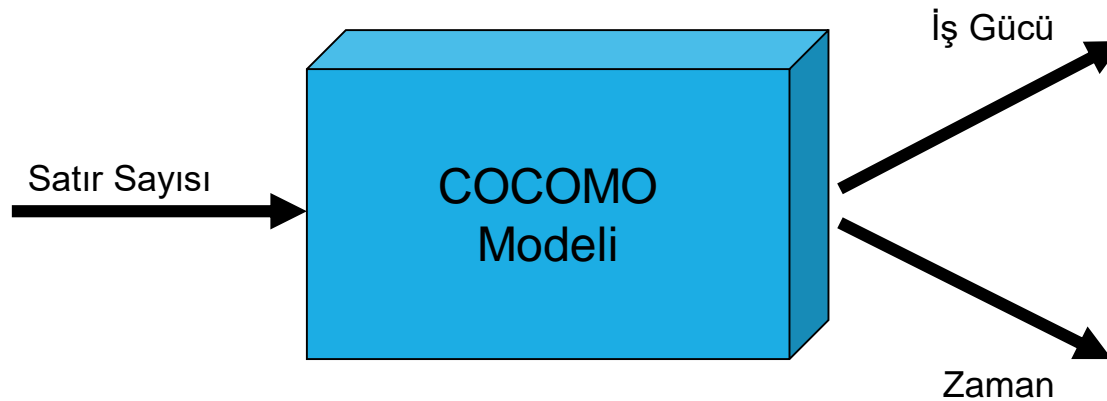
# Etkin Maliyet Modeli

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



# COCOMO Modeli

---



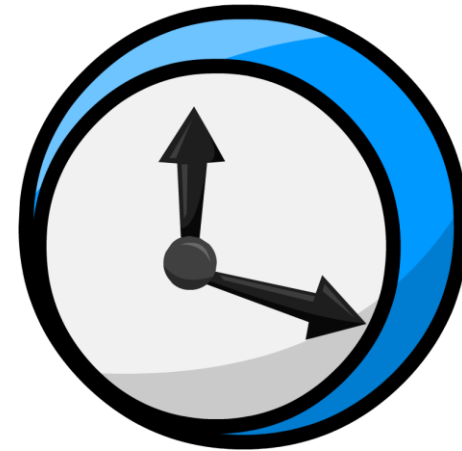
# COCOMO Formülleri

İs 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

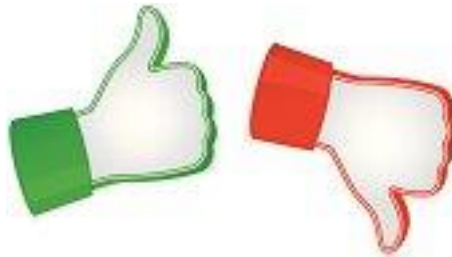
## ■ **Gömülü Projeler:**

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



# Temel Model

## ■ Ayrık Projeler

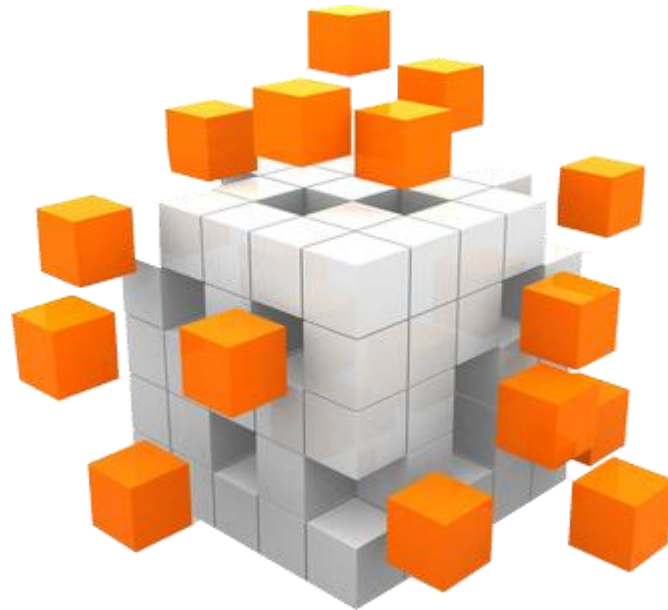
- İş Gücü  $K=2.4*S^{1,05}$
- Zaman  $T=2.5*K^{0,38}$

## ■ 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

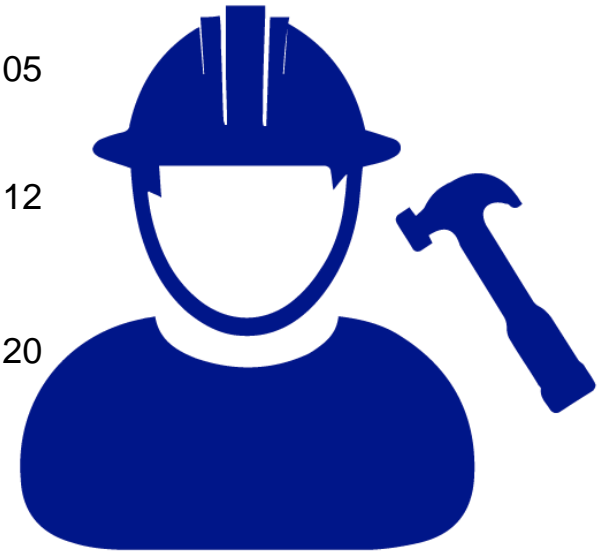
---

- Ayrık Projeler
- Yarı Gömülü Projeler
- Gömülü Projeler

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

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

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



# 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



“İyi bir ekip ruhu ile  
başarılmayacak iş yoktur!”

# 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





# Gereksinim Nedir?

---

- Sistemin amaçlarını yerine getirme yeteneği olan bir özellik ya da belirtim olarak tanımlanmaktadır.
- Gereksinim sistemin yada işlevlerinin nasıl yerine getirileceği ile ilgili değildir. **Ne olduğu ile ilgilidir.**
  - hangi veri tabanı,
  - hangi tablolar,
  - ne kadar bellek kullanılıyor,bunlar tasarım ve gerçekleştirim aşamasında ele alınır.
- Gereksinim, kullanıcı ve tasarımcı ya da yazılım mühendisi ile ilgili olarak iki amaca yönelik olacak biçimde tanımlanmalıdır;
  - **Kullanıcılar**, geliştirilecek sistemin amaçları istenilen ölçüde tanımlanmış mı sorusuna yanıt ararken,
  - **Tasarımcılar** ise gereksinimlerin tasarıma dönüştürülebilme uygunluğunu ararlar.

# Gereksinim Çeşitleri

İşlevsel  
Gereksinimler



- Sistem ile çevresi arasındaki iletişimi belirleyen gereksinimler işlevsel gereksinim olarak tanımlanır.
- İşlevsel gereksinimler ayrıca sistemin herhangi bir durum karşısındaki davranışını belirler..
- Ör: Maaş çeki hazırlama da, işlevsel gereksinimler;
  - Maaş çekinin ne zaman hazırlanacağı,
  - Çek hazırlamak için ne tür girdiler verilmesi gerektiği,
  - Hangi koşullarda bir kişiye maaş verilmeyeceği gibi gereksinimler işlevsel gereksinimlerdir.

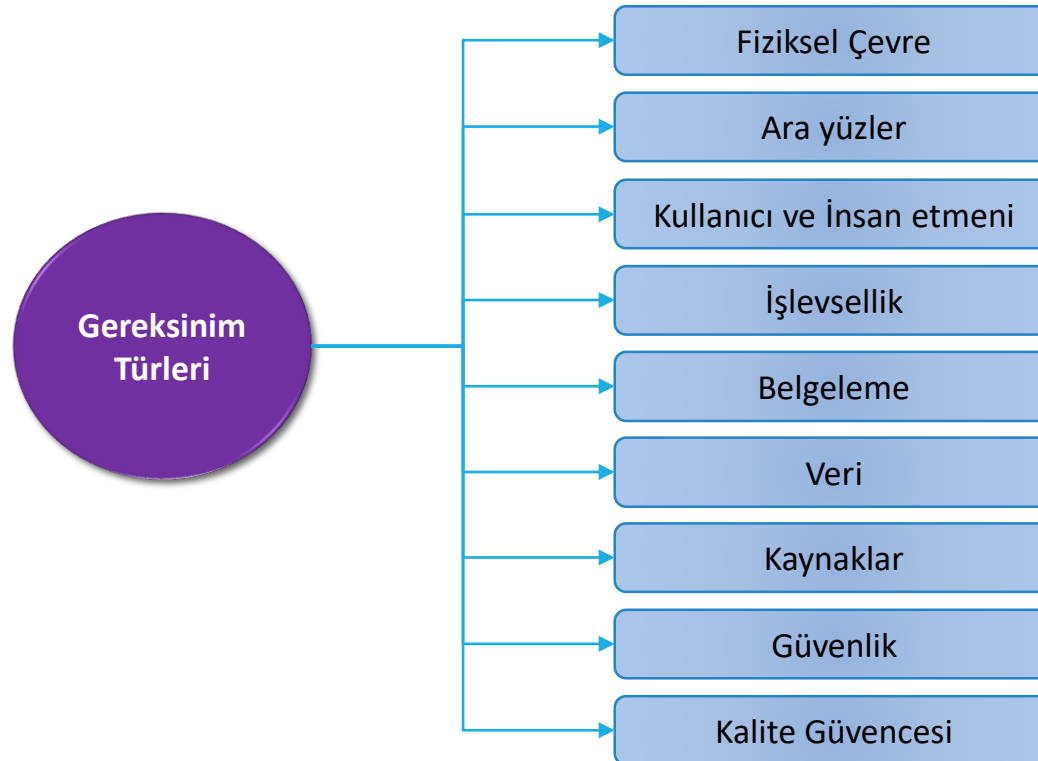
İşlevsel  
Olmayan  
Gereksinimler



- Kullanıcının sorunundan bağımsız olarak çözülmesi gereken sorunlar işlevsel olmayan gereksinim olarak tanımlanır.
- Bir çok kaynakta bu terim yerine "Sistem Kısıtları" terimi de kullanılmaktadır.
- Ör: İşlevsel olmayan gereksinimler
  - Kullanılacak bilgisayar türü,
  - yazılım geliştirme ortamı,
  - kullanılacak veri tabanı yönetim sistemi vb.bu tür gereksinimlere örnek verilebilir.

# Gereksinim Türleri

---



# Fiziksel Çevre

---

- İşlevlerin geliştirileceği, işletileceği aygıtlar nerededir.
- Sistem tek bir yerde mi olacak? birden çok ve fiziksel olarak birbirinden ayrılmış yerler söz konusu mu?
- Sıcaklık nem oranı veya manyetik etkileşim gibi çevresel kısıtlamalar var mı?



# Ara yüzler

---

- Girdiler bir mi yoksa birden çok sistemden mi geliyor?
- Çıktılar bir mi yoksa birden çok sisteme mi gidiyor?
- Verilerin nasıl biçimlendirileceğine ilişkin bir yol var mı?
- Verilerin kullanılacağı önerilen bir ortam var mı?



# Kullanıcı ve İnsan etmeni

---

- Sistemi kim kullanacak?
- Farklı tiplerde kullanıcılar olacak mı?
- Her bir kullanıcı tipinin yetenek düzeyi nedir?
- Her kullanıcı tipi için ne tür eğitimler gerekli?
- Bir kullanıcının sistemi kötü amaçlı kullanması ne ölçüde zordur?

# İşlevsellik

---

- Sistem ne yapacak?
- Sistem bunu ne zaman gerçekleştirecek?
- Sistem nasıl ve ne zaman değiştirilebilir ve/veya güçlendirilebilir?
- Çalışma hızı, yanıt süresi ya da çıktı üzerinde kısıtlayıcı etmenler var mı?



# Belgeleme

- Ne kadar belgeleme gereklidir?
- Belgeleme hangi kullanıcı kitlesini hedeflemektedir?





# Veri

---

- Hem giriş hem çıkış için verinin biçimi ne olmalıdır?
- Bu veri ne sıklıkla alınacak veya gönderilecektir?
- Bu verinin doğruluk (kesinlik) ölçüsü ne olmalıdır?
- Hesaplamalar hangi duyarlık derecesine kadar yapılandırılacaktır?
- Sistemde ne kadar veri akışı olacaktır?
- Veri belirli bir zaman süresince kaynağında saklanacak mı?



# Kaynaklar

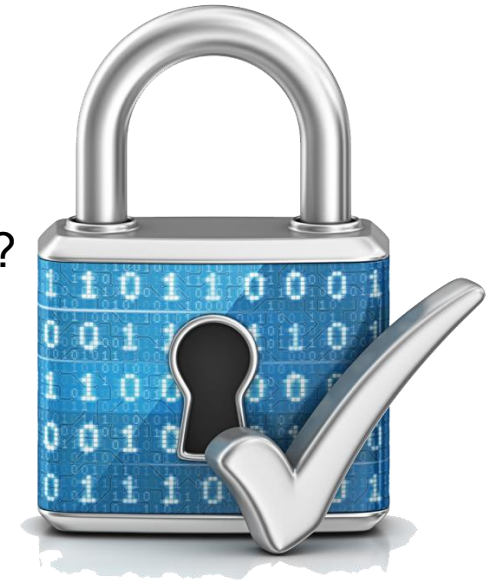
---

- Sistemi kurmak, kullanmak ve bakımını yapmak için ne kadar malzeme, personel ve diğer kaynaklara ihtiyaç var?
- Geliştiriciler hangi yeteneklere sahip olmalı?
- Sistem ne kadar fiziksel yer kaplayacak?
- Güç, ısıtma ve soğutma için kısıtlar nelerdir?
- Geliştirim için tavsiye edilen bir zaman çizelgesi var mı?

# Güvenlik

---

- Sisteme ya da bilgiye erişim denetlenmeli midir?
- Bir kullanıcının verisi diğerinden nasıl ayrılacaktır?
- Kullanıcı programları, diğer program ve işletim sisteminden nasıl ayrı tutulacaktır?
- Sistem hangi sıklıkla yedeklenecektir?
- Yedek kopyaları başka yerde saklanacak mıdır?
- Yangın ve hırsızlığa karşı ne tür önlemler alınacaktır?
- İnternet erişimi var mı? Güvenlik kullanılıyor mu?



# Kalite Güvencesi

---

- Güvenirlilik için gereksinimler nelerdir?
- Sistemin özellikleri insanlara nasıl aktarılmalıdır?
- Sistem çökmeleri arasında öngörülen zaman aralığı nedir?
- Kaynak kullanımı ve yanıt süresine ilişkin verimlilik ölçütleri nelerdir?
- Hataları kendisi bulup gidermeli mi?
- Çökmeler?
- Tasarımda yapılacak değişiklikler
- Kaynak kullanımı ve yanıt süresine ilişkin verimlilik ölçütleri?
- Sistemi bir bilgisayardan diğerine aktarmalı mı?

# Gereksinim Özellikleri

Gereksinimler üç amaca hizmet eder. Bunlar;

1. Geliştiricilerin, müşterilerin sistemin nasıl çalışmasını istediklerini anlamalarını sağlar.
2. Gereksinimler, sonuç sistemin ne özellikte ve işlevsellikte olacağını söyler.
3. Gereksinimler sınama ekibine, kullanıcıyı, sunulan sistemin istenen sistem olduğuna ikna etmek için neler göstermeleri gerektiğini söyler.



# Özet

---

- **Planlama**, yazılım geliştirme sürecinin ilk aşamasıdır.
- **Proje Kaynakları**: İnsan, donanım ve yazılım olarak 3 kategoride incelenir.
- **Maliyet kestirimi**; bir bilgi sistemi ya da yazılım için gerekebilecek iş gücü ve zaman maliyetlerinin üretimden önce belirlenebilmesi için yapılan işlemlerdir.
- Problemin bilgi ortamının incelenmesinde, **kullanıcı girdileri, kullanıcı çıktıları, kullanıcı sorguları, dosyalar** ve **dışsal ara yüzler** mevcuttur.
- **Proje Sınıfları**: Ayrık, Yarı Gömülü ve Gömülü Projeler olmak üzere 3 şekildedir.
- Maliyet Çarpanı hesaplanması 15 maliyet etmeninin çarpımı sonucunda bulunur.
- COCOMO Modeli kullanılacak ayrıntı düzeyine göre üç ayrı model biçiminde yapılabilir bunlar: Temel, Ara ve Ayrıntı Modelidir.
- Gereksinim, sistemin amaçlarını yerine getirme yeteneği olan bir özellik ya da belirtim olarak tanımlanmaktadır.

# Özet

---

- Gereksinim, kullanıcı ve tasarımcı ya da yazılım mühendisi ile ilgili olarak iki amaca yönelik olacak biçimde tanımlanmalıdır;

# Sorular

---

1. Kaç çeşit proje kaynağı mevcuttur?
2. Maliyet kestirimi nedir ve ne için yapılmaktadır?
3. Proje sınıfları nelerdir?
4. COCOMO nedir?
5. Yazılımda Sistem .çözümlemesi aşamasını açıklayınız.
6. Gereksinim nedir ve ne amaçla yapılmaktadır?



# Kaynaklar

---

“Software Engineering A Practitioner’s Approach” (7th. Ed.), Roger S. Pressman, 2013.

“Software Engineering” (8th. Ed.), Ian Sommerville, 2007.

“Guide to the Software Engineering Body of Knowledge”, 2004.

” Yazılım Mühendisliğine Giriş”, TBİL-211, Dr. Ali Arifoğlu.

”Yazılım Mühendisliği” (2. Basım), Dr. M. Erhan Sarıdoğan, 2008, İstanbul: Papatya Yayıncılık.

Kalıpsız, O., Buharalı, A., Biricik, G. (2005). Bilgisayar Bilimlerinde Sistem Analizi ve Tasarımı Nesneye Yönelik Modelleme. İstanbul: Papatya Yayıncılık.

Buzluca, F. (2010) Yazılım Modelleme ve Tasarımı ders notları (<http://www.buzluca.info/dersler.html>)

Hacettepe Üniversitesi BBS-651, A. Tarhan, 2010.

Yazılım Proje Yönetimi, Yrd. Doç. Dr. Hacer KARACAN

# Sorularınız

---

