

5. Hafta

# YMÜ225

## Yazılım Gereksinim ve Analizi

Dr. Feyza Altunbey Özbay

# İçerik

- Planlama
- Proje Planı
- Proje Kaynakları
- Proje Maliyetleri
- İşlev Noktaları Yöntemi

# Planlama

- Yazılım geliştirme sürecinin ilk aşamasıdır.
- Başarılı bir proje geliştirebilmek için projenin tüm resminin çıkarılması gerekmektedir. Bu resim, proje planının çıkarılması ile mümkündür.

Proje planlaması, proje yaşam döngüsünün üç aşamasında gerçekleşir:

- Bir yazılım sistemi geliştirmek veya temin etmek amacıyla bir ihale için teklif verildiği öneri aşamasında
- Projede kimlerin çalışacağını, projenin artırımlara nasıl bölüneceğini, şirkette kaynakların nasıl paylaşılacağını planlanması gereken başlangıç aşamasında
- Periyodik olarak proje boyunca

# Planlama

- Proje planlama aşamasında yapılması gereken çalışmalar:

- Proje Kaynaklarının Belirlenmesi
- Proje Maliyetlerinin Kestirilmesi
- Proje Ekip Yapısının Oluşturulması
- Ayrıntılı Proje Planının Yapılması
- Projenin İzlenmesi

biçiminde özetlenebilir.

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

# Yazılım Fiyatlandırma

Prensip olarak, bir müşteri için geliştirilen bir yazılım sisteminin fiyatı, basitçe geliştirme maliyeti + geliştiricinin kârıdır.

Ancak uygulamada, proje maliyeti ve müşteriye teklif edilen fiyat arasındaki o kadar basit değildir.

Faktör	Tanım
Sözleşme hükümleri	Bir müşteri, geliştiricinin, kaynak kodun sahipliğini elinde tutmasına ve onu diğer projelerde yeniden kullanmasına izin vermeye istekli olabilir. Faturalanacak fiyat, kaynak kodun geliştirici için değerini yansıtacak şekilde azaltılabilir.
Maliyet tahmini belirsizliği	Bir kurum, maliyet tahmini konusunda emin değilse, fiyatı, beklenmedik bir duruma karşı normal kârına ilaveten arttırabilir.
Finansal sağlık	Finansal problemleri olan şirketler, bir ihaleyi kazanmak için fiyatlarını düşürebilir. Normalden daha az kâr etmek ya da başa baş gelmek, iflas etmekten daha iyidir. Zor ekonomik zamanlarda nakit akışı kârdan daha önemlidir.
Pazar fırsatı	Bir geliştirme şirketi, yazılım pazarında yeni bir faaliyet alanına girmek istediği için, düşük fiyatlı teklif verebilir. Bir projede düşük kâr etmeyi kabul etmek, şirkete daha sonra daha büyük bir kâr sağlayabilme fırsatı verebilir. Kazanılan deneyim ayrıca yeni ürünler geliştirmek için yardımcı olabilir.
Gereksinim değişkenliği	Gereksinimlerin değişme olasılığı varsa, bir şirket, bir ihaleyi kazanmak için fiyatını düşürebilir. İhale kazanıldıktan sonra, gereksinimlerdeki değişiklikler için yüksek fiyatlar talep edilebilir.

# Proje Planı

Genel olarak bir proje planı aşağıdaki adımları içerir:

- **Giriş:** Kısaca projenin hedeflerini anlatır ve projenin yönetimini etkileyen kısıtlamaları (bütçe, süre gibi) ortaya koyar.
- **Proje organizasyonu:** Geliştirme takımını organize edilme şeklini, dahil olan kişileri ve bu kişilerin takımdaki rollerini tanımlar.
- **Risk analizi:** Olası proje risklerini, bu risklerin ortaya çıkma olasılıklarını ve önerilen risk azaltma stratejilerini tanımlar.
- **Donanım ve yazılım kaynak gereksinimleri:** Geliştirmeyi yapmak için gerekli donanım ve destek yazılımı belirtir. Donanım satın almak gerekiyorsa fiyatlar ve teslimat takvimi için tahminler dahil edilebilir.

# Proje Planı

- **İş dağılımı:** Projenin etkinliklere dağıtımını gösterir ve her bir proje etkinliğinin girdi ve çıktılarını gösterir.
- **Proje zaman planı:** Etkinlikler arasındaki bağımlılıkları ve kişilerin etkinliklere dağılımını gösterir.
- **İzleme ve zamanlama mekanizmaları:** Üretilmesi gereken yönetim raporlarını, bunların ne zaman üretileceğini ve kullanılacak proje izleme yöntemlerini tanımlar.

# Proje Planı

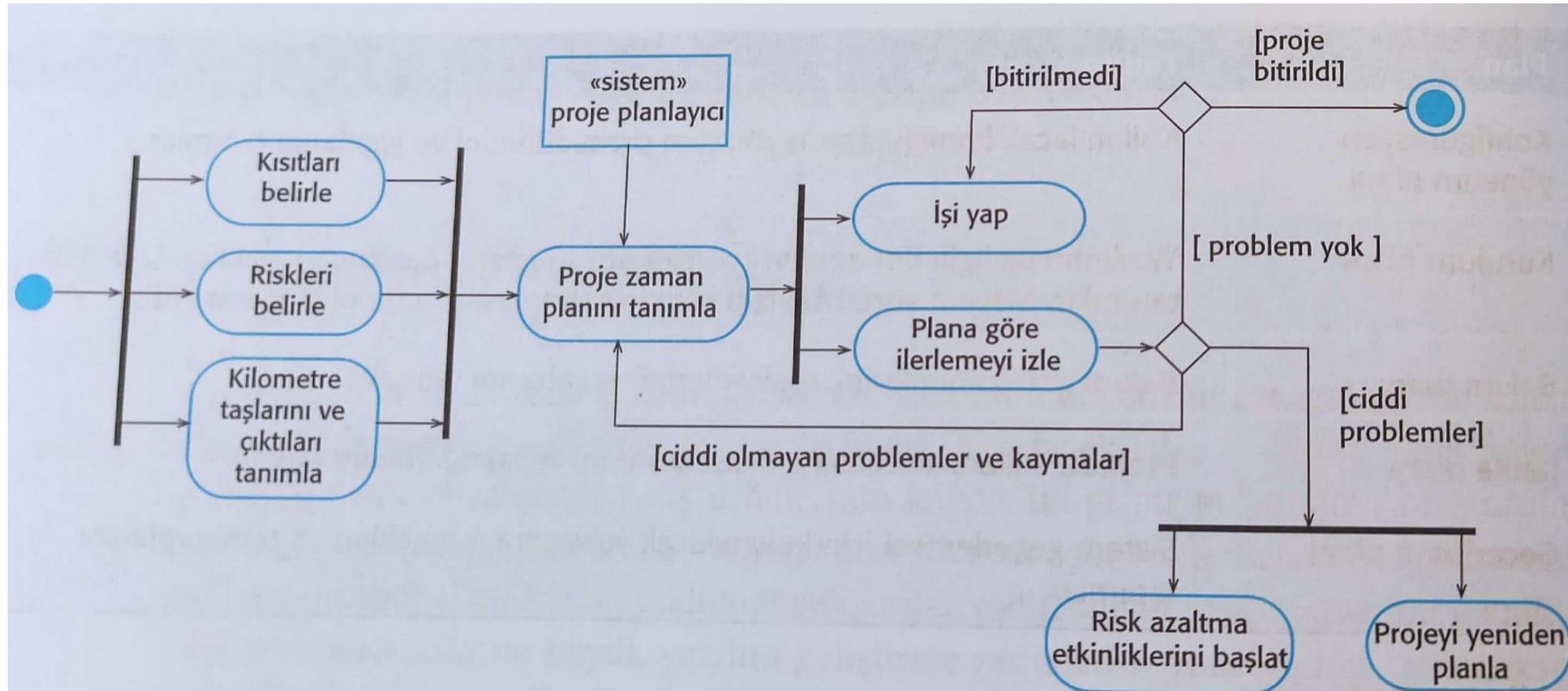
- Proje planının ek özellikleri

Plan	Tanım
Konfigürasyon yönetim planı	Kullanılacak konfigürasyon yönetim prosedürlerini ve yapılarını tanımlar.
Kurulum planı	Yazılımın ve ilgili donanımın (gerekliyse) müşteri ortamında nasıl kurulacağını tanımlar. Mevcut sistemlerdeki verinin taşınması için bir plan içermelidir.
Bakım planı	Bakım gereksinimlerini, maliyetlerini ve çabasını tanımlar.
Kalite planı	Projede kullanılacak kalite prosedürlerini ve standartlarını tanımlar.
Geçerleme planı	Sistem geçerlemesi için kullanılacak yaklaşımı, kaynakları ve zaman planını tanımlar.



# Proje Planlama Süreci

Projenin başlangıç aşamasında proje planı ilk oluşturduğumuz zaman başlayan yinelemeli bir süreçtir.



# Proje Kaynakları



Projelerin başarılı olarak yönetilmesi, bir projenin gerçekleştirilmesi için harcanan çabaların tümüdür. Diğer bir ifadeyle proje yönetimi, belirli amaç ve hedeflere ulaşabilmek için işletme kaynaklarının verimli ve etkin bir şekilde planlanması, organize edilmesi, yönetilmesi ve kontrolüdür

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

- Donanım kaynakları planlanırken, yazılımın geliştirileceği ortamın, gerçek kullanım ya da uygulama ortamı dışında bulundurulmasına özen gösterilmelidir.
- Geliştirilme sırasında ortaya çıkabilecek donanımsal hataların uygulamayı etkilememesi gerekmektedir.
- Geliştirme ortamı ile uygulama ortamlarının aynı konfigürasyonda olmaları, ileride kurulum, gerçekleştirme sırasında ortaya çıkabilecek sorunları büyük ölçüde giderecektir.

# Yazılım Kaynakları

- **İş Sistemleri Planlama Araçları:** Kurumlardaki iş akış yapısının üst modelinin üretilmesinde kullanılmaktadır. Bilgi akışı, bilgi yapısı, iş birimlerindeki tıkanıklıklar bu araçlar kanalıyla ortaya çıkarılır.
- **Proje Yönetim Araçları:** Proje yöneticisi tarafından, projede yapılan işlerin izlenmesi, kaynak ataması, proje iş yapısının üretilmesi, gözlenen değerlerin işlenmesi türündeki işlerin yapılmasını sağlayan araçlardır.
- **Çözümleme ve Tasarım Araçları:** Sistem yazılım kullanıcı yordamları, metin düzenleyiciler, derleyiciler, hata ayıklayıcılar, nesne kökenli programlama araçları, görsel programlama platformları türündeki programlama araçları yazılım geliştirmede kullanılan araçlardır.
- **Programlama Araçları:** Doğruluk, bilginin hatasız olması ile özdeştir. Büyük bir veri yığınıyla uğraşıldığında, genelde kayıt ve hesaplama hataları ortaya çıkar. Bu gibi durumlarda, doğruluk özelliği daha fazla önem kazanır

# Yazılım Kaynakları

- **Sinama Araçları:** Kapsam çözümleyiciler, sinama verisi üreticiler, otomatik sinama yordamları, yazılımın doğrulama ve geçerleme işlemlerinde kullanılmaktadır.
- **Prototipleme ve Benzetim Araçları:** Bu araçları temel olarak, geliştirmenin erken aşamalarında kullanıcıya, sonuç ürünün çalışması ile ilgili fikir vermek ve yönlendirmek amacıyla kullanılmaktadır. Bu araçlar sayesinde, kullanıcı sonuçta elde edeceği ürünün davranışı ile ilgili bilgiler edinir ve sonradan ortaya çıkabilecek farklı yorum ve algılamalar önlenmiş olur.
- **Bakım Araçları:** Bu araçlar, verilen bir kaynak kodundan, program şemalarının üretilmesi, program veri yapısının ortaya çıkarılması gibi işlevleri yerine getirirler.
- **Destek Araçları:** İşletim sistemleri, belge işleme sistemleri, ağ yazılımları, elektronik posta ve ortam yönetim araçları, bu araçlara örnek olarak verilebilir.

# Proje Maliyetleri

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

Maliyet kestirimi yapılırken kullanılan unsurlar:

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

birden çok kez uygulanabilir.



# Maliyet Yönetimi

Maliyet yönetimi ile;

- gecikmelerin önlenmesi,
- bilgi sistemi geliştirme sürecinin kolaylaştırılması ,
- daha etkin kaynak kullanımının sağlanması,
- iş zaman planının etkin olarak gerçekleştirilmesi,
- ürünün sağlıklı olarak fiyatlandırılması,
- ürünün zamanında ve hedeflenen bütçe sınırları içerisinde bitirilmesi

sağlanır.

# Öngörülebilir Değerler

Bir projenin tümü ya da belirli bir kısmı bitirildikten sonra projeye ilişkin bir takım bilgiler daha sonraki projelerde maliyet kestirimi açısından oldukça önem taşır. Bu tür bilgilere örnek olarak:

- Projenin (başlangıçtan sonuca) toplam süresi,
- Projenin (başlangıçtan sonuca) toplam maliyeti,
- Projede çalışan eleman sayısı, niteliği, çalışma süreleri,
- Toplam satır sayısı,
- Bir satırın parasal maliyeti (ortalama),
- Bir kişi-ay da gerçekleştirilen satır sayısı (ortalama),
- Toplam işlev noktası sayısı,
- Bir işlev noktasının parasal maliyeti (ortalama),
- Bir ayda gerçekleştirilen işlev noktası sayısı (ortalama),
- Bir kişi-ay da gerçekleştirilen işlev noktası sayısı (ortalama),
- Bir kişi-ay 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. Uygulama 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

- Yazılım projelerinin başlangıcında yaşanan en büyük zorluk, geliştirme süreci sonucunda ortaya çıkacak yazılım ürününün büyüklüğünün doğru tahmin edilememesidir.
- Bu tahminin yapılma aşamasında kullanılabilecek bir çok metot mevcuttur. Bunlardan en yaygın olarak kullanılanı ise «İşlev Noktaları Analizi» dir.
- Bu analizin, kullanılacak yazılım kodlama dilinden bağımsız olarak yapılabiliyor olması, bunun her türlü yazılım projesinde uygulanabilirliğini sağlamaktadır.

# İşlev Noktaları Yöntemi

- İşlev noktalarının hesaplamasında problem tanımı girdi olarak alınarak üç temel adım izlenir:
  1. Problemin bilgi ortamının incelenmesi
  2. Problemin teknik karmaşıklığının incelenmesi
  3. İşlev noktası hesaplanması

# Problemin Bilgi Ortamının İncelenmesi

## **Kullanıcı Girdileri**

Yazılıma girdi olarak verilen her farklı uygulama bileşeni bir kullanıcı girdisi olarak sayılır. Kullanıcı girdilerinin sayısı hesaplanırken, alan bazında değil, daha genel olarak mantıksal kayıt bazında uygulama yapılmalıdır. Örneğin, bir personel sisteminin geliştirilmesinden söz edildiğinde, personel sicil bilgileri, personel izin bilgileri gibi bilgiler ayrı kullanıcı girdileri olarak sayılabilir. Personel sicil no, personel adı soyadı türündeki alan bilgileri kullanıcı girdisi olarak sayılmamalıdır.

## **Kullanıcı Çıktıları**

Kullanıcıyı ilgilendiren her tür mantıksal çıktı, kullanıcı çıktısı olarak sayılmalıdır. Bu kapsamda kullanıcı çıktılarına örnek olarak, raporlar, ekran çıktıları, hata iletileri vb. verilebilir. Girdiler gibi çıktılar da genel olarak sayılmalıdır. Bir raporun içerisindeki bir alan kullanıcı çıktısı olarak sayılmamalıdır. Öte yandan, aynı bilgileri içeren ancak bilgi düzeni ya da sıralaması farklı olan raporlar da ayrı kullanıcı çıktısı olarak ele alınmamalıdır. Örneğin, soyadına göre sıralanmış personel listesi ve sicil no'suna göre sıralanmış personel listesi raporları farklı kullanıcı çıktıları olarak sayılmamalıdır.

# Problemin Bilgi Ortamının İncelenmesi

## **Kullanıcı Sorguları**

Kullanıcı sorgusu, çevrim içi olarak bilgisayara verilen bir girdi sonucu yine çevrim içi olarak bir kullanıcı çıktısı alınması biçiminde tanımlanmaktadır. Her farklı sorgu ayrı olarak sayılmalıdır. Örneğin; personel sicil bilgilerinin sorgulanması, personel maaş bilgilerinin sorgulanması vb.

## **Kütükler**

Her mantıksal bilgi yığını ya da kütük ayrı olarak sayılmalıdır. Personel sicil kütüğü, personel maaş kütüğü, personel hareket kütüğü vb.

## **Dışsal Arayüzler**

Geliştirilmesi öngörülen bilgi sisteminin, gerek kurum içinde gerekse kurum dışında bir başka bilgi sistemi ile bilgisayar ortamında bir iletişimi (çevrim dışı disket vb. ya da çevrim-içi doğrudan bağlantı) söz konusu ise bu durum bir dışsal arayüz olarak sayılmalıdır. Kağıt ya da rapor türündeki arayüzler bir dışsal arayüz sayılmazlar. Örneğin, personel bilgi sistemi, yine aynı kurumda olduğu varsayılan muhasebe bilgi sistemine, personel maaşlarına ilişkin bilgiyi bilgisayar ortamında aktarması bir dışsal arayüz olarak sayılabilir.

# Problemin Bilgi Ortamının İncelenmesi

Bu 5 ana bileşenin projedeki tahmini toplamı ile ağırlık faktörünün çarpımı sonucu ayarlanmamış **ayarlanmış işlev nokta sayısı** elde edilecektir.

Örnek:

Ölçüm Parametresi	Sayı	Ağırlık Faktörü		
		Basitliği	Ortalama	Karmaşıklığı
Girdiler	26	3	4	6
Çıktılar	24	4	5	7
Dışsal Talepler	11	3	4	6
Kütükler	63	7	10	15
Dışsal ara yüzler	0	5	7	10

Ayarlanmış İşlev Nokta Sayısı= $\sum$  (sayı \* Ortalama)

$$AİN=(26 * 4) + (24 * 5) + (11 * 4) + (63 * 10) + (0 * 7)$$

$$=104 + 120 + 44 + 630 + 0$$

$$= 898 \text{ dir.}$$



Karakteristik	0	1	2	3	4	5
1-sistem güvenilir yedekleme ve kurtarma gerektiriyor mu?						*
2-Veri iletişimi gerekiyor mu?						*
3-Dağıtılmış işlem fonksiyonları var mı?				*		
4- Performans kritik mi?			*			
5- Sistem varolan, ağır yükü olan bir işletim ortamında mı çalışacak?			*			
6- Sistem çevrimiçi (on-line) veri girişi gerektiriyor mu?				*		
7- Sistem çevrimiçi (on-line) veri girişi gerektiriyor ve giriş işlemi birçok ekran veya işlem üzerine mi inşa edilecek?				*		
8- Ana dosya/kütük çevrimiçi olarak mı güncellenecek?					*	
9-Girdiler çıktılar, dosyalar/kütükler veya sorgular karmaşık mı?					*	
10-Dâhili işlemler karmaşık mı?					*	
11-Tasarlanacak kod tekrar kullanılabilir mi olacak?					*	
12- Dönüştürme ve yükleme tasarıma dahil mi olacak?					*	
13- Sistem farklı kuruluşlar ve birden fazla yükleme için mi tasarlanmıştır?		*				
14-Tasarlanan uygulama, kullanıcı tarafından kolay kullanılabilir ve kolay değiştirebilir mi olacak?						*

# Projenin Teknik Karmaşıklığının İncelenmesi

Teknik Karmaşıklık bulabilmek için bazı sorulara yanıt verilir. Cevaplar 0 ile 5 arasında puanlandırılır.

- ✓ 0: hiç yok ya da etkisiz,
- ✓ 1: önemsiz etki,
- ✓ 2: az etkili ,
- ✓ 3:orta düzeyde etkili
- ✓ 4: önemli düzeyde etkili,
- ✓ 5: güçlü etki

TKF=49

# İşlev Nokta Sayısı

Problem bilgi ortamı için elde edilen değerler, karmaşıklık düzeylerine göre seçilerek karmaşıklık değeri ile çarpılır ve çarpımların sonucu elde edilen toplam işlev nokta sayısı, **işlev nokta sayısı** (Unadjusted Function Points\_UFP) olarak adlandırılır.

İşlev Nokta Sayısı Hesaplama

$$\dot{IN} = A\dot{IN} * (0,65 + 0,01 * TKF)$$

$$\dot{IN} = A\dot{IN} * (0,65 + 0,01 * 49)$$

$$= 898 * (0,65 + 0,01 * 49) = 1023,72$$

# Örnek Proje

Girdiler:	2 Basit
	2 Orta
Çıktılar:	3 Orta
	1 Karmaşık
İç Dosyalar :	1 Orta
➤ Personel Demirbaş Dosyası	
Dış Arayüz Dosyaları:	1 Orta
➤ Personel Listesi	

$$AİN = [Dış Girdiler \times W(1)] + [Dış Çıktılar \times W(2)] + [Dış Sorgular \times W(3)] + [İç Mantıksal Dosyalar \times W(4) + Dış Arayüz Dosyaları \times W(5)]$$

$$AİN = [(2 \times 3) + (2 \times 5)] + [(3 \times 6) + (1 \times 7)] + [1 \times 13] + [1 \times 5] = 59$$

		Basit	Orta	Karmaşık
(1)	Dış Girdiler	3	5	6
(2)	Dış Çıktılar	4	6	7
(3)	Dış Sorgular	3	5	6
(4)	İç Dosyalar	7	13	15
(5)	Dış Arayüz Dosyaları	5	9	10

# Teknik Karmaşıklık

1.	Sistem güvenilir yedekleme ve kurtarma gerektiriyor mu?	2
2.	Veri iletişimi gerekiyor mu?	2
3.	Dağıtık fonksiyon var mı?	3
4.	Performans kritik mi?	2
5.	Sistem çok kullanılan bir işletim ortamında mı çalışacak?	2
6.	Sistem on-line veri girişi gerektiriyor mu?	3
7.	On-line veri girişi, giriş işlemlerinin birden fazla ekran ya da işlem üzerinden olmasını mı gerektiriyor?	4
8.	Ana dosyalar on-line mı güncelleniyor?	3
9.	Girdiler, çıktılar, dosyalar ve sorgular karmaşık mı?	1
10.	Kod yeniden kullanılabilir olarak mı tasarlanmış?	3
11.	İç süreç karmaşık mı?	2
12.	Dönüşüm ve kurulum tasarım içerisinde mi?	1
13.	Uygulama değişik kurulumlarda birden fazla kurulum gerektirecek şekilde mi tasarlanmış?	1
14.	Uygulama kullanıcı tarafından kolaylıkla kullanmayı ve değiştirmek üzere mi tasarlanmış?	5

$$TKF = \sum \text{Cevap } i = 34$$

$$iN = A_iN \times (0,65 + 0,01 \times TKF) = 59 \times (0,65 + 0,01 \times 34) = 58,41$$

# Satır Sayısı Kestirimi

İşlev Noktaları, başka yöntemlerle birlikte kullanılmak istenildiğinde, kullanılacak yazılım geliştirme platformuna göre tablodaki değerler kullanılarak kolayca satır-sayısı kestirimine dönüştürülebilir

Ortalama Satır Sayıları:

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

$$\begin{aligned}\text{Satır Sayısı} &= \text{İN} * \text{Nesneye kökenli dillerin Ortalama Satır sayısı} \\ &= 1023,72 * 30 = 30711,6 \text{ dir.}\end{aligned}$$

# Farklı Parametreler

İşlev nokta sayısı farklı ölçüm parametrelerinde kullanılabilir:

Üretkenlik = işlev nokta sayısı / Kişi-Ay

Kalite = Hatalar / işlev nokta sayısı

Maliyet= TL/ işlev nokta sayısı