



# YMT 312-Yazılım Tasarım ve Mimarisi

## Yazılım Mühendisliği'ne Giriş

Bölüm-1

**Doç. Dr. Resul DAŞ**

Fırat Üniversitesi Yazılım Mühendisliği Bölümü

## Bu Haftaki Konular

Yazılım Nedir?.....	4
Yazılım Mühendisliği.....	15
Yazılımların Sınanması.....	20
Yazılım Maliyetleri.....	22
Yazılım Sistemlerinin Sınıflandırılması.....	23
Yazılım Kalite.....	27

# Amaçlar

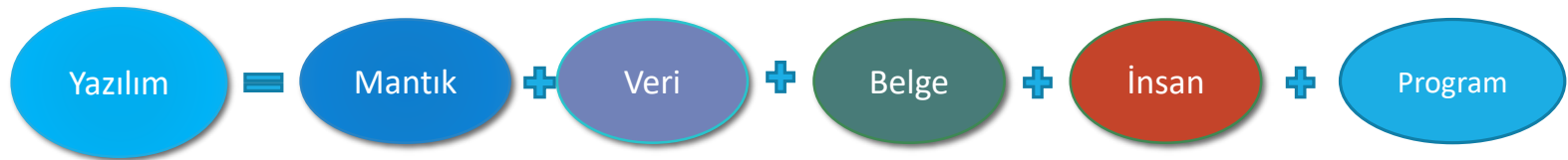
---

- Yazılım nedir?
- Yazılım ve Donanım karşılaştırılması?
- Yazılım Mühendisliği nedir?
- Yazılım Mühendisi kime denir?
- Yazılımda Hatalar?
- Yazılım Maliyetleri?
- Yazılımda Kalite?



# Yazılım Nedir?

---



- Bileşenlerinin, belirli bir üretim amacına yönelik olarak bir araya getirilmesi, yönetilebilmesi için kullanılabilecek ve üretilen, yöntem, araç, bilgi ve belgelerin tümünü içerir.
- Yazılım en yalın biçimiyle, **"Bir sistemin donanım bileşenleri dışında kalan her şey"** olarak tanımlanabilir.

# Mantık (Algoritma)

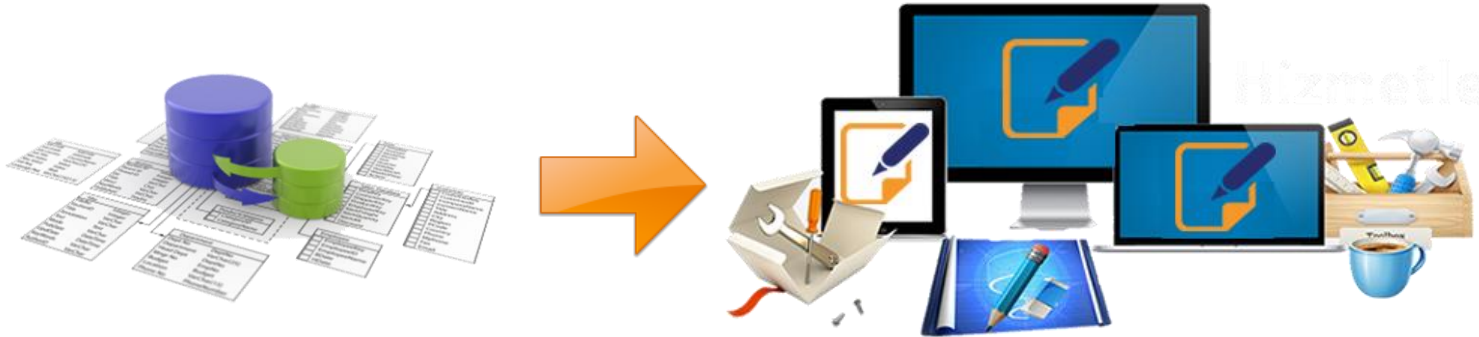
---

- Mantık, bilginin yapısını incelerken, kesin sonuca ulaşmak için doğru ile yanlış arasındaki akıl yürütme ayrımı yapmaktır.
- Bilgisayarlaştırılmak istenen işin mevcut mantığı yazılıma yansıtılmak durumundadır.
- Bu nedenle mantık (algoritma) bileşeni yazılımın en önemli bileşenlerinden biridir.



# Veri (Bilgi)

- Veri (ing. data), işlenmemiş bilgi veya bilginin ham halidir. Bilgi ise, en basit anlamda verinin işlenmiş şeklidir. Her tür yazılım mutlaka bir veri üzerinde çalışmak durumundadır. Veri dış ortamdan alınabileceği gibi, yazılım içerisinde de üretilebilir.
- Yazılımın temel amacı “veri” yi “bilgi” ye dönüştürmektir.



# Belge (Dokümanlar)

---

- Yazılım üretimi bir mühendislik disiplini gerektirir. Mühendislik çalışmalarında izlenen yol ya da kullanılan yaklaşımlar yazılım üretimi için de geçerlidir.
- Yazılım üretimi sırasında, birçok aşamada yapılan ara üretilere ait bilgiler (planlama, analiz, tasarım, gerçekleştirim, vb. bilgileri) belli bir düzende belgelenmelidirler.



# İnsan (Kullanıcı, geliştirici)

---

- İki boyutludur; yazılımı geliştirenler ve kullananlar.
- Günümüzde artık tek kişi ile yazılım geliştirmekten söz edilmemektedir.
- Yazılım üretimi için bir takım oluşturulmakta ve takımın uyumlu çalışabilmesi için çeşitli yöntemler geliştirilmektedir.





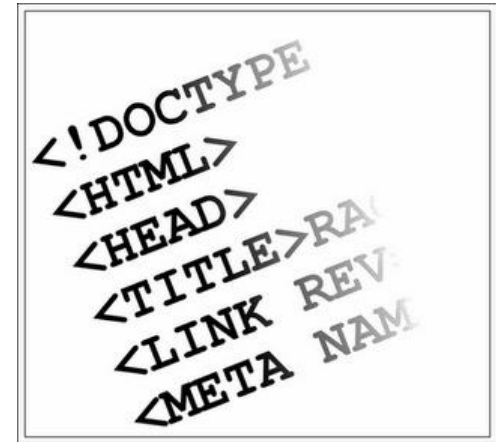
# Program (Kod)

---

- Yazılımın ana çıktısı sonuçta bir bilgisayar programıdır.
- Program işleme alındıktan sonra bakım çalışmaları sürekli olarak gündeme gelir.

## Bunun iki temel nedeni:

- Hiç bir program bütünüyle her olasılık göz önüne alınarak test edilemez.
- İşletmeler doğaları gereği dinamik bir yapıya sahiptir ve zaman içerisinde sürekli olarak yeni istek ve gereksinimler ortaya çıkabilmektedir.



# Yazılım Donanım Karşılaştırması

- Yazılım geliştirilir vs donanım üretilir. (fabrika ortamında seri üretim)
- Donanım bileşenleri dışarıdan temin edilebilir, ancak yazılımı oluşturan parçalar için bu çoğu zaman mümkün değildir (günümüzde “yeniden kullanılabilir yazılım” %1-2).



# Yazılım Donanım Karşılaştırması

---

- Yazılım eskimez.
- Oysa, her donanımın belli bir ömrü vardır. Ömrünü tamamlayan donanım yenisi ile değiştirilir.
- Yazılımın eskimesi ortaya çıkabilecek yeni ihtiyaçları karşılayamaması, kullandığı teknolojinin eskimesi olarak tanımlanabilir.
- Yeni gereksinimler yazılıma ekler yaparak yansıtılır.

# Yazılım Donanım Karşılaştırması

---

- Yazılım en az donanım kadar önemlidir.
- Diyaliz makinelerinde kullanılan yazılımların 2000 yılı uyumsuzluğundan ötürü, bir çok diyaliz makinesi çalışamamış ve böbrek hastaları zor durumda kalmıştır.
- Japonya'da telefon yazılımında ortaya çıkan bir yazılım hatası onbinlerce abonenin saatlerce telefon konuşması yapamamasına neden olmuştur.

# Yazılım Donanım Karşılaştırması

---

- Yazılım kopyalama ve donanım kopyalama farklıdır.
- Hata toleransı amacıyla, hayati olan bir donanımın sistemde bir kopyası daha bulundurulur ve sistemde biri arızalandığında diğeri çalışmayı devralabilir.
- Oysa, bir yazılımı sistemde iki ayrı bilgisayar üzerine kopyalamak oluşabilecek hatalara çözüm olmayacaktır. Belki, sisteme aynı işi yapan iki farklı eş yazılım yüklenmesi çözüm olabilir (**kritik yazılım sistemleri- uçak avionics**).

# Yazılım Üretim Ortamı

- Değişik yetenekte bir çok personel (analist, programcı, test uzmanı, vs.)
- Yazılım çıktısı ile ilgilenen kullanıcılar
- Yeniliğe tepki gösteren kullanıcılar ve yöneticiler
- Yeterince tanımlanmamış kullanıcı beklentileri
- Personel değişim oranının yüksekliği
- Yüksek eğitim maliyetleri
- Dışsal ve içsel kısıtlar (zaman, maliyet, işgücü, vs.)
- Standart ve yöntem eksiklikleri
- Verimsiz kaynak kullanımı
- Mevcut yazılımlardaki kalitesizlik
- Yüksek üretim maliyeti



# Yazılım Mühendisliği

---

## IEEE Tanımı (1993)

“Yazılım Mühendisliği:

Sistemli, düzenli, ölçülebilir bir yaklaşımın yazılım geliştirmede, yazılımın işlenilmesinde ve bakımında uygulanmasıdır.

Diğer bir deyişle mühendisliğin yazılıma uygulanmasıdır.



# Yazılım Mühendisliği

---

- Yazılım üretiminin mühendislik yöntemleriyle yapılmasını öngören ve bu yönde;
- yöntem,
  - araç
  - teknik ve
  - metodolojiler
- üreten bir disiplindir.





# Yazılım Mühendisliği

---

- Yazılım mühendisliği bir yöntemler, teknikler ve araçlar kümesi olarak değerlendirilebilir.
- Yazılım mühendisliğinin hedefi; yazılım üretimindeki karmaşıklıkları gidermektir.
- Geçmişte kullanılan iş akış şemaları gibi yöntemler günümüzde yetersiz kalmaktadır.
- Ayrıca, yazılım üretimi işi tek kişinin başarabileceği boyuttan çıkmış ve bir takım iş biçimine dönüşmüştür.

# Yazılım Mühendisi

---

- Yazılım Mühendisliği İşini yapan kişidir.
- Temel hedefi; **üretimin en az maliyet ve en yüksek nitelikte yapılmasını sağlamaktır.**
- Programcı değildir. Ancak programcının tüm yeteneklerine sahiptir.
- Yazılımın daha çok mantıksal boyutuyla ilgilenir ve işi insanlarla ilişkiyi gerektirir.
- Sistem analisti de değildir. Farkı; analist sadece sistemin analiz aşaması ile ilgilenirken, yazılım mühendisi tüm aşamaların içindedir.

# Yazılım Hataları

---

- Yazılım hataları, yazılım yaşam döngüsünde çok önemli yer tutan unsurlardan biridir.
- Yazılım geliştirme de karşılaşılan en sıkıcı durumdur.
- Hatalar yüzünden yazılım maliyeti artmaktadır.
- Yazılım geliştirme sürecini uzatmaktadır.
- Çözümü erken bulunmayan hatalar bazen uyulması gereken sistemi zor durumda bırakarak zamansal problemler oluşturmaktadır



# Yazılımların Sınanması

---

- Bir programı tüm ayrıntıları ile test etmek teorik olarak mümkün olmakla birlikte, uygulamada bu mümkün değildir.
- Yazılım ancak sınırlı sayıda veri ile sınanabilir.

Mantıksal Tasarım	%20
İşlevsel Tasarım	%15
Kodlama	%30

# Hataların “Yayılma” Özelliği

## ➤ Yazılımda Hata Düzeltme Maliyetleri:

- Yazılım üretimindeki hatalar yayılma özelliği gösterir.
- Bu nedenle, hata düzeltme maliyetleri ilerleyen aşamalarda giderek artar.

<b>Analiz</b>	1
<b>Tasarım</b>	5
<b>Kodlama</b>	10
<b>Test</b>	25
<b>Kabul Testi</b>	50
<b>İşletim</b>	100



# Yazılım Maliyetleri

- Günümüzde yazılım maliyetlerindeki artışlar giderek artmaktadır. "Yazılımımızı alırsanız yanında donanımı ücretsiz olarak sağlayacağız" deyişi zaman içerisinde giderek doğrulanmaktadır.
- Örneğin, günümüzde bir kopyası yüz bin dolar dolayında satılan kurumsal kaynak planlama yazılımlarının bulunduğu gözlemlenmektedir.
- Öte yandan bir kişisel bilgisayar ise 1000 ABD dolarının altında satılmaktadır.
- Yazılımın kopyalanma maliyeti ile donanım kopyalama maliyetinin arasındaki farklılık dikkate alındığında, yazılım maliyetlerinin, donanım maliyetlerine oranla oldukça yüksek olduğu ortaya çıkar.



# Yazılım Sistemlerin Sınıflandırılması

---

- İşlevlerine göre sınıflandırma
- Zamana dayalı özelliklere göre sınıflandırma
- Boyuta göre sınıflandırma



# İşlevlerine Göre Sınıflandırma

---

<b>Hesaplama</b>	Mühendislik Çözümleme
<b>Veri İşleme</b>	Bankacılık
<b>Süreç Temelli</b>	Gömülü Sistemler
<b>Kural Temelli</b>	Robotik, Yapay Zekâ
<b>CAD</b>	Sinyal İşleme



# Zamana Dayalı Özelliklere Göre Sınıflandırma

Toplu (Çevrim-Dışı)	Çevrim-İçi
Gerçek Zamanlı	

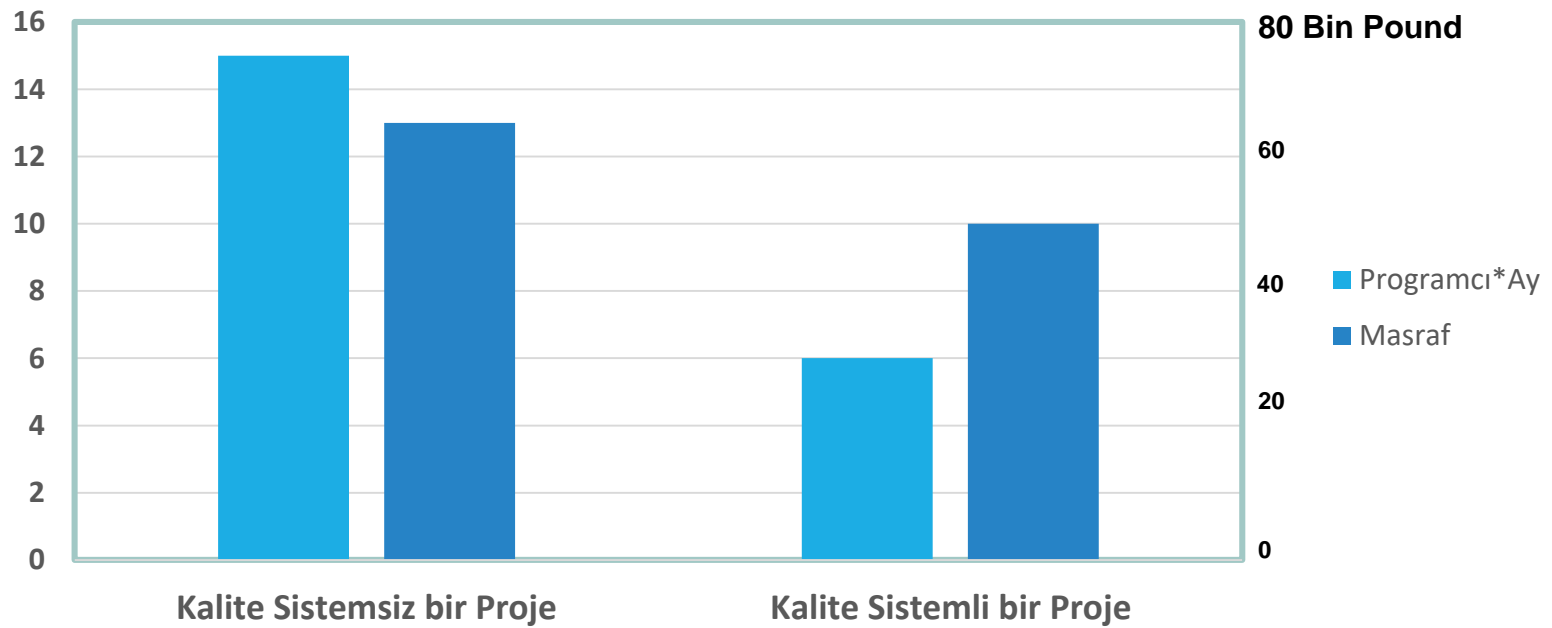


# Boyuta Göre Sınıflandırma

---

<b>Küçük (<math>SS &lt; 2000</math>)</b>	PC Oyunları Öğrenci Projeleri
<b>Orta (<math>2000 &lt; SS &lt; 100,000</math>)</b>	CAD BDE Yazılımları
<b>Büyük (<math>100,000 &lt; SS &lt; 1 \text{ Milyon}</math>)</b>	İşletim Sistemleri
<b>Çok Büyük (<math>SS &gt; 1 \text{ Milyon}</math>)</b>	Komuta Kontrol Sistemleri Hava Tahmini Sistemleri Yıldız Savaşları Sistemleri

# Yazılımda Kalite



# Yazılımda Kalite

---

➤ Üretim Süreci Boyunca ara ürünlere ilişkin kalite standartlarının geliştirilmesi ve geliştirme işlemlerinin bu standartlara uygunluğunun denetlenmesidir.

Yazılım kalite sağlama etkinlikleriyle;

- Yazılım maliyetleri düşürülür,
- Yazılım üretiminin yönetimi kolaylaşır,
- Belgeleme ve standart sorunları giderilir.



# Yazılımda Kalite

---

Ekonomi	Tamlık	Yeniden Kullanılabilirlik	Etkinlik	Bütünlük
Güvenirlik	Modülerlik	Belgeleme	Kullanılabilirlik	Temizlik
Değiştirilebilirlik	Geçerlik	Esneklik	Genellik	Sınanabilirlik
Taşınabilirlik	Bakılabilirlik	Anlaşılabilirlik	Birlikte Çalışabilirlik	

# Özet

---

- **Yazılım**= Mantık, veri, belge, insan ve program bileşenlerinin, belirli bir üretim amacına yönelik olarak bir araya getirilmesi, yönetilebilmesi için kullanılabilecek ve üretilen, yöntem, araç, bilgi ve belgelerin tümünü içerir.
- **Yazılım Mühendisliği**: Sistemli, düzenli, ölçülebilir bir yaklaşımın yazılım geliştirmede, yazılımın işlenilmesinde ve bakımında uygulanmasıdır. Diğer bir deyişle mühendisliğin yazılıma uygulanmasıdır.
- **Yazılım mühendisliğinin hedefi**; yazılım üretimindeki karmaşıklıkları gidermektir.
- **Yazılım kalite sağlama etkinlikleriyle**;
  - Yazılım maliyetleri düşürülür,
  - Yazılım üretiminin yönetimi kolaylaşır,
  - Belgeleme ve standart sorunları giderilir.

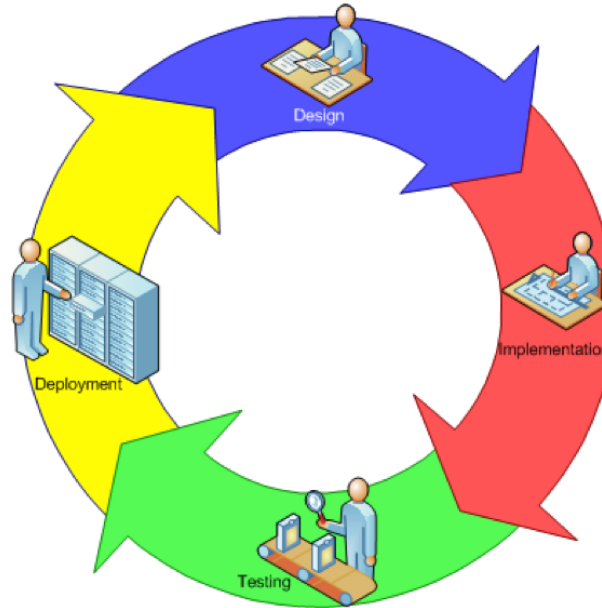
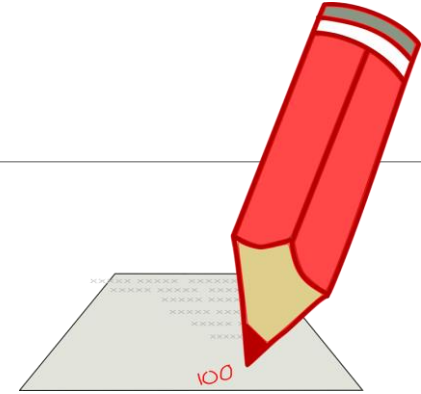
# Sorular

---

1. Yazılım ile program arasındaki farklılığı belirtiniz.
2. Yazılım ile donanım arasındaki farklılıkları belirtiniz.
3. Neden yazılım giderek pahalılaşırken, donanım ucuzlamaktadır?
4. Yazılım mühendisliği ile diğer mühendislik disiplinlerini karşılaştırınız.
5. Bu bölümde verilen yazılım sınıflandırmasını dikkate alarak, her sınıflandırma için bir yazılım örneği veriniz.
6. Yazılım mühendisi ile programcı arasındaki farklılığı belirtiniz.
7. Yazılım Hatalarında Yayılma neyi ifade etmektedir?
8. Yazılım Maliyetinde hangi Sınıflandırma ağır basmaktadır?
9. İşlevlerine Göre Sınıflandırma da Kural Temelli 'ye Örnek veriniz.
10. Yazılım Kalitesini Tanımlayınız.

# Ödev

1. Yazılım Yaşam Döngüsünü Araştırınız.
2. Yazılım Süreç Modellerini Araştırınız.





# 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 Saridoğan, 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

<http://blog.alisuleymantopuz.com/2014/08/30/yazilim-mimarisi-ve-tasarimi-nedir/> \*

<http://www.akifsahman.com/?p=175>

<https://ece.uwaterloo.ca/~se464/08ST/index.php?src=lecture> \* <http://info.psu.edu.sa/psu/cis/azarrad/se505.htm>

<http://www.metinakbulut.com/YAZILIM-MIMARISI/> \* [http://ceng.gazi.edu.tr/~hkaracan/source/YPY\\_H3.pdf](http://ceng.gazi.edu.tr/~hkaracan/source/YPY_H3.pdf)

<http://iiscs.wssu.edu/drupal/node/3399> \* <http://www.cs.toronto.edu/~sme/CSC340F/slides/21-architecture.pdf>

<http://www.users.abo.fi/lpetre/SA10/> \* <http://sulc3.com/model.html>

<http://salyangoz.com.tr/blog/2013/11/23/digerleri/yazilim-gelistirme-surec-modelleri-3/>

# Sorularınız?

---

