



# Bilgisayar Mühendisliğine Giriş

Yrd.Doç.Dr.Hacer KARACAN

# Yazılım Mühendisliği

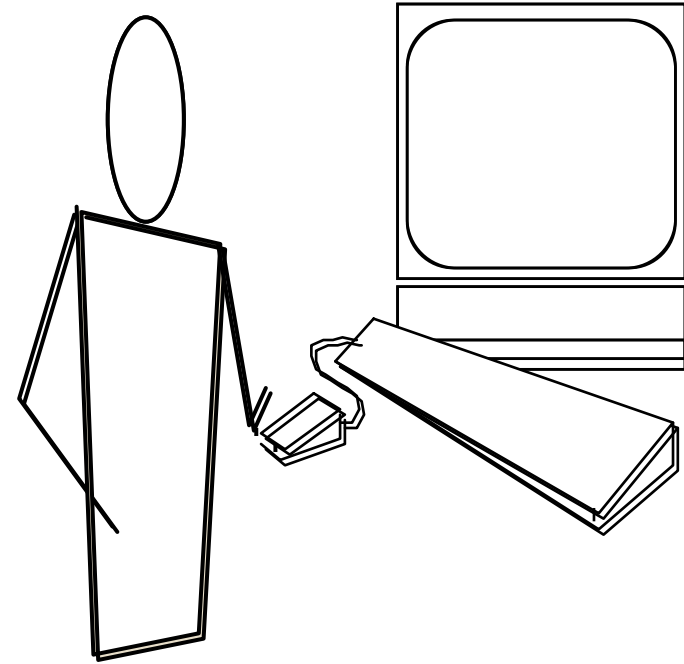
- Temel Kavramlar
- Yazılım Özellikleri
- Genel Yazılım Süreci Çerçevesi
- Yazılımda pratik prensipler

# Temel Kavramlar

- Yazılım
  - Yazılım mühendisleri tarafından tasarlanır ve geliştirilir
  - Toplumdaki herkes tarafından kullanılabilir
    - Ticaret, kültür ve günlük yaşantılarımızda yaygın olarak kullanılmaktadır
- Yazılım mühendisleri
  - İnsanlara zarar vermeyen yazılımlar geliştirme zorunlulukları vardır
- Yazılım kullanıcıları
  - Sadece yazılım ürünlerinin ihtiyaçlarını karşılamaları ve görevlerini kolaylaştırmaları ile ilgilenirler

# Yazılım –Bilgisayarların ilk yılları

- Oldukça küçük programlar
- Tek kişinin yazdığı programlar
- Sadece alan uzmanlarının geliştirip yine kendilerinin kullandığı programlar
- Bazı programlama dillerinde bilinen algoritmaların kullanım eğilimi



# Yazılım - Günümüz

- Programlar
  - Oldukça büyük ve ***karmaşık***
  - Uzun süreler zarfında birbirleriyle işbirliği içinde çalışan takımlar tarafından geliştiriliyorlar
  - Geliştiriciler artık geliştirilen yazılımın son kullanıcısı değiller
    - ➔ Sistemin asıl kullanıcıların alanla ilgili uzman bilgileri yok

# Bilgisayar Programı nedir?

- Bilgisayarların veri işlemek, bir işlemi gerçekleştirme ya da mantıksal bir problemi çözmek için takip ettikleri; bir programlama dilinde (C, C++, Java,...) yazılmış komutlar grubu

# Yazılım nedir?

- Bilgisayar programları
- Bu programları kurmak için kullanılan yapılandırma (configuration) dosyaları
- Programın nasıl kullanılacağını anlatan kullanıcı dokümanları
- Destek hizmetleri
- Yazılımın yapısını anlatan sistem dokümanları

***Bilgisayar Programları***  
***(bazen birden fazla farklı program)***

**+**

***İlgili dokümanlar***

**+**

***Yapılandırma dosyaları***

# Yazılımın ikili rolü

- Ürün olarak yazılım
  - Programlama potansiyeli sunar
  - Bilgi üretir, yönetir, edinir, değiştirir, görüntüler ya da iletir
- Ürün sunmak için bir araç olarak yazılım
  - Sistem fonksiyonelliğini direk olarak sağlar ya da destekler
  - Diğer programları kontrol eder (örn. işletim sistemleri)
  - İletişim sağlar (örn. ağ yazılımları)
  - Başka yazılımlar geliştirmeyi sağlar (örn. yazılım araçları)



# Yazılım Çeşitleri

- **Genel**

- Pek çok farklı müşteriye satılmak üzere üretilmiş hazır ticari ürünler (Commercial Off The Shelf – COTS)

- **İsmarlama/İsteğe özelleştirilmiş**

- Tek bir müşteri için onun belirtileri (ihtiyaçları) doğrultusunda hazırlanmış

# Kompleks yazılımlara gerçekten ihtiyacımız var mı?

- Neil Armstrong un Apollo uzay aracı ile aya ilk ayak basışı insanoğlu için çok önemli bir adımdır.
- Uzaya giden Apollo uzay mekiği 74KB bellek, 4KB RAM ve sabit diski bulunmayan bir bilgisayar ile bu görevi gerçekleştirmiştir.  
→ Bu çok şaşırtıcı değil mi?
- Bu da bize gösteriyor ki pek çok şeyi aslında yazılım olmadan da gerçekleştirebiliriz !???

# Kompleks yazılımlara gerçekten ihtiyacımız var mı?

- Bilimsel ve teknolojik merak (kuşkuculuk-skepticism) insanlar için yemek kadar vazgeçilmez bir ihtiyaç.
- ➔ Böyle olduğu sürece de yazılımla kontrol edilen sistemler gün geçtikçe artmaya devam edecek.

# Yazılım Uygulama Alanları

- Sistem yazılımları
- Uygulama yazılımları
- Mühendislik ya da bilimsel yazılımlar
- Gömülü yazılımlar
- Ürün hattı yazılımları (eğlence yazılımları da dahil)
- Web-uygulamaları
- Yapay zeka yazılımları

# Yeni kategoriler

- Kablosuz Ağ Yazılımları
- Netsourcing—the Web as a computing engine
- Açık Kaynak Yazılımlar
- Ayrıca ...
  - Veri Madenciliği
  - Grid computing
  - Bilişsel Makineler
  - Nanoteknoloji yazılımları

# Mühendislik nedir?

- TANIM: Doğadaki maddenin ve enerji kaynaklarının insanların kullanımı için yararlı hale getirilmesi için bilimsel ve matematiksel prensiplerin **uygulanmasıdır**.
- Mühendisler
  - Uygun olan yerlerde **teori** + **metot** + **araçları** uygulayarak işlerin yürümesini sağlarlar.
  - Çeşitli **kısıtlamalar** içerisinde çözümler bulmaya çalışırlar.

# Mühendislik nedir?

- Mühendislik aktivitelerinin prensipleri
    - Tüm projeler
      - Umulan/önceden tahmin edilen bütçe
        - ➔ MALİYET
      - Umulan/önceden tahmin edilen zaman çizelgesi
        - ➔ ZAMAN
      - Müşterinin gereksinim/isterlerine uygun ,
        - ➔ KALİTE
- şekilde tamamlanması gerekmektedir.

# Maliyet + Zaman + Kalite





# Yazılım Mühendisliği Nedir?

- “Yazılım mühendisliği **bilimsel bilginin** bilgisayar programlarının tasarımı ve oluşturulması için pratik uygulaması ve onları geliştirme, **çalıştırma** ve **devam ettirmeye** (operate and maintain) ilgili **belgelerdir**.”

[Boehm, 1976].

- Yazılım geliştirmek, çalıştırma ve devam ettirmek için sistematik disiplinli ölçülebilir yaklaşımın uygulanması → İşte bu yazılıma mühendisliğin uygulanmasıdır.

[IEEEComputer Society, 1990].

Bilgisayar profesyonelleri için dünyanın önde gelen organizasyonu

→ Institute of Electrical and Electronics Engineering (IEEE) Computer Society  
= Elektrik ve Elektronik Mühendisleri Enstitüsü Bilgisayar Topluluğu

<http://www.ieee.org>

# Yazılım Mühendisliği Nedir?

- Yazılım mühendisliği yazılım üretimi ile ilgili tüm durumlarla ilgilenen bir mühendislik bilim dalıdır.
- Yazılım mühendisleri
  - İşlerinde sistematik ve organize yaklaşımlar benimsemelidirler.
  - Çözmek istedikleri probleme, geliştirme kısıtlamalarına ve de mevcut kaynaklara uygun araç ve teknikleri kullanmalıdırlar.

# SİSTEM

- Ne için mühendislik yapıyoruz?
  - ➔ Bilgisayar sistemleri
- SİSTEM
  - Tanım: Bir amacı ya da bir grup amacı gerçekleştirmek için veri toplayan, işleyen veya üreten bir grup öğedir



# BS nedir?

- BS:
  - insan,
  - veri,
  - süreçler,
  - bilginin sunumu ve
  - iş ortamındaki günlük işlemlerin/ operasyonlar ile yöneticiler ve kullanıcıların kara verme ihtiyaçları ve problem çözmelerini destekleyen ***bilgi teknolojilerinin*** düzenlenmesidir.

# BT nedir?

- Bilgisayar teknolojisinin (donanım ve yazılım) telekomünikasyon / iletişim teknolojisi ile birleşimini tanımlayan güncel bir terimdir.

# Kavramlar arasındaki ilişki

- Aşağıdaki kavramları tipik bir sistem içerisinde öneme ve boyuta göre sıraya koyunuz:
  - Donanım
  - Yazılım
  - BS (Bilgi Sistemi/ Information System)
  - BT (Bilgi Teknolojileri / Information Technology)

# Kavramlar arasındaki ilişki

- $BS \gg BT$
- $BT = \text{Donanım} + \text{Yazılım}$
- $BS = \text{İnsan} + \text{Veri} + \text{Süreçler} + \text{Bilginin Sunumu} + \underbrace{\text{Donanım} + \text{Yazılım}}_{BT}$

# Yazılım Mühendisliği gerçekten önemli mi?

- TÜM gelişmiş ulusların ekonomiler günümüzde yazılıma dayanmaktadır.
- Gün geçtikçe yazılım ile kontrol edilen sistemler artış göstermektedir
- Yazılım mühendisliği harcamaları ülkelerin gayri safi milli hasıllarının kayda değer bir bölümünü oluşturmaktadır.



# Yazılım Mühendisliği gerçekten önemli mi?

- Yazılım maliyetleri sistem maliyetlerinin büyük kısmını oluşturmakta.
- Bilgisayar üzerinde çalışacak yazılımın maliyeti donanımın maliyetinden genellikle daha fazla
- Yazılımın sürdürülebilirlik maliyeti geliştirme maliyetinden daha fazla.
  - Uzun süreli kullanılacak sistemler için, sürdürülebilirlik maliyetleri geliştirme maliyetlerinin birkaç katı olabilir
- Yazılım mühendisliği maliyet-etkin yazılımlar geliştirmekle ilgilidir.

# Yazılım Mühendisi

- Bir kodlayıcı, yani programlayıcı değildir.
- Yazılım mühendisliği disiplini uygulayarak yazılım geliştiren kişidir.
- Herhangi bir programlama dilini bilen bir kişi programcı olabilir ama eğitimi almadan yazılım mühendisliği işini yapamaz.
  - Salt kodlayıcı değil ama kod yazma tekniklerini çok iyi bilir
  - İyi bir belge düzenleyici olmayabilir ama çok iyi gözden geçiricidir
  - Uygulama alanında az bilgisi olabilir fakat kullanıcı isteklerini nasıl aktarabileceğini bilir

# İnşaat Mühendisliği vs. Yazılım Mühendisliği

## • Köprüler

- Devamlı (Continuous)
  - Matematik (Calculus)
  - Sınanması ve çözümlenmesi kolay: Eğer köprü  $1\text{ M kg}$  taşıyabilirse  $0,99\text{ M kg}$  da taşıyabilir.



## □ Yazılım

- ▣ Devamsız (Discrete)
  - Mantık, Soyut matematik
  - Sınanması ve çözümlenmesi zor



# İnşaat Mühendisliği vs. Yazılım Mühendisliği

## • Köprüler

- Fiziksel maddelerden yapılırlar
  - Bazı maliyetler aşıkardır
  - Yapım aşamasından sonra değişiklik yapmak zordur.



## □ Yazılım

- ▣ Sanal maddelerden yapılırlar
  - Tüm maliyetler aşıkardır değildir
  - Değişiklik yapmak kolay olabilir( Aslında değildir.)

```
for (int i = 0; i < rows; i++) {  
    for (int j = 0; j < columns; j++) {  
        nextStates [i][j] =  
            getCellAt (i, j).getNextState ();  
    }  
}
```

# İnşaat Mühendisliği vs. Yazılım Mühendisliği

## ● Köprüler

- Eğer çökerlerse yapımı gerçekleştirenlerin mahkeme edilecekleri aşikardır.
- Geliştiricilerin ehliyete ihtiyacı vardır



## □ Yazılım

- Eğer çökerlerse yazılım sağlayıcılar kullanıcıları suçlar ve yükseltmeler için masraf keserler.
- Herkes yazılım üretebilir, kimse mahkeme edilmez.



# İnşaat Mühendisliği vs. Yazılım Mühendisliği

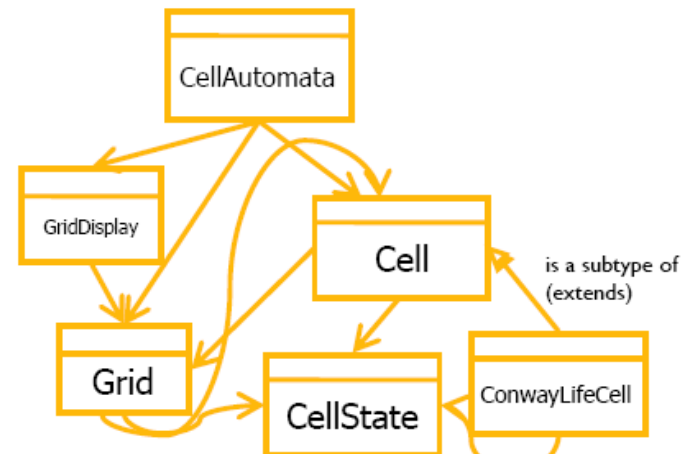
## • Köprüler

- İsterler (genellikle) aşık ve tanımlanmaları kolaydır.
- İyi bir tasarım herkes için hemen aşıkardır.



## □ Yazılım

- İsterler anlaşılmaz (gizemli:) ve tanımlanması zordur.
- İyi bir tasarım ancak uzmanlar için açıktır, etkileri de daha sonradır.



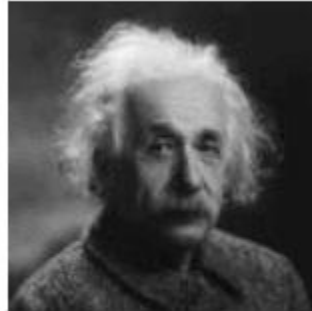
# Yazılım Özellikleri

1. Yazılım **geliştirilir (developed)**, klasik anlamda imal (manufactured) edilmez.



HAM MADDE

+



İNSAN YARATICILIĞI

=



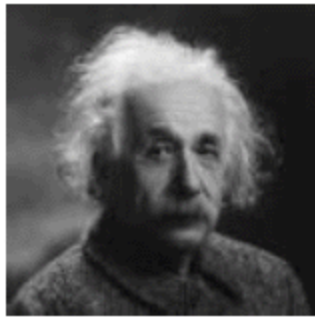
FİZİKSEL ÖĞE

örn. \$5



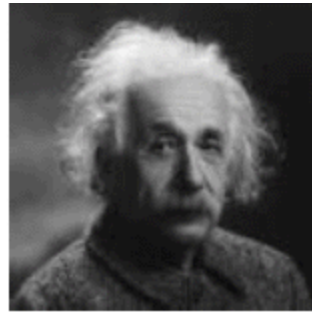
# Yazılım Özellikleri

1. Yazılım **geliştirilir (developed)**, klasik anlamda imal (manufactured) edilmez.



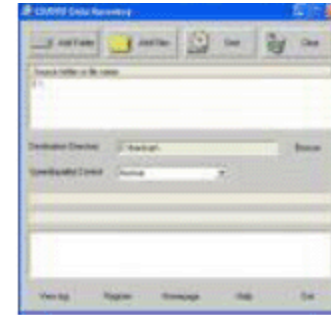
İNSAN YARATICILIĞI

+



İNSAN YARATICILIĞI

=



MANTIKSAL ÖĞE  
örn. \$5M

+



FİZİKSEL ÖĞE  
örn. \$0,05

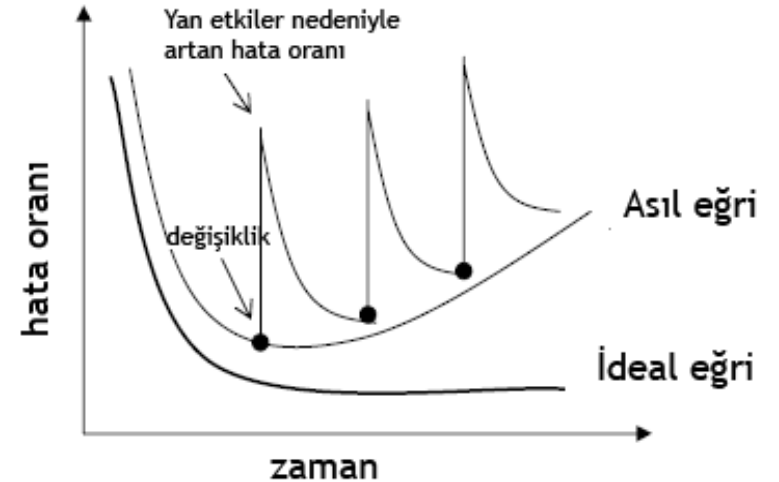


# Yazılım Özellikleri

2. Yazılım aşınmaz / yıpranmaz, fakat yapılan değişiklikler nedeniyle geriler.
  - Pek çok değişiklikten sonra baştan tasarlanması gerekebilir



**DONANIM**



**YAZILIM**

# Yazılım Özellikleri

3. Endüstri bileşen tabanlı geliştirmeye yönelmiş olsa bile, halen yazılımlar özel geliştirilmektedir.



```
for (int i = 0; i < rows; i++) {  
    for (int j = 0; j < columns; j++) {  
        nextStates [i][j] =  
            getCellAt (i, j).getNextState ();  
    }  
}
```

# İyi yazılımın özellikleri

- Yazılım kullanıcısına istenen işlevleri ve performansı sunmalı ve buna ek olarak:
  - Sürdürülebilirlik (maintainable)
    - Yazılım değişen ihtiyaçlara göre gelişebilmelidir.
  - Güvenilebilirlik (dependable)
    - Yazılım güvenilir olmalıdır
  - Verimlilik (efficiency)
    - Yazılım sistem kaynaklarını boşa harcamamalıdır.
  - Kullanılabilirlik (usability)
    - Sistem tasarlandığı kullanıcıları için kolay kullanılabilir olmalıdır.

# Yazılım Mühendisliği Yöntemleri

- **Yöntem Tanımları**
  - Grafik yöntem tanımları (örn. Veri Akış Şeması-DFD, Varlık İlişki Şeması –ER)
- **Kurallar**
  - Sistem yöntemlerine uygulanan kısıtlamalar (Sistem yönteminde yer alan her varlığın benzersiz bir adı olmalıdır)
- **Tavsiyeler**
  - İyi tasarım pratikleri için tavsiyeleri içerir. (örn. Tasarım örüntüleri, ilişki/sınıfların max sayısı)
- **Süreç Kılavuzları**
  - Takip edilecek aktiviteler (örn. İsterlerde yapılacak herhangi bir değişiklik SİB (SRS) belgesinde güncellenmelidir.

# Genel yazılım süreci çerçevesi

- İletişim
  - müşteri işbirliği ve gereksinim toplama
- Planlama
  - Mühendislik iş planını oluşturur, teknik riskleri tanımlar, gerekli kaynakları listeler, iş ürünleri üretilir ve iş zaman planı tanımlanır
- Modelleme
  - Geliştirici ve müşterilerin yazılım gerekleri ve yazılım tasarımını anlamasına yardımcı olacak modellerin oluşturulması
- Yapım
  - Kod oluşturma ve test
- Kurulum
  - Müşteriye yazılımın değerlendirme ve geri bildirim için teslimatı

# Yazılım Süreci

- Görevler arasındaki genel akış ve bağımlılık seviyeleri
  - Problemi anlama (iletişim ve analiz)
  - Çözüm planlama (yazılım tasarımı)
  - Planı yürütme (kod üretme)
  - Doğruluk için sonuçları inceleme (test ve kalite güvencesi)

# Yazılımda pratik prensipler

- Yazılım kullanıcılarını değer sağlamak için vardır
- Keep it simple stupid (KISS)
- Açık bir vizyon herhangi bir yazılım projesi için gereklidir
- Her zaman başka birinin işlerini ona göre yürüteceğiniz düşünerek tanımla, tasarla ve uygula
- Gelecekteki değişikliklere açık ol
- Önceden yeniden kullanım için planlamak maliyeti düşürür ve yeniden kullanılan bileşenlerin ve onu gerektiren sistemin değerini artırır

# Konular

- Yazılım Mühendisliği Yöntembilimleri
- Yazılım Süreçleri
- Yazılım İsterleri
- Modelleme
- Prototipleme
- Yazılım Tasarımı ve Gösterimi
- Kullanıcı Arayüzü tasarımı
- Yazılım Testi
- Yazılım Proje Yönetimi
- Yazılım Nitelik Güvencesi
- Yazılım Süreç İyileştirme