YMH 114 YAZILIM MÜHENDİSLİĞİNİN TEMELLERİ

Konular

- Ders içeriği, başarı değerlendirme
- YM kavramları, süreçler, modeller, yöntemler
- Yazılım Gereksinim Çözümlemesi
- Yazılım Tasarımı
- Yazılım Gerçekleştirimi
- Test ve Bakım
- Yazılım Geliştirme Yöntem Bilimleri
- Risk, Kalite ve Proje Yönetimi
- UML

Genel

- Ders Kitabı: Yazılım Mühendisliği

  Erhan Sarıdoğan- papatya Yayıncılık (kitapyurdu.com)

  Diğer Kaynaklar:
  - Ders Notları.
  - Ali Arifoğlu, Yazılım Mühendisliği. SAS bilişim Yayınları
  - İnternet, UML Kaynakları
  - Roger S. Pressman, Software Engineering – Practitioner's Approach

Araştırma ve Proje

- Araştırma ve Sunum

(Gereksinim, Tasarım, Programlama (Kodlama/Hata Ayıklayıcı) (Kullanıcı Önyüz), Test, ProjeYönetimi, Düzenleşim (Konfigürasyon)/Değişim Yönetimi konularında kullanılan araç ve gereçleri araştır)

- Proje

(Bir konu üzerinde yazılım mühendisliği aşamalarının UML kullanılarak uygulanması)

Kullanılacak Yazılımlar

- Proje raporlama ve dökümantasyon işlemlerinde ihtiyaç duyulabilecek bazı araçlar şöyledir;
- MS Office Word, (proje raporu yazımı)
- MS Office Excel (Tablolama vb.)
- Adobe Photoshop (şekil çizimlerinde vb.)
- IBM RSA (Retional Software Architect) (UML diyagram çizimlerinde)
- MS Visio (Akış şeması çizimlerinde vb.)

Yazılım Nedir

Yazılım
- Tanımlanmış bir işlevi yerine getiren,
- Girdi ve Çıktıları olan,
- Herhangi bir donanım üzerinde çalışan,
- Bilgisayar programı veya programlarından ve
- Kullanım ve bakım kılavuzları gibi belgelerden oluşan

bir üründür.

Yazılım Mühendisliği (YM) - Nedir

"Mühendislik eylemlerinin, (Geliştirme, İşletme, ve Bakım), disiplinli, sistematik ve nicelikli bir şekilde yazılıma uygulanması"

YM – Önemi

- Yazılımın hayatımıza girmediği yer var mı?
- Yazılımsız hayat nasıl olurdu?
- Yazılım ve Eğitim
- Yazılım ve Ekonomi
- Yazılım ve Haberleşme
- Yazılımın verimliliğe katkısı
- Yazılımın kültüre etkisi

YM – Tarihçesi

- İlk Bilgisayarlar ve Makine ve Assembly Dili
- İşletim Sistemleri ve Anabilgisayarlar (Mainframe),
- Kart okuyucuları
- DOS ve PC'ler
- Derleyici (Compiler) ve Yorumlayıcılar (Interpreter) ve Yeni nesil yazılım dilleri
- Windows
- Yarı İletken teknolojisinin fiyat ve boyutlara etkisi
- Veri haberleşmesindeki gelişmeler
- **Internet**

Yazılım Geliştirme İstatistikleri

- Tipik yazılım projesinin geliştirilmesi 1-2 yıl sürüyor ve en azından 500.000 kod satır içeriyor
- Tüm projelerin yalnız %70-80'i başarıyla tamamlanıyor
- Tüm geliştirme sürecinde her birey günde ortalama 10 satırdan az kod yazıyor
- Geliştirme süresince her 1000 kaynak kod satırında 50-60 hata bulunuyor (satışa sunulmuş sistemde hata sayısı 4/1000'e düşüyor)

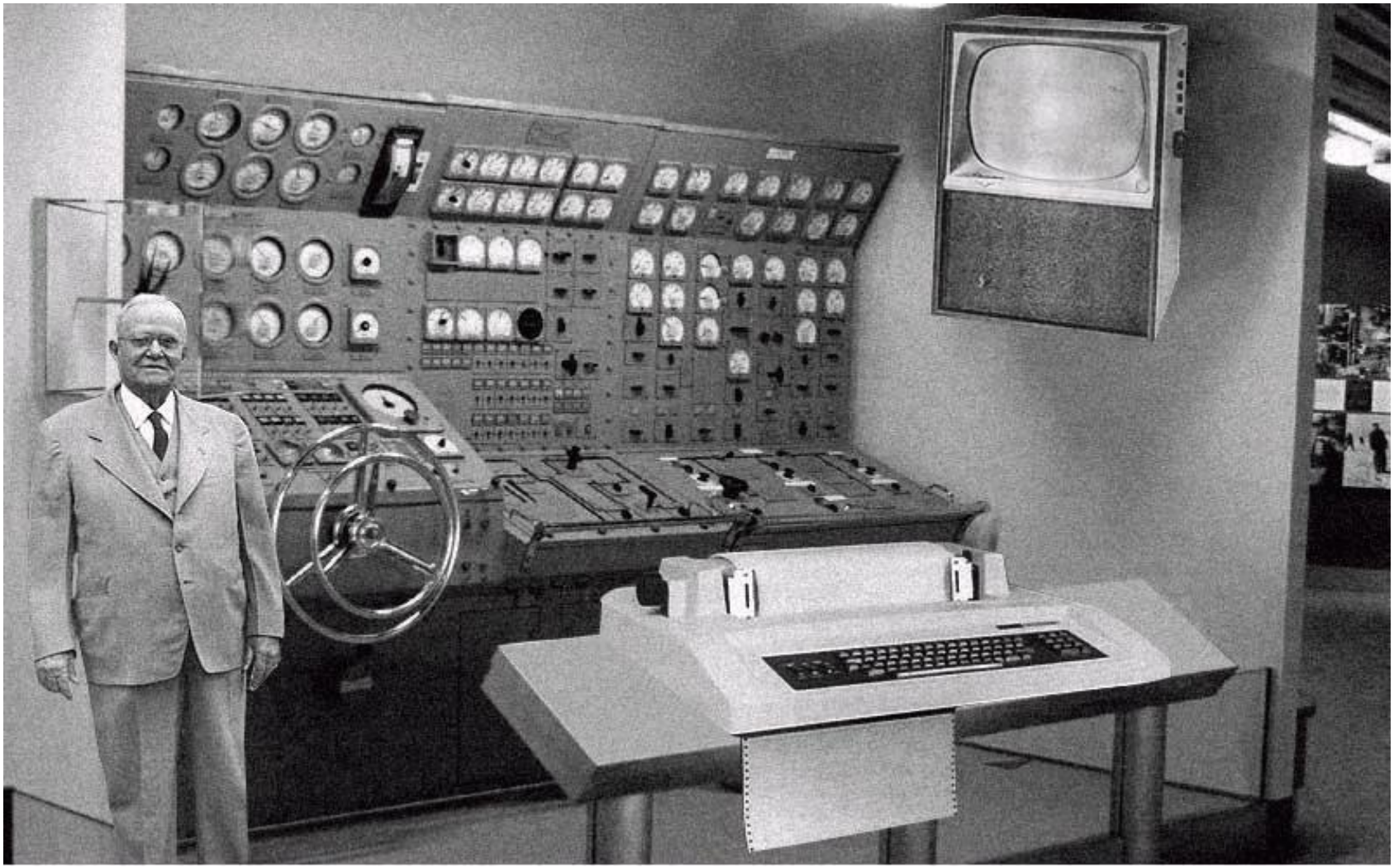
Yazılım Problemleri (Krizleri)

- Tasarlanan zamanın gerisinde kalma
- Bütçeyi aşma
- Düşük Kalite
  - Güvenilir olmayan yazılım
  - Kullanıcı taleplerinin karşılanmasında yetersizlik
  - Sürekliliğinin sağlanmasındaki zorluk

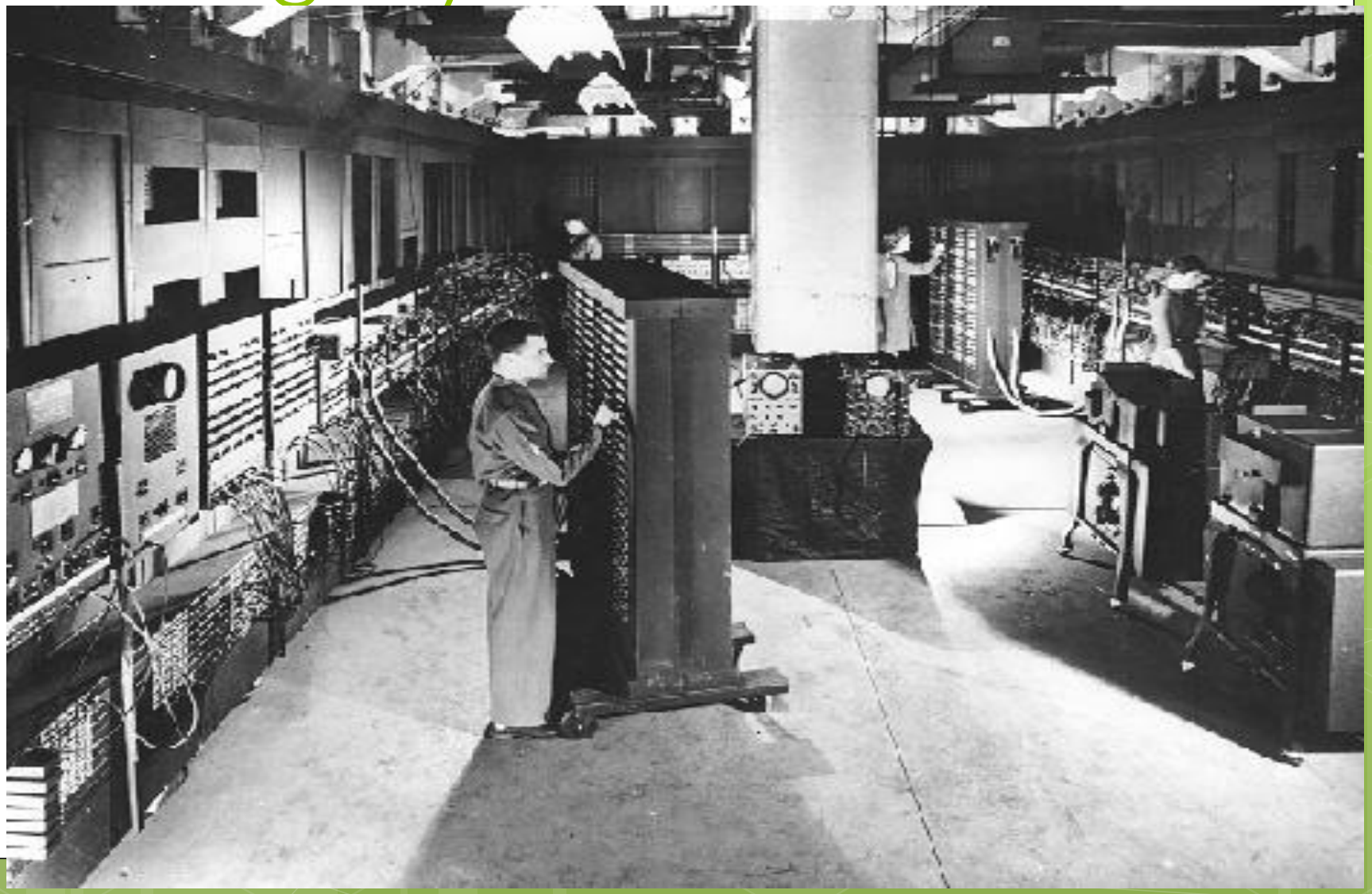
Yazılım-Donanım Evrimi

- Erken Yıllar 1950-1960 ilk bilgisayarlar, makine dili, 3-4 şirket
- Yazılımlar; kullanıcı ile birebir iletişimde bulunmayan, işlerin toplu olarak verilip, yalnızca yazıcı çıktılarının alındığı biçimde geliştirilmekte idi.
- Ayrıca, yazılımlar bu günkü gibi ürün tarzında değil, kuruluşa özel olarak geliştirilmekte idi.

İlk Bilgisayarlar



İlk Bilgisayarlar



Yazılım-Donanım Evrimi

- İkinci Dönem 1960-1975 Ana çatı, çok kullanıcılı sistemler,
  Veri tabanı yönetimi
- Çok kullanıcılı, gerçek zamanlı yazılımlar üretilmeye başlandı.
- VTYS önce yapılandırılmış dosya sistemleri ile ortaya çıktı.
- Yavaş yavaş ürün türü yazılımlar ortaya çıkmaya başladı.

Yazılım-Donanım Evrimi

- Üçüncü Dönem 1975-1990 Süper bilgisayarlar, Kişisel bilgisayarlar
- Açık sistem mimarisinin tanıtıldığı bu dönemde ürün bazlı yazılımlar oldukça yaygınlaştı.
- Kişisel bilgisayarlar yaygınlaşarak evlere girmeye başladı.
- Ağ yapısının gelişmesi ve güçlenmesiyle, dağıtılmış yazılım sistemleri geliştirilmeye başlandı.
- Yapay zeka teknolojisinin gelişmesiyle "akıllı uygulama yazılımları" üretilmeye başlandı

Yazılım-Donanım Evrimi

- Dördüncü Dönem 1990 - Yapay Zeka, Gömülü Sistemler, Paralel Sistemler, Yazılım Kaliteleri
- Uzman sistem yazılımları oldukça gelişmiş ve mikro-bilgisayarlar üzerinde yaygınlaşmıştır.
- "Yazılımda Kalite" olgusu önem kazanmış ve yazılım ile ilgili standartlar olgunlaşmaya başlamıştır.
- Yazılım üretimi ve ürünlerinin değerlendirilmesi amacıyla kurumlar oluşmaya başlamıştır.

Programlama Dillerinin Seviyeleri

1.Kuşak **Makine Dili**

10101110 10010001

2. Kuşak **Assembler**

8085, Z80, 68000, vs

3.Kuşak **Üst Seviye Diller**

Pascal Coral66 Basic

Bildirimsel

LISP Hope Prolog

Nesneye Yönelik Diller

Smalltalk, C++, Java

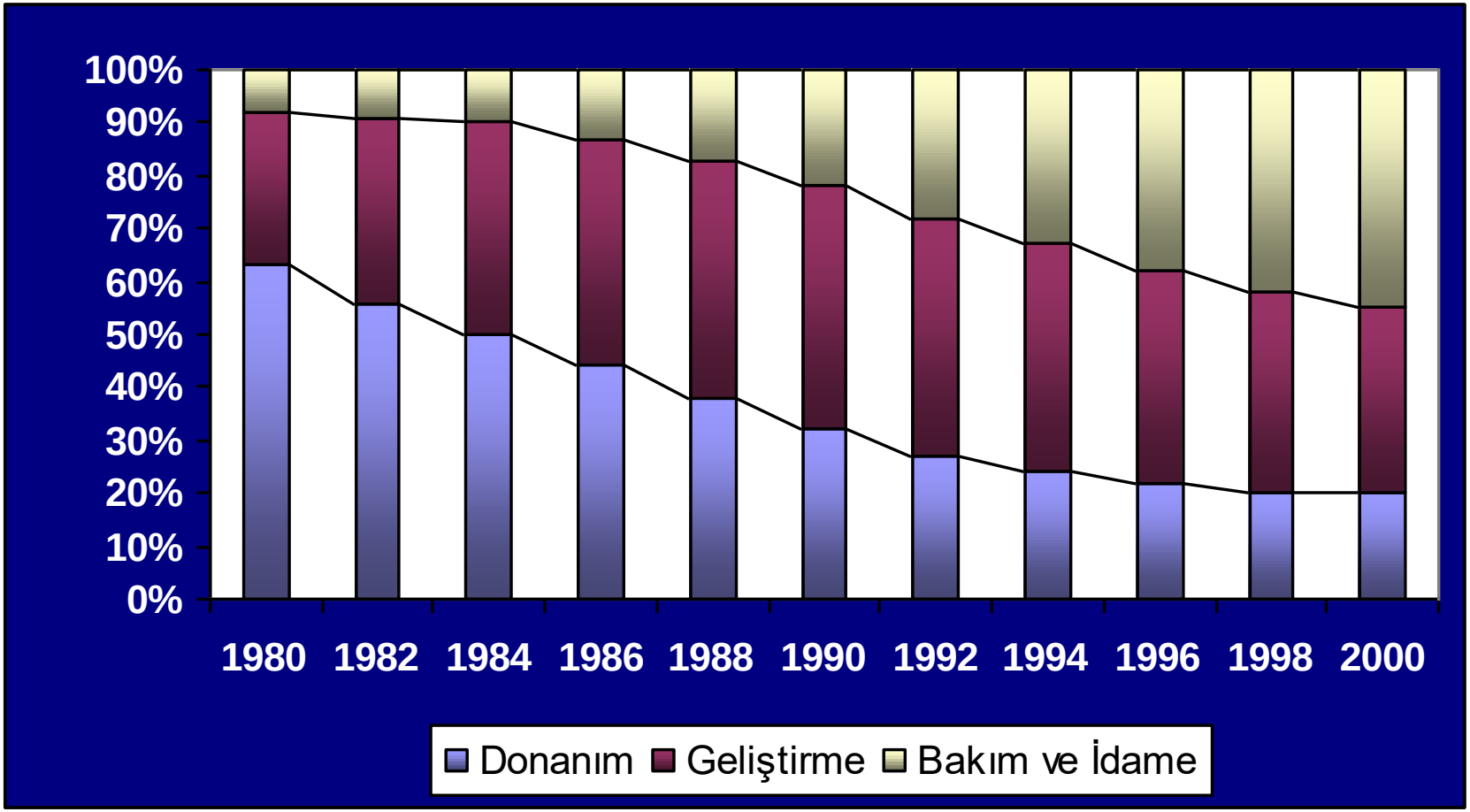
4. Kuşak **Veri Yapısal**

CICS, SQL

5. Kuşak **Yapay Zeka**
Ve Paralel Programlama

CSP, OCCAM

Sistem Harcamaları



Yazılım

Yazılım = Mantık (algoritma) +
Veri (test verisi, bilgi?) +
Belge (dokümanlar) +
İnsan (kullanıcı, geliştirici) +
Program (kod)

- "Bilgisayar sisteminin donanım bileşenleri dışında kalan her şey"

Yazılım

- Mantık, veri, belge, insan ve program bileşenlerinin belirli bir üretim amacına yönelik olarak bir araya getirilmesi, ve

  yönetilebilmesi için kullanılabilecek ve üretilen, yöntem, araç, bilgi ve belgelerin tümünü içerir.

Mantık (algoritma)

- Bilgisayarlaştırılmak istenen işin mevcut mantığı yazılıma yansıtılmak durumundadır.
- Bu nedenle mantık (algoritma) bileşeni yazılımın en önemli bileşenlerinden biridir.

Veri

- 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, bir çok aşamada yapılan ara üretimlere 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 işletime alındıktan sonra bakım çalışmaları sürekli olarak gündeme gelir.
- Bunun iki temel nedeni:
  - hiç bir program bütünüyle her olasılık göz önüne alınarak test edilemez.
  - işletmeler doğaları gereği dinamik bir yapıya sahiptir ve zaman içerisinde sürekli olarak yeni istek ve gereksinimler ortaya çıkabilmektedir.

Yazılım vs Donanım

- 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 vs Donanım

- **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 vs Donanım

- 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 ve Donanım

- 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).

Tipik Bir 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 işi 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

- 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
Belgeleme ve Diğerleri	%35

Yazılımda Hata Düzeltme Maliyetleri

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

Analiz	1
Tasarım	5
Kodlama	10
Test	25
Kabul Testi	50
İşletim	100

Yazılım Maliyetleri

- Yazılım = $ 100.000
- Donanım = $ 1000

Yazılım Sistemlerinin Sınıflandırılması

- İşlevlerine göre
- Zamana dayalı özelliklere göre
- Boyuta göre

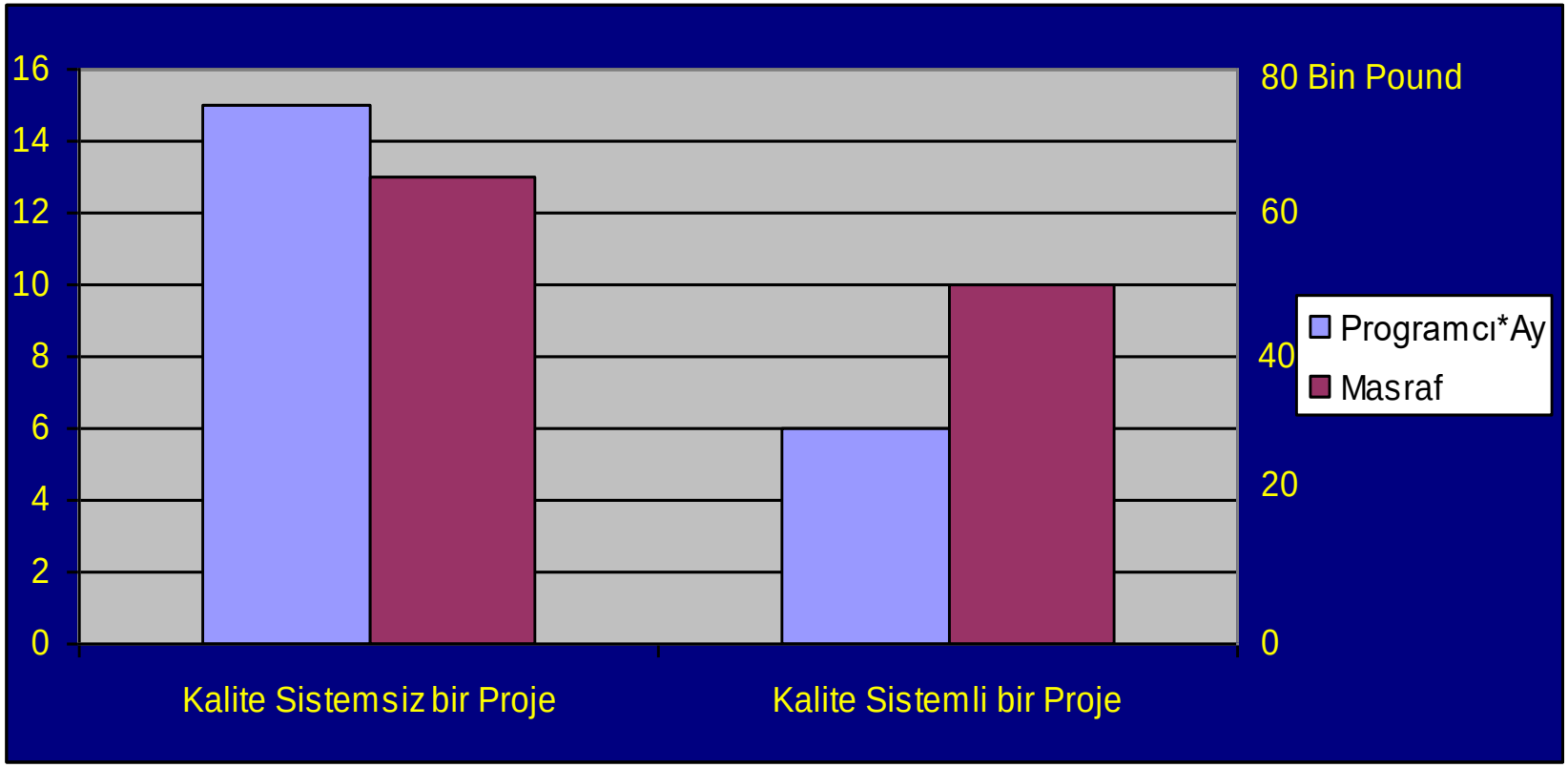
İşleve Göre Sınıflandırma

Hesaplama	Mühendislik Çözümleme
Veri İşleme	Bankacılık
Süreç Temelli	Gömülü Sistemler
Kural Temelli	Robotik, Yapay Zeka
CAD	Sinyal İşleme

Boyuta Göre Sınıflandırma

Küçük (SS<2000)	PC Oyunları Öğrenci Projeleri
Orta (2000<SS<100,000)	CAD BDE Yazılımları
Büyük(100,000<SS<1 Milyon)	İşletim Sistemleri
Çok Büyük (SS>1 Milyon)	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	

Temel terminoloji

- Yazılım (**Software**):
- Yazılım sadece bir bilgisayar programı değildir.
- Basılı veya elektronik ortamdaki her tür dokümanı da içeren ürün.
- Dokümanlar yazılım mühendislerine ve son kullanıcıya yönelik olabilir.
- Uygulama (**Application**) kelimesi de kullanılabilir
- Yazılım ve Donanım adlandırması:
- Yazılım: Software (SW)
- Donanım: **Hardware** (HW)
- İngilizce adlandırma, yazılımın kolaylıkla değiştirilebilecek, oyun hamuru gibi yumuşak bir şeyler olduğu kanısına yöneltir.
- Ancak yazılım daha çok kil veya cam gibidir, bir kere tamamlandıktan sonra değiştirmesi zordur.

Yazılım Türleri

Sistem Yazılımı: System Software

- Diğer programlara hizmet sunmak üzere hazırlanmış programlar.
- Derleyiciler, işletim sistemleri, vb.
- Karmaşık olsa bile belirli, iyi tanımlanmış bilgi yapıları ile uğraşır.
- **Mühendislik Yazılımı / Bilimsel Yazılım : Engineering / Scientific Software**
- Mühendislik ve bilimsel hesaplamalarda kullanılmak üzere hazırlanmış programlar.
- "Numara öğütmek / Number crunching": Bu tip programlar büyük hacimli verilerle uğraştığından bu deyimle karşılaşabilirsiniz.

Yazılım Türleri

Şirket Yazılımı / Kurumsal Uygulamalar (Enterprise software):

- Belirli ticari iş gereksinimlerine yönelik programlar.
- İş süreçleri (business process) ile ilgili bilgiye sahip olmalıdır.
- Genellikle müşteriye özel tasarlanır.
- Veri dönüştürme ve değerlendirme uygulamaları, iş süreçlerinin kimi zaman gerçek zamanlı izlenilmesi, vb.
- Uygulama Yazılımı (Application software):
- Product-line, shrink-wrapped, off-the-shelf, vb.
- Farklı müşteriler tarafından kullanılabilecek genel amaçlı yazılımlar
- Cari hesap uygulamaları, çeşitli otomasyon programları, kelime işlem uygulamaları, vb.

Yazılım Türleri

Gömülü (Embedded) Yazılım :

- Bir ürün veya sistemin bir parçası olup, bu sistemin kendisi ve/veya son kullanıcısı için denetim işlemleri yürüten programlar.
- Gerçek zamanlı (Real Time) uygulamalardır.
- Programın yanıt verme / tepki süresinin (response time) belli bir zaman aşımını (timeout) geçmemesi gerekir.
- Öyle ki, çok hassas bir yanıtın geç gelmesi yerine, yeterli bir yanıtın çabuk gelmesi daha önemli olabilir.

Yazılım Türleri

Ağ Uygulamaları (Web applications):

- Ağ üzerinden haberleşerek hizmet almaya veya vermeye yönelik uygulamalar.
- Şirket yazılımları ile etkileşimde bulunabilirler.
- E-ticaret, B2B, B2C, web servisleri, web tarayıcıları, vb.
- B2B: Business to Business
- İki veya daha fazla ticari firma arasında çalışan uygulamalar.
- Birden çoğa, çoktan çoğa, işbirliğine ve ticari işlemlere yönelik
çeşitli uygulamalardır.
- B2C: Business to Customer
- Doğrudan son kullanıcıya satış amaçlı.

Yazılım Türleri

Yapay Zeka (Artificial Intelligence: AI)
Yazılımları :

- Sayısal olmayan algoritmalarla karmaşık sorunları çözmeye yönelik yazılımlar.

- Robotik, uzman sistemler (expert systems), örüntü tanıma (pattern recognition) (ses ve görüntü), vb.

Yazılım Türleri

Eski Yazılım (Legacy Software)

- İş sürecinin önemli bir parçası olan ve çok uzun süredir kullanılan yazılımlar.
- Şirketler, yazılım sistemleri dahil, yaptıkları yatırımı mümkün olan en uzun sürece kullanmak ister.
- Ancak iş alanındaki gereksinimler hızla değişebilir.
- Yazılım artık yeni ihtiyaçları karşılayacak şekilde esnetilemiyorsa, yazılım yeniden tasarlanmalıdır.
- Eski uygulamaya şirketin diğer bir çok süreci ve bilgi sistemi bağımlı ise, tümleştirme (integration) çalışmaları zor olabilir.

Yazılım Yaşam Döngüsü

Yazılımın bir fikir olarak doğmasından, kullanım dışı bırakılmasına kadar geçen aşamalardır.

- Döngü: Kullanım dışı bırakılan yazılımın yerine yenisi hazırlanabilir.
- Döngünün aşamalarının belirlenmesi ve tanımlanması ile yazılım geliştirme modelleri/süreçleri elde edilir.

Yazılım Geliştirme Süreçleri / Modelleri

Yazılım geliştirme bir süreçtir (**sw development** process)

- Süreç: Önceden belirlenmiş adımlardan oluşan iş akışı.
- Yazılım geliştirme modelleri, sürecin yapısını ve adımlarını belirler.

Yazılım Geliştirme Süreçleri

Modellerin tanımladığı adımlar arasında farklar olmakla beraber, her süreç modelinde bulunan genel işlemler şu şekildedir:

- Çözümleme (Analysis)
- Ne yapılacak?
- Tasarım (Design)
- Nasıl yapılacak?
- Gerçekleme (Implementation)
- Haydi yap!
- Sınama (Testing)
- Doğru yaptın mı?
- Bakım (Maintenance)
- Değişmeyen tek şey değişimin kendisidir!