

B.B-T

Protokol ile birlikte çalışabilme

Standart

Protokol kümeleri

Hizmet Kalitesi

Topoloji ve ağ haritası

Güvenmeli bağlantı

Seri ve paralel iletim

~~Topoloji ve Ağ haritası:~~

Bilgisayar ağını oluşturan bileşenlerin arabaylaşımını, sistemin işlevini ve coğrafi konum açısından belirler.

Ağ katmanı ise ağ üzerindeki sistemi ve onların birbirleriyle bağlantısını gösterir.

~~Band Genişliği:~~ Bir iletim ortamının birim zamanda aktardığı veri miktarıdır. Yani 1 saniyede aktardığı veri miktarı - bps (bit per second) ile gösterilir.

Ortak Yol

Yıldız

Ağaç

Halıka

Örçü

Ethernet (2,84 Mbps, 5 Mbps)  
10 Mbps

Fast Ethernet (100 Mbps)

Gigabit Ethernet (1 Gbps)

10G Ethernet (10 Gbps)

## ÜNİTE - 7 -

### - Bilgisayar Ağları ve İnternet -

Yerel alan ağı (LAN)

Kompis ağı

Genis alan ağı (WAN)

Hot bağlantı

### - OSI Katmanları -

Uygulama Katmanı

İletişim Katmanı

Ötürüm Katmanı

Veri bütü Katmanı

Sunuc Katmanı

Ağ Katmanı

Fiziksel Katmanı

### TCP / IP Katmanları

Uygulama Katmanı

İletişim Katmanı

Yönlendirme Katmanı

Fiziksel Katmanı

Ağ Kartı

HUB cihazı

Modem cihazı

Anahtar cihazı

Güvenlik duvarı

Geant yolu

Yönlendirici

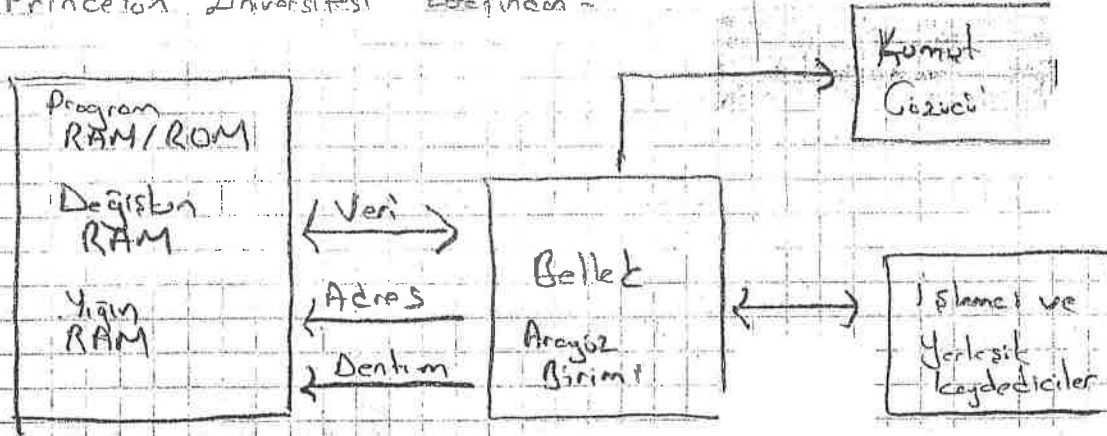
Erişim sunucu

İnternet erişim sağlayıcı

## - Von Neuman Mimari -

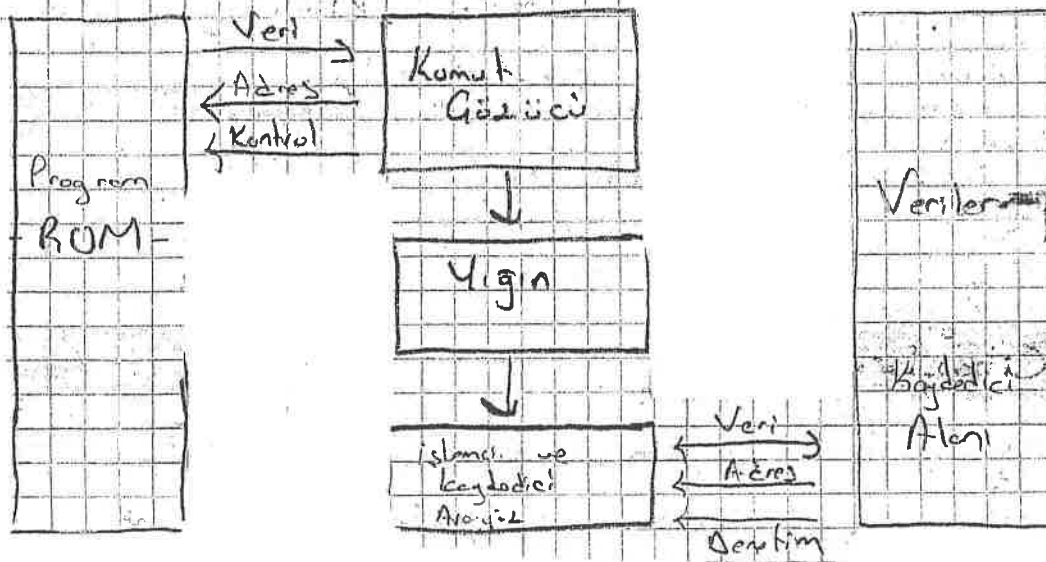
ABD savunma bakanlığının bir serüveni projesi olan bir tasarım geliştirme süreci ortaya çıkmıştır.

Princeton Üniversitesi tarafından -



## - Harvard Mimari -

Harvard Üniversitesi tarafından sunulmuştur.



## RISC-

• Daha basit komutlar kullanarak tümdeure karmaşıklığını azaltmaktadır.

\* Ancak RISC komutlarının daha kısa olması belirli bir görevin tanımlanması için daha fazla komuta gereksinim duyulmasına yol açabilir.

• RISC mimarileri için üretilen derleyiciler daha önce CISC mimarisinde bulunan donanım birimlerinin görevini üstlenmek için fazla komutlar kullanılmaktadır.

• Bazı önemli RISC: Alpha, ARC, ARM, AVR, MIPS, PA RISC vs.

Özellikler	RISC	CISC
Geriim Süresi	+ : Mikro-kod gerim aşamaları içinde gerçek zamanlı olarak çalışır. Komutlar daha hızlı işlenir.	- : Uzun süren kod gerim aşamaları.
Programlama Programlaması	+ : Daha kolaydır daha basit kod kullanımı. Aynı işlev için, daha az kod kullanımı.	- : Karmaşık assembly komutları. + : Daha az gerimde daha çok işlevi sağlar.
Derleme Performansı	+ : Tekil komutlardan dolayı daha basit kod gerimi.	+ : Daha kısa sürede kod gerimi.
	+ : Basit komutlar ve basit kod aşamaları, daha az donanım.	+ : Karmaşık komutlar ve karmaşık kod aşamaları, daha çok donanım.

## Bellek Yönetim Arasından-

1. mimari üzerinde tasarımlar

1 Von Neuman

2 Harvard

İlikt okarı

-141 EA

İşlem kodu

Adresleme Sektörü

İşlevler

Yürümler.

Sok luyıcı Adresleme

Doğrudan //

Dolaylı //

İvedir //

Bazı Adresleme

Komut İşlem Teknik Adresleme

• Mikro işlemciler komut işleme teknik - eaisinden id grup mimari altında toplanabilir.

1 Karmaşık komut kumalı bilgisayar - CISC -

2 Azaltılmış komut kumalı bilgisayar - RISC -

CISC

• Tasarım ilkesi = Donanım yazılımdan her zaman hızlıdır.

• Her işlem için farklı bir komut kullanmak işlemleri hızlandırır, ancak donanımın yükü, tüm donanımın ve güc gereksinimi artar.

• Oldukça az sayıda olan komutları çalıştırmak için MIKRO-KOD kullanılmaktadır.

• Farklı uzunlukta olan bu komutların çalıştırılmasında oldukça karmaşık devrelere (kod arazüçülere) ihtiyaç vardır.

• Bazı önemli CISC işlemcileri: Intel x86 serisi

8051, 280 ve

Motorola 68000

## LİNİTE -6-

### -Assembly Kodunun Gelişme Aşamaları-

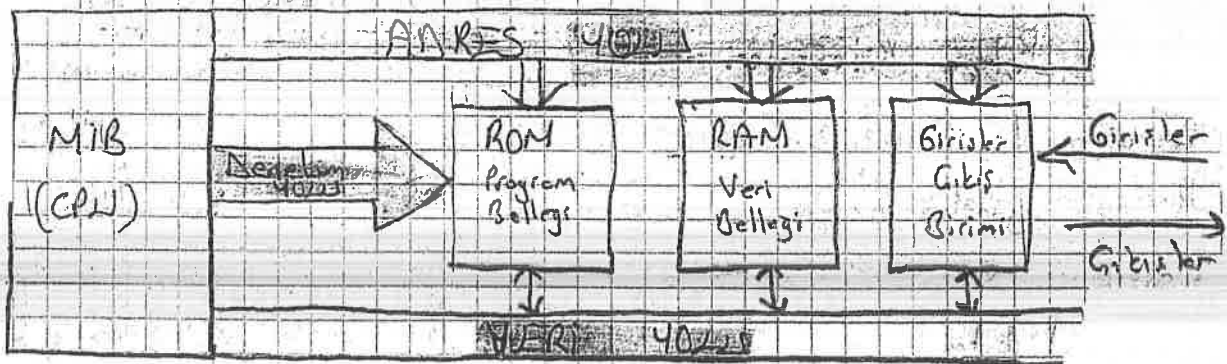
- Getir
- İşlem kodunu al
- Geliş
- Sonucu yaz

### Mik. Gevreyi Denetlemek İçin

#### Sys. Etkileşimleri

- ADRES YOLU
- DENETİM YOLU
- VERİ YOLU

### - Mikro İşlemcilerin Genel Birimleri -



### Mikroişlemci ve Uygulama Program İlişkisi-

Mikroişlemci > İşletim Sistemi > Derleyici > Uygulama

Yekir-dogin Goreleri -

- 1 İşlemciye prosasleri atmak
- 2 Kesmeleri yönetmek
- 3 Prosesler arasındaki iletişimi sağlamak

Proses Yönetimi

- 1 Çalışıyor
- 2 Askıda
- 3 Hazır

Verimlilik

Güvenilirlik

Kuruyuculuk

Sezdiricilik

Elverişlilik

Donanım ve Çevre Birimleri

Bellek yönetim birimi

Dosya ve G/C yönetim birimi

Çekirdek katmanı

Sistem Çağruları

Kullanıcı Arayüzü

Uygulama Programları



## - İŞLETİM SİSTEMİ -

### ① Bilgisayar Sistemin Bileşenleri

Donanım (Bellek, İşlemci, Saklama Birimi, G/Ç...)

İşletim Sistemi

Uygulama Programları

Kullanıcı

İşletim Sistemi = Bilgi ile kullanıcı arasında aracı görevi yapar.  
program topluluğudur.

- Yazılım donanım arasındaki bütünlüğü sağlar.
- Kaynakların yönetimi
- Kullanıcıyla sistem arasındaki ilişki uyum ve düzeyin kurulmasını sağlar.

Bilgi Hızına Erişim Düzeyi

Hazırlık + Sorus - İşletim + Soruşturma

### İşletim Sistemleri

Windows Ailesi

UNIX Ailesi

Linux Ailesi

VB Ailesi

Android İşletim Sistemi

Bilgi Sistemin Kaynakları

Bellek

CPU (İşlemci)

Çevre Birimleri (Yazıcı, Disk, vb.)

Veri





## Programlama Dilleri

### Yazılım Geliştirme Süreci

- Gereksinim Analizi
- Yazılım Tasarımı
- Kodlama
- Sertifikasyon
- Bakım

### Yazılım Geliştirme Süreci (Yerli Örnek)

#### - Programlama Ortamı Temel Unsurları -

- Editör
- Derleyici
- Kütüphane
- Bağlayıcı
- Yükleme
- Hata Ayıklayıcı
- Yorumlayıcı

#### - Programlama Dilleri Elemanları

- Söz Dizimi (Syntax)
- Anlam Bilimi (Semantics)
- Veri (Data)
- Atama Değeri
- Tür Kontrolü
- Kontrol Sistemi
- Alt Programlar
- Modüller

#### - Nesneye Yönelik Programlama -

- Veri Nesneleri
- Kolaylık
- Gök birimlilik

## - ALGORITMA -

### - Algoritma Olması gereken Özellikleri -

- Etkin ve genel olma
- Sönü olma
- Yanılsızlık
- Giris- Çıkış tanımlı olmalı
- Basarım

### - Algoritma Tasarım

- Alty adı
- Yaptığı iş
- İşlem adımları
- Açıklama

### - Alty ve İlişki Tanımlaması İcin Çesitli Yöntemler -

- Metinsel tanımlama
- Akış Seması
- N-S Semaları
- W-O Semaları
- Bechmen Notasyonu (Veri tabanı)

○ → Başla

□ → Bitir

▱ → input (Veri girişi)

▭ → Yazdırma (Ekran)

→ → Program sonu

⊙ → Devam

◇ → Karşılaştırma  
(Bundan sonra şart gelir)  
(E/H  
- if -

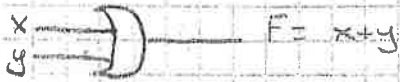
◁ ▢ ▷ → Döngü  
(Bundan sonra E ya da  
H belirtilerek devam eder)  
(doğulu)

▭ → Alt program

⊙ → Dosya işlemleri  
- Kayıt -

## ÜNİTE -2-

Veya Kapısı (OR) = Toplam



Ve Kapısı (AND) = Çarpım



Değil Kapısı (NOT) =



Ve Değil Kapısı (NAND)



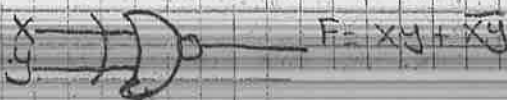
Veya Değil Kapısı (NOR) =



Ya Da Kapısı (XOR)



Ya Da Değil Kapısı (XNOR) =



Tampun Kapısı (Buffer)



Değişme Kuralı =

$$x + y = y + x$$

$$x \cdot y = y \cdot x$$

Özdeşlik Kuralı =

$$x + x = x$$

$$x \cdot x = x$$

Ve Kuralı =

$$x \cdot 1 = x$$

$$x \cdot 0 = 0$$

Veya Kuralı =

$$x + 0 = x$$

$$x + 1 = 1$$



u n n

## LÜNİTE -1-

Temel Konular (Kavramlar):

$$8 \text{ bit} = 1 \text{ Byte}$$

$$1 \text{ KB} = 1024 \text{ Byte} = 8 \cdot 1024 \text{ Bit}$$

$$1 \text{ MB} = 1024 \text{ KB} = 8 \cdot 1024 \text{ Byte} = 8 \cdot 1024 \cdot 1024 \text{ Bit}$$

$$1 \text{ GB} = 1024 \text{ MB} = 8 \cdot 1024 \cdot 1024 \text{ Byte} = 8 \cdot 1024 \cdot 1024 \cdot 1024 \text{ Bit}$$

$$1 \text{ TB} = 1024 \text{ GB} = 8 \cdot 1024 \cdot 1024 \cdot 1024 \text{ Byte} = 8 \cdot 1024 \cdot 1024 \cdot 1024 \cdot 1024 \text{ Bit}$$

**Donanım** = Genel olarak elektronik elementlerden oluşan bilgisayar sistemini oluşturan birimlere verilen addır.

- İşlemci
- Disk
- Ana kart
- Bellek
- Hard disk

**Protokol**: Ağlarda kullanılan veri aktarım kurallarıdır.  
- PROTOCOL denir

**CPU**  $\Rightarrow$  Merkezi İşlem Birimi "Central Processing Unit"

Tüm işlemlerin yapılmasında sorumludur. Verilerin okunması gerekli hesapların yapılması sonuçların üretilmesi ve saklanması hep bu işlem birimi sayesinde olur.

**Bellek**  $\Rightarrow$  Donanım parçasıdır.

En çok kullanılan bellek RAM'dir.

Programlar ve kodların üzerinde işlem yapılacak verilerin ve üretilen sonuçların içinde saklandığı donanım parçasıdır.

RAM  $\rightarrow$  Geçici 1. Bellektirler

ROM  $\rightarrow$  Kalıcı 2

**Bellek Hiyerarşisi**

Erism  
Hızı  
Artıyor

Saklayıcı  
Cep Bellek  
1a Bellek (RAM)  
Saklama Birimleri  
Yedekleme Birimleri

Sığa  
Artıyor

## ÜNİTE-8-

### - Sayı ve Kodlama Sistemi -

#### - Veri Sıkıştırma -

Bir blokta sıkıştırma dosya boyutunu küçültme işlemidir.

##### 1-) Kayıpsız Sıkıştırma:

Dosya içerisinde tekrar eden dosyaların daha bir den kapsayarak birimde gösterilmesine dayanır.

##### 2-) Kayıplı Sıkıştırma:

Yinekiinde ilke ise kodlama yapılırken bazı bazı bilgilerin silinmesidir.

#### - Dört Kendi Notum -

2 tabanlı sayı sistemi  $\rightarrow$  2 göster - Binary

8 tabanlı sayı sistemi  $\rightarrow$  Decimel

10 tabanlı sayı sistemi  $\rightarrow$  Oktal

16 tabanlı sayı sistemi  $\rightarrow$  16 ile gösterilir - Hexadecimel

#### - İkili Sayı Sistemi -

İki sayı sis. Onlu Sayı sistemine çevirirken 2'nin üstleriyle çarpılır.

$$(111)_2 \Rightarrow ( )_{10} = ?$$

$$2^0 \cdot 1 = 1$$

$$2^1 \cdot 1 = 2$$

$$2^2 \cdot 1 = 4$$

$$(111)_2 \Rightarrow (7)_{10}$$

7



## "Onlu Sayı Sistemi"

Tebani olarak 10 sayıları kullanılır.

ör

$$\begin{aligned} (6789)_{10} &= \\ &\begin{cases} 10^0 \cdot 9 = 9 \\ 10^1 \cdot 8 = 80 \\ 10^2 \cdot 7 = 700 \\ 10^3 \cdot 6 = 6000 \end{cases} \\ &= 6000 + 700 + 80 + 9 \\ &= 6789 \end{aligned}$$

2'li tabana çevirmek istiyorsa şöyle  
2'ye bölünüz.

$$\begin{array}{r|l} 6789 & 2 \\ \hline & 12,2 \end{array}$$

$$= (110110000101)_2$$

gibi sayı olur.

## "8'li Tabanlı Sayı Sistemi"

Bu tabanlı sayı 8'lige çevirmek için Sayıları 3'er 3'er  
ayırıp 2'nin üstleriyle karşılarız.

Bu tabanlı sayı 16'lige çevirmek için Sayıları 4'er 4'er  
ayırıp 2'nin üstleriyle karşılarız.

$$A=10$$

$$B=11$$

$$C=12$$

$$D=13$$

$$E=14$$

$$F=15$$

$$\text{ör} \quad (11111111)_{16} = (?)_{16}$$

$$\begin{array}{r} 1111 \\ 2^3 2^2 2^1 2^0 = 15 \end{array}$$

$$11111 = 15$$

$$\Rightarrow (FF)_{16}$$

## 16'li Sayı Sistemi

Onaltılı sayı sisteminde toplam 16 olduğu için tek bir basamakla ifade  
geçer. Bu nedenle 0, 1, 2, 3, ..., 7, 8, 9, 10, 11, 12, 13, 14, 15 kadar olan basamaklar 16'lı  
sistemde olduğu gibi okunur.

$$10=A$$

$$11=B$$

$$12=C$$

$$13=D$$

$$14=E$$

$$15=F$$

en küçük rakam 0

en büyük rakam = 16-1 = 15'tir.

$$\text{ör} \quad (2BE)_{16} = (202)_{10}$$

PDF Eraser Free

İkili sayıları 16'lı sayılara çevirmek için sayıları bölümlenerek dörder dörder ayırıp 16'lı kısıtlarını yazılır.

$2^4 = 16$  old için

$\frac{1100101001000101}{2^3 2^2 2^1 2^0 / 2^3 2^2 2^1 2^0 / 2^3 2^2 2^1 2^0 / 2^3 2^2 2^1 2^0}$   
 12 (C)      10 (A)      4      5

$$(110010100100101)_2 = (CAL5)_{16}$$

Toplane =

Örne sayı sisteminde aynı düzeydeki her bir basamak rakamların toplamı en büyük rakam olan 9'dan büyük olduğunda, bu toplamdan 10 ve 10'un katları çıkarılır. ve bir üst düzeye 10'ların sayısı eklenir.

2 İki grupta ise toplam 2 olduğu için iki ve ikinin kolları sayıları bir sonraki hanteye eklenerek toplanır.

$$\begin{array}{r}
 1011 \\
 0101 \\
 1100 \\
 + 1110 \\
 \hline
 101010
 \end{array}$$

$$1 \cdot 1 + 2 \cdot 1 + 2 \cdot 0 + 2 \cdot 1 = 11$$

$$= 5$$

$$= 12$$

$$= 14$$

$$42 \quad (\text{10th system})$$

Q. Karna 2  
2 liketok

$$\begin{array}{r} 11011 \\ - 10101 \\ \hline 00110 \end{array}$$

Qorpma =

$$\begin{array}{r} 1011011 \\ \times \quad 101 \\ \hline 1011011 \\ 0000000 \\ 1011011 \\ \hline 111000111 \end{array}$$

- Bölme -

$$\begin{array}{r} (111000111)_2 \\ \underline{2^9 \ 2^8 \ 2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0} \end{array} \div \begin{array}{r} (101)_2 \\ \underline{2^2 \ 2^1 \ 2^0} \end{array}$$

$$1 + 2 + 4 + 64 + 128 + 256 \rightarrow 455$$

$$1 + 4 = 5$$

$$455 / 5 \rightarrow \text{iki sifirinde } (1011011)_2$$

- KIRILAN KINNASI -

Onlu Sayı Sis =

$$365.789 \Rightarrow 365 \text{ ten binde } 789 \text{ diye okunur.}$$

İki Sayı Sis =

$$\begin{array}{r} 111.11 \\ \underline{2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0} \end{array} \Rightarrow 7 \text{ ten binde } 3$$

## ÜNİTE - 9-

### YAZILIM MÜHENDİSLİĞİ

Bilgi teknolojisi insanların ne istediklerini iyi anlayabilmeleri, onlara hizmet edebilmeleri için çeşitli programlama dilleri kullanılır.

#### Sistem ve Yazılım

- Bilgisayar Sistemi Mühendisliği
- Sistem geliştirme süreci
- Yazılım mühendisliği

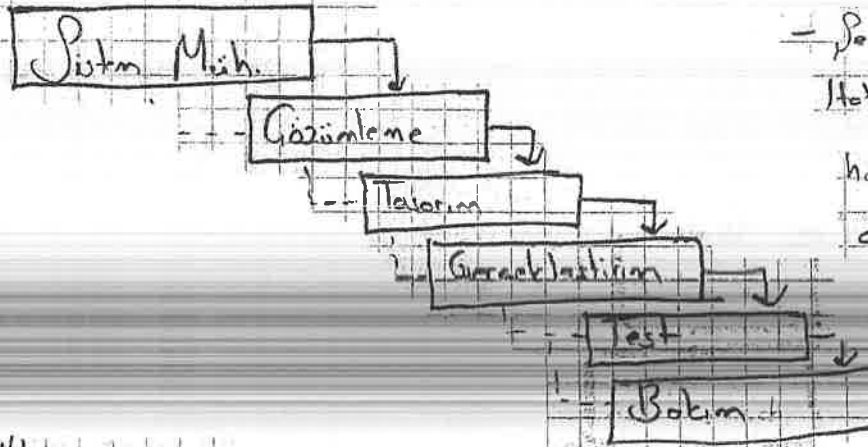
#### Yazılım Mühendisliği ve Yöntembilimleri

- Yazılım geliştirme standartları
- Yazılım geliştirme modelleri
  - ✓ Klasik Çevrim (Sade)
  - ✓ Evimsel Geliştirme
  - ✓ V Modeli
  - ✓ Artırımı Geliştirme
  - ✓ Prototiplene
  - ✓ Yeni Teknikler
  - ✓ Spiral Model
- Yazılım geliştirme süreci
  - ✓ Yazılım istekleri gözümleme
  - ✓ Yazılım tasarımı
  - ✓ Yazılım gerçekleştirim
  - ✓ Yazılım testi
  - ✓ Yazılım bakımı
- Yardımcı süreçler
  - ✓ Yazılım nitelik güvence
  - ✓ Yazılım düzenlem yönetimi
  - ✓ Yazılım proje yönetimi

## - Yazılım Geliştirme Modelleri -

Klasik Gevrim = Sadece Waterfall yöntemi olarak bilinir.

Sistematiik olarak ilerleyen ardışık bir yaklaşımla yazılım geliştirilmesini sağlar. Bu modelde önce tüm istekler tam olarak tanımlanır ve sabitlenir. Bu isteklere göre tasarım yapılır, sabitlendikten sonra gerçekleştirilme yapılır. Geliştirilen birimler kendi aralarında birleştirilir. Daha sonra da kabule esas olan sistem testleri yapılır ve sistem kullanıma hazır hale getirilerek bakım aşamasına başlatılır.

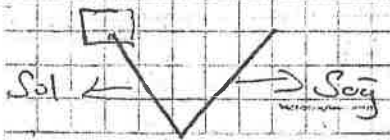


- Sadece modelin dezavantajı  
Her bir adım  
her bir önceki adıma  
bağımlıdır.

- Klasik modelde yazılımın gelişimi doğrusaldır.
- Bilgi akışı yalnız bir yönde gerçekleşir.
- Klasik gevrim, özellikle iyi tanımlı, istekleri kesinleşmiş, fazla değişim olmayan, sabit fiyat ve hükümlere göre ödeme ilkesine dayanan satın alma projesi için uygun bir yöntemdir.

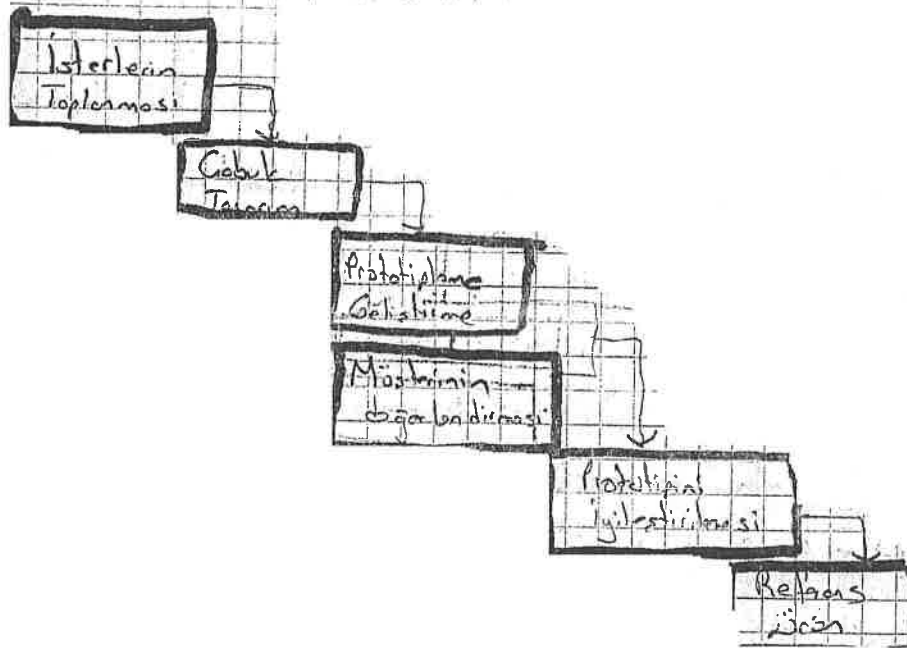
V Modeli - Geliştirme sürecine sistem kavramıyla başlayarak Klasik Modeldeki test işleminin ne zaman yapılacağını ön plana çıkarır.

- Sol konut üretim etkinlikleri
- Sağ konut da test etkinlikleri ile ilgilidir.

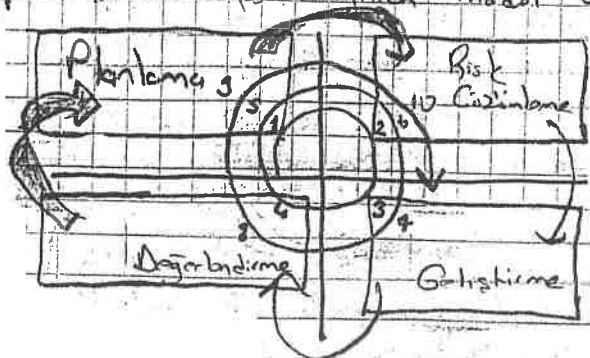


## Prototipleme

Bu yöntemde geliştirici ile kullanıcı beraberce sistemin genel isteklerini tanımlarlar, üzerinde daha fazla düşünülmesi gereken noktaları belirlerler. Ondan sonra cabuk bir tasarım yapılır. Bu tasarım daha çok kullanıcı ile olan etkileşimini ya da sistemin en temel işlevini belirler ve ön ürünün, yani prototipin buna göre yapılmasını sağlar. Daha sonra bu ön ürün müşteri tarafından denenerek istekler yeniden gözden geçirilir; geliştirici tarafından da bu istekler ön ürüne yansıtılır.



Hem klasik çevrim hem de prototipleme yöntemlerinin iyi yönlerinin iyi yönlerinin birleştirilmesiyle oluşturulmuştur. Spiral model genel olarak dört aşamadan meydana gelir. Bu aşamalar en küçükten başlayarak büyüyerek ürünün temamlanmasına kadar tekrar eden bir çevrim halinde olduğundan ve bir spiral şeklinde gösterildiğinden model bu adı almıştır.



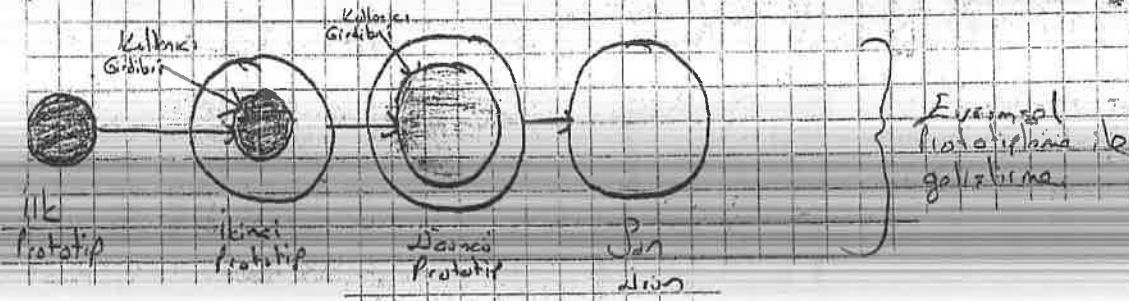
Spiral modelde bir döngü vardır.



- Her evrimde geliştirilen ürünler uygulama alanında tam işlevselliğe sahiptir.
- Ürünün kullanımı sırasında elde edilen deneyimler, geri bildirimleri ve yeni gereksinimlerle bir sonraki evrime geçer. Her yeni evrim, sistemin kapsamını, işlevlerini ve yeteneklerini biraz daha artırır.
- Ortaya çıkan her ürün testim edilebilir kullanıma sunulur.

Benzer bir model olarak evrimsel prototiplama modelinde ise;

Evrimsel geliştirilmeden bir önceki olarak, her evrimde ortaya bir prototip, yeni ön ürün çıkarılır. Ön ürünün uygulama alanında davranışlarında kullanıcı deneyimleri alınarak bir başka ön ürün oluşturulur ve denemeye sunulur. Bu şekilde geliştirilmeye devam edilerek ürün son haline getirilir ve sonra tam kullanıma sunulur.



"Kısa döngü yaklaşımı"

İsterler birinin tamamı belirli olan bir ürünün sistemler geliştirilmesi ilkesine dayanmaktadır.

Önce ortaya isteklere göre belirli bir yolda olan ilk ürün geliştirilir.

Her süreç bir birimden daha fazla işlevselliğe sahiptir.

Bu yüzden isteklerin belirli bir tabana göre dağıldığı uygulamaları geliştirme için kullanılır.

Yazılım geliştirme tekniklerinde ortaya çıkan son yazılımlar çeşitli yazılım ve gereksinimlerini kullanımı ön görmektedir.

Geliştirilecek yazılımın özellikleri bu araç ve gereçler yardımıyla üst düzeyde tanımlanır, sonra da otomatik olarak bu işleri yapacak olan kaynak kod üretilir.



## - İsterlerin Tanımlanması -

- Kullanım Senaryoları (USE CASES)
  - Veri akış diyagramları (Data flow diagrams)
  - Nesne diyagramları (Object diagrams)
  - İşlevlik diyagramları (Activity diagrams)
  - Etkileşim diyagramları (Interaction diagrams)
  - Durum haritası diyagramları (Statechart diagrams)
  - Durum geçiş diyagramları (State transition diagrams)
  - Resmi belirtim dilleri (Formal specification languages)
  - Veri sözlüğü (Data dictionary)
- İsterlerin sınıflandırılması daha iyi anlaşılmasını ve iyi tanımlanmasını sağlar.
- Belirtilen özelliklerden dikkate alınması gereken sistem özellikleri şöyle listelenebiliriz:

### - İşlevsel Özellikler

■ Güvenlik (Kısıtlı kullanıcı, Yetenek güvenliği, Doğrulama güv., Erişim güv. Veri)

■ Kullanım kolaylığı

■ Diğer yazılımlar ile uyumluluk

■ Bakım kolaylığı

■ Bilgi sistem platformları arası taşınabilirlik

■ Başka sistemlerle beraber çalışabilirlik

■ Veri yedekleme ve kurtarma

## ÜNİTE - 10 -

### VERİ YAPILARI VE VERİ MODELLERİ

Veri yapısı, = bilginin anlamlı yapıda bellekte veya saklama birimlerinde tutulması şeklini gösterir. Bir nesit tutulması islenidir.

Veri Modeli = Bir küme içindeki verilerin birbirleriyle olan ilişkisel veya sırası durumunu gösterir.

Veri Tabanı = Birbirleriyle ilişkili verilerin bütünleşik olarak tutulduğu bir alandır.

#### Tanımlanmış Veri Yapıları

Topluluk oluşturma = Birden çok veri yapısının bir araya getirilip yeni veri yapısı bir aile ortaya aktır.

Ortaklık Oluşturma = Birden çok değişkenin aynı bellek alanını kullanmasını sağlayan veri yapısı tanımlamasıdır.

Bir sınıftan çok öğrenciler birer veri ise, bu öğrencilerin sınıfta oturacak sıralar veri yapısına karşılık gelir. Öğrencinin nasıl oturacağını sıraların durumu belirler. Ancak öğrencilerin sıralara anlamlı bir şekilde oturması ise veri modeline karşılık düşer.

#### Veri Temel Yapıları

Tamsayı (int)

Karakter (char)

String (string)

Gerçek Sayı (float, double)

Dizi, Matris

## Topluluk Yapı (Structures)

Arabic  
int mod = 1996  
char  
string

Array Ortaklık

int dist[10]

int i, a, x=5

Char y = 'A'

String s = "Abdullah"

float }

double }

= Veri Modelleri =

Bağlantılı liste veri modeli

Ağac veri modeli

Graf " " "

Durum makinesi veri modeli

Veritabanında ilişkisel veri modeli

Ağ veri modeli

Veri Yapıları ve Veri Modelleri Arasındaki farklılıklar-

Simple olarak veri yapısını gösterir

Veri yapısı ve Modelleridir.

Topluluk Oluşturma

Ortaklık Oluşturma

Bit dizeyinde Erişim

Bir çok  
veri yapısının  
birbirine  
yatırıp yani  
bir veri yapısı  
bir diğer ortama  
göçmektedir.

- Diziyile Bogil liste arasindaki fark; ikisinin de avantajlari ve dezavantajlari -

- Diziler -

Isledigin kederi yer ayirilir

Verije ulasmak kolay

Elb - Giderim geymek zar

Bogut degistirme zor

- Bogil Liste -

Ulasim zar

Dinamik olarak kullanilabilir

Elb - giderim kolay

Hariza avantajlari var

Her bir ebacın bir sanetini gösterir

İşareti sayılar  
ödevimiz

## ÜNİTE-11-

### VERİTABANI YÖNETİMİ VE SQL

# İLİŞKİSEL VERİTABANI VE İLİŞKİSEL MODEL

- Temel yapı 2 boyutlu tablodur
- Tablonun sıralarında bir verile ilgili kayıtlar tutulur.
- Tablodaki 1 sıra ile sütunun kapladığı yere veri hücresi denir.
- Fiziksel ortamda tablo bir veri dosyası olarak saklanır.

### Veri Tabanının Tasarım Adımları

Gereksinim Analizi

Kavramsal Tasarım

Montiksel Tasarım

Sistemin uygulanması

Sistemin test edilmesi ve bakım

**SORU:** Aşağıda veri tabanı tablolarının fiziksel formlarına göre bu tablolardan oluşacak Kurum ve tabanın fiziksel formu no yarınit. Aşağıdaki tabloları oluşturmak ve tablodaki verileri girme için gerekli SQL komutlarını yazınız

ÖĞRENCİ		
TCNO	ADI	Gns Tarih
11111	Ahmet	2010
22222	Ayşe	2011
33333	Ayşe	2012

DERSLER		
DERS	KREDİ	Öğretim Görsi
BGT	2	Korokörte
ENG	1	Can

CREATE → oluşturma

KURUM { ÖĞRENCİ, DERSLER }  
ÖĞRENCİ (TCNO, ADI, GİRİTARİH)  
DERSLER (DERS, KREDİ, ÖĞRETMEN)

Başlıkların oluşturulması

CREATE TABLE ÖĞRENCİ (  
TCNO VARCHAR2(5) PRIMARY KEY, NOTNULL → Başlıkların oluşturulması  
ADI VARCHAR2(5) NOTNULL → Anonim bir isim verilebilir (TCNO gibi)  
GİRİTARİH DATE);  
(Tarih old. için)

CREATE TABLE DERS (  
DERS VARCHAR2(15) NOTNULL  
KREDİ NUMBER(3) NOTNULL  
ÖĞRETMEN VARCHAR2(10) NOTNULL);  
(Sıralı old. için)

INSERT INTO ÖĞRENCİ VALUES (11111, 'Ahmet', '2010');  
INSERT INTO ÖĞRENCİ VALUES (22222, 'Ayşe', '2011');  
INSERT INTO DERSLER VALUES ('BBT', 2, 'Korkmaz');  
INSERT INTO DERSLER VALUES ('BMG', 1, 'Can');  
INSERT INTO DERSLER VALUES ('Fizik', 3, 'Kaya');

SÖZ

CREATE TABLE ⇒ Veri Tabanı içerisinde yeni bir tablo yaratır

CREATE TABLE ... AS ⇒ Var olan bir tablonun fiziksel yapısını ve kayıtlarının kopyasını yeni bir tablo yaratır.

INSERT INTO ... VALUES ( ) ⇒ Tabloya bir kayıt ekler

DELETE FROM ⇒ Tablodan kayıtları siler

SELECT ... FROM ?

WHERE ...

ORDER BY ...

GROUP BY ...

Temel sorgulama konusudur. İstediğin kısıtları uygun kayıtları seçer ve gösterir.

~~ör~~ 3 des bir sayısal rakamla toplamı yapan  
ekimce okis dijitalıyı görür

~~ör~~ 188416 bit ka KB

1 Bit 8. 1024 Bit

188416

188416. 8024 R

~~ör~~ Kullandıkları gelen ilk harfleri sonra herfi yapan

programı akrona yollar

Aslında verilecek



