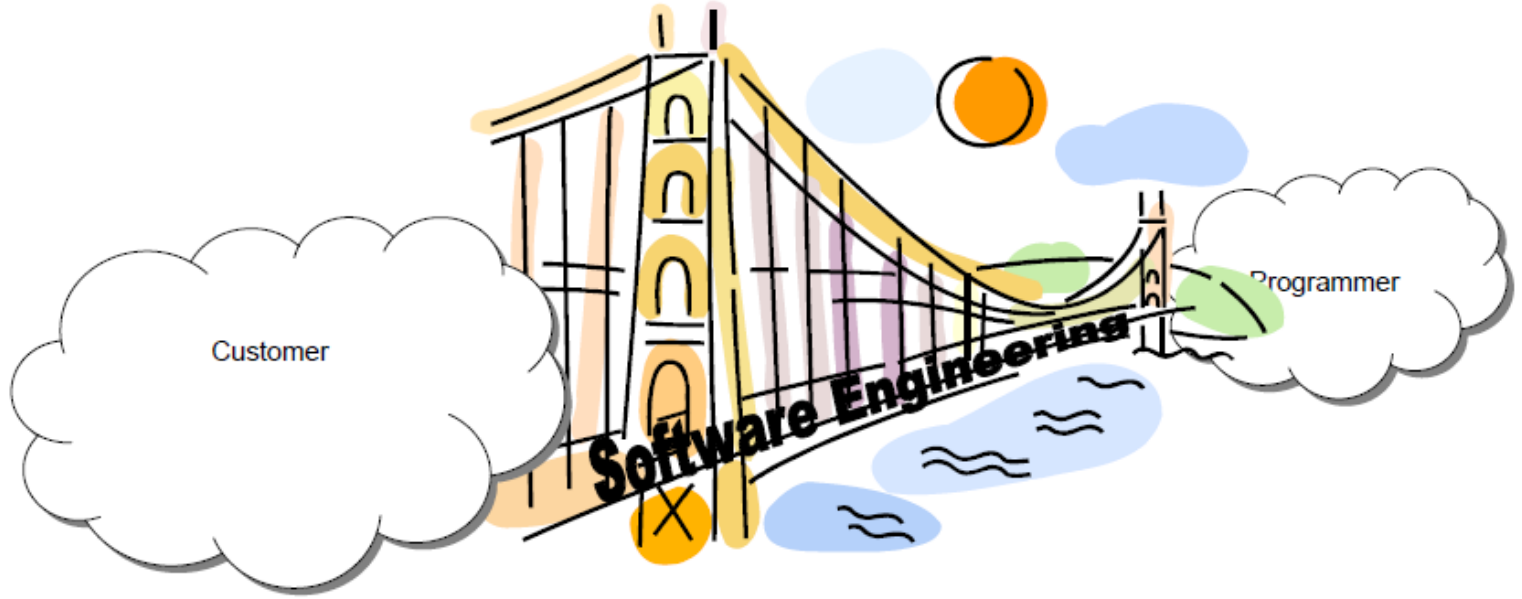
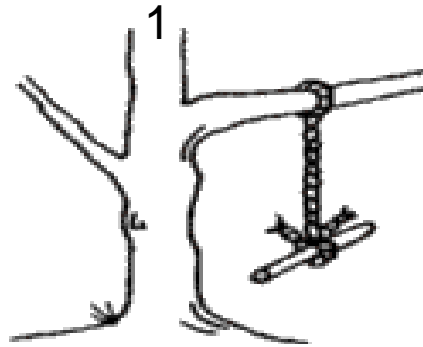


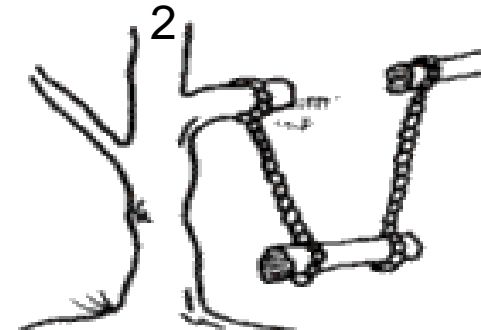
Yazılım Mühendisliği



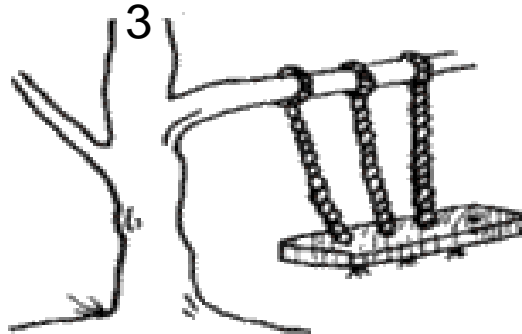
Ekip çalışması ☺



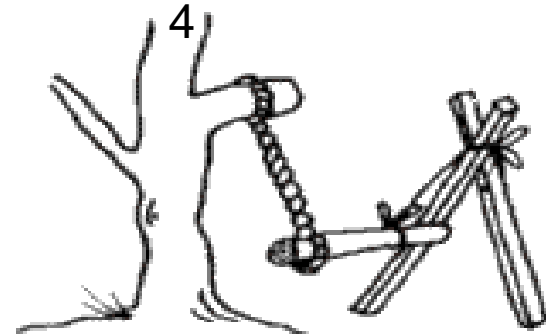
What the user asked for



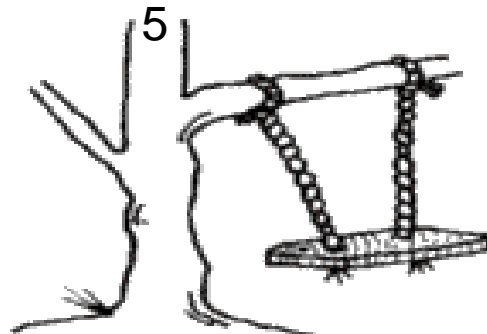
How the analyst saw it



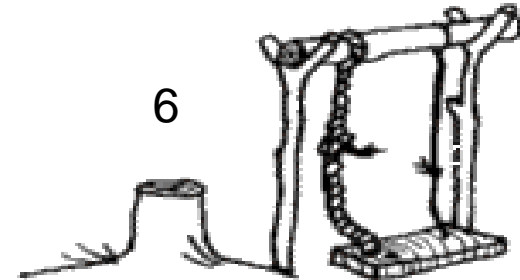
How the system was designed



As the programmer wrote it

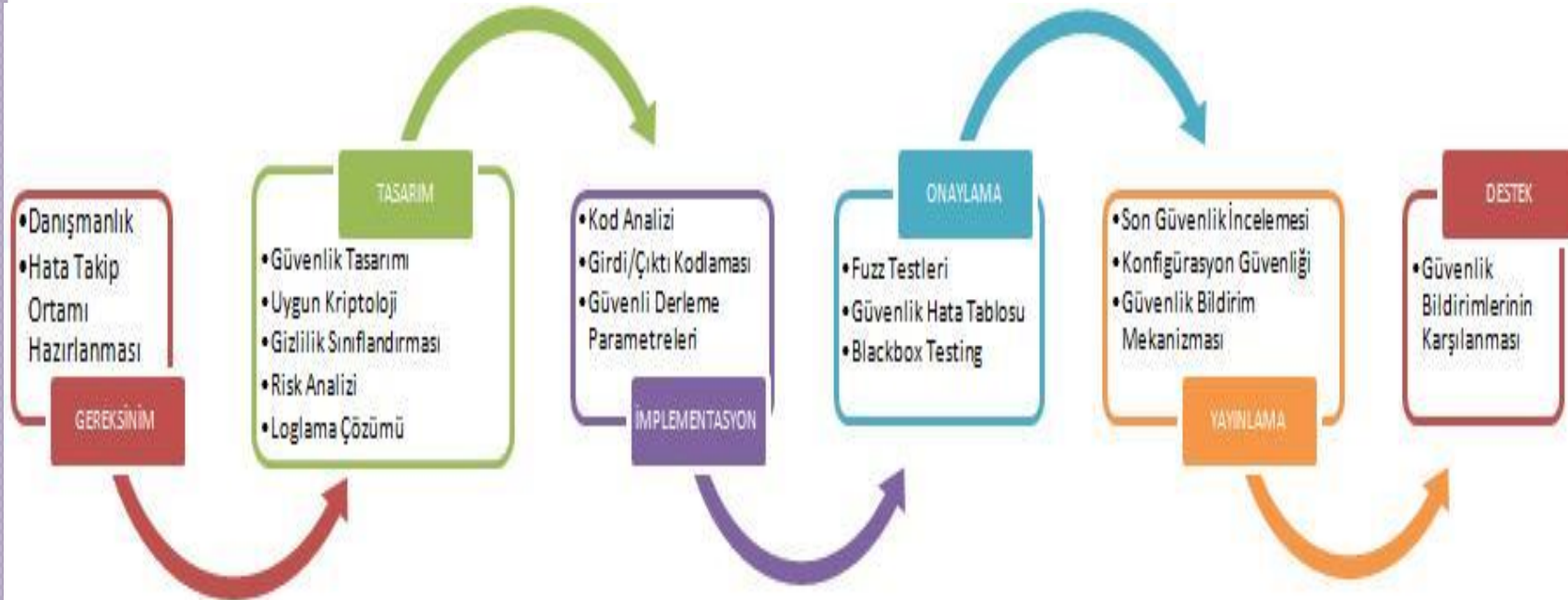


What the user really wanted



How it actually works

Yazılım Yaşam Döngüsü (YYD)

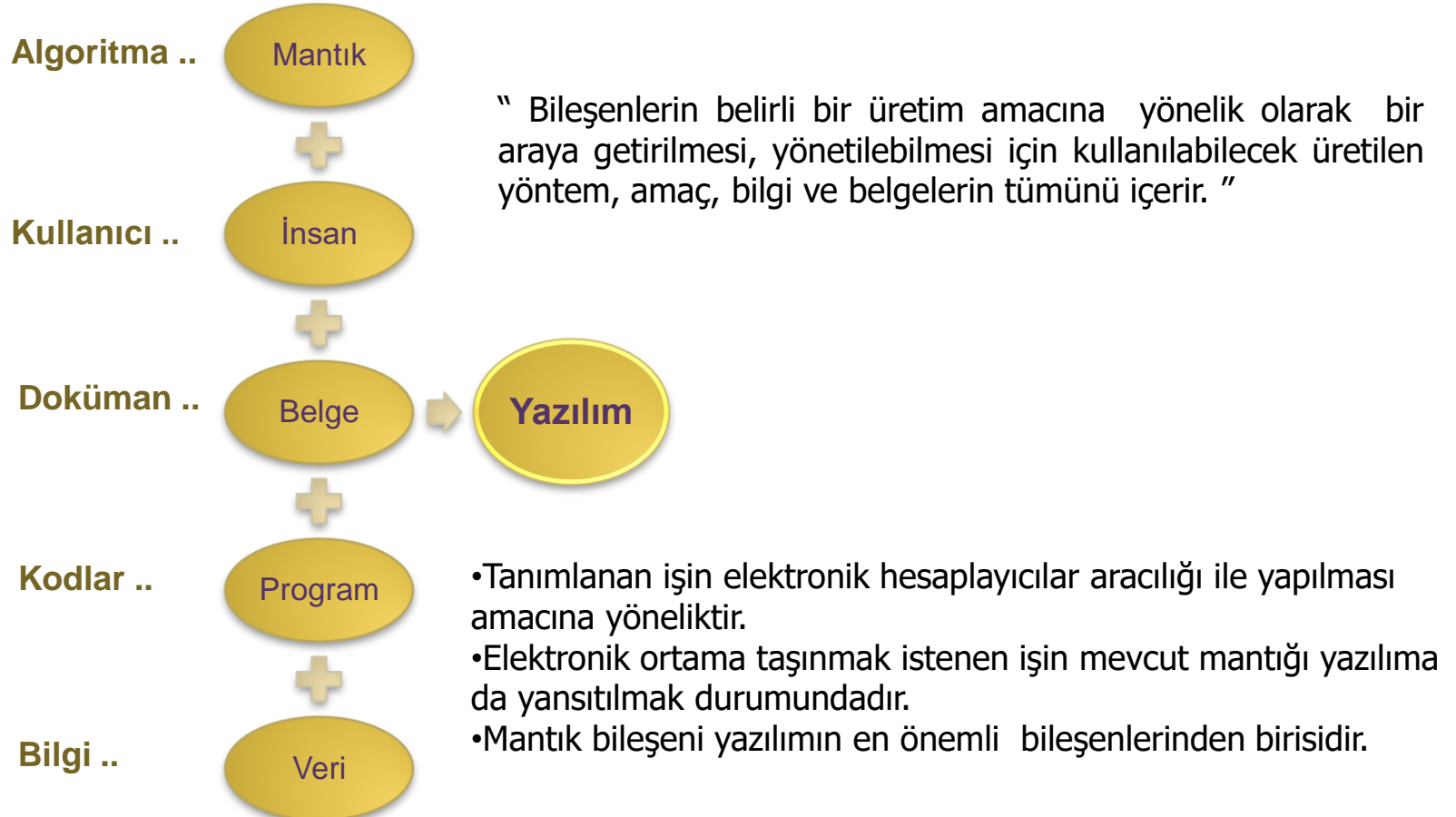


Temel Kavramlar ;

Yazılım: Bir sistemin donanım bileşenleri dışındaki her şey olarak tanımlanabilir.

Yazılım; Bilgisayar program parçası yada programlar grubu olarak an

Yazılımı oluşturan bileşenler



- Her tür yazılım mutlaka bir veri üzerinde çalışma durumundadır.
- Veri dış ortamdan alınabilir yada yazılım içerisinde üretilebilir. 'veri' yi 'bilgi' ye dönüştürme en önemli amaçtır.
- Yazılım üretimi, bir mühendislik disiplini gerektirir.
- Yazılım Yasam Döngüsü (YYD) mühendisler tarafından üretim sırasında kullanılan yasam döngüsünden esinlenerek oluşturulmuştur.
- Yazılım üretimi sırasında, birçok aşamada yapılan ara üretimler, bilgi belge üretimidir.
 - Planlama bilgileri,
 - Çözümleme bilgileri,
 - Tasarım bilgileri,
 - Gerçekleştirim bilgileri

Yazılımın insan bileşeni iki boyutludur.

1. Yazılımı geliştirenler
2. Yazılımı kullananlar

“Çok kişili ekiplerle geliştirilmektedir. Temel nedeni yazılımın yaygınlaşması ve boyutlarının büyümesidir.”

Yazılımın ana çıktısı sonuçta bir bilgisayar programıdır.

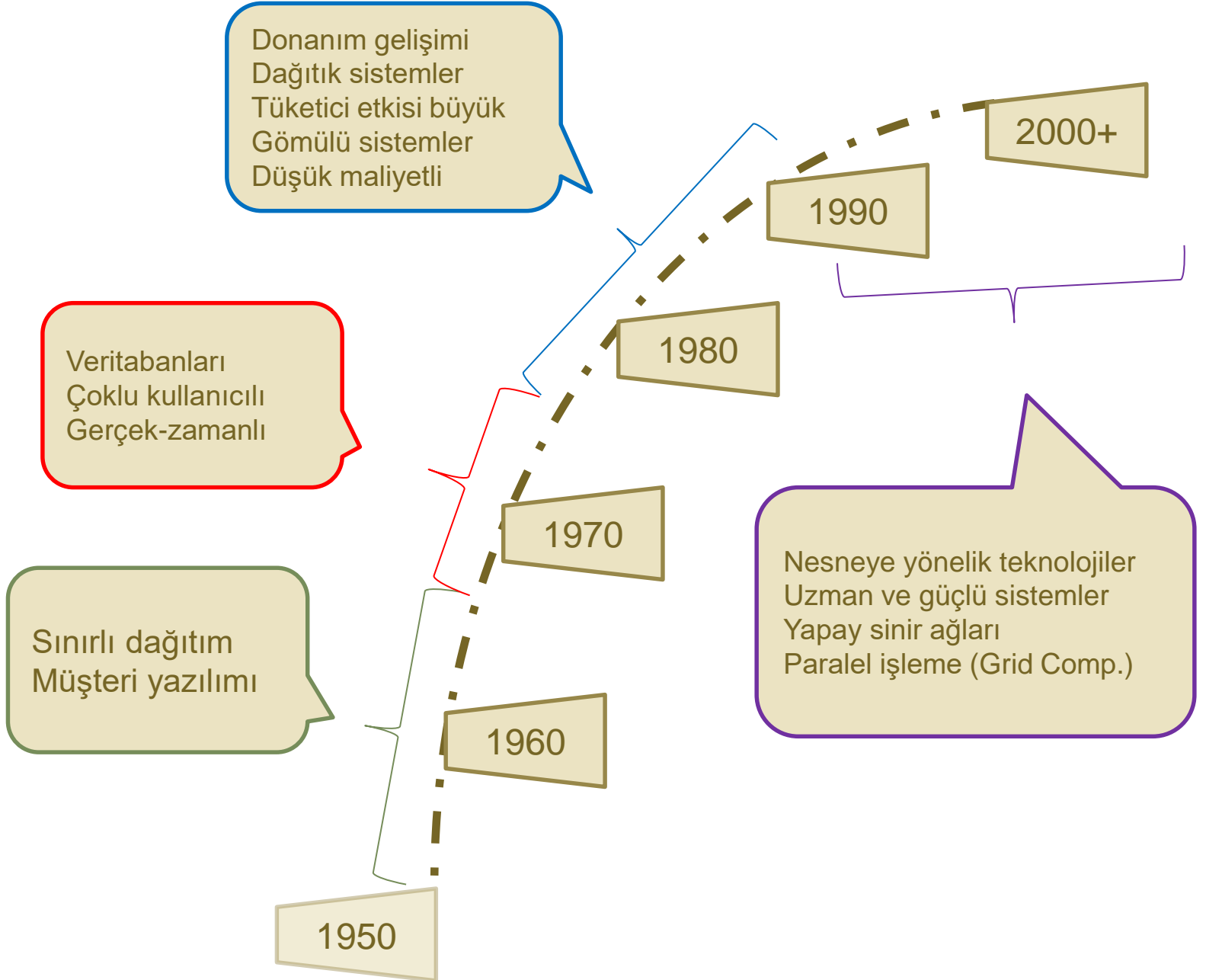
- İşletime alınan programın hemen ardından bakım çalışmaları sürekli olarak gündeme gelir.
- Hiçbir program bütünüyle her olasılık göz önüne alınarak sınanamaz, dolayısıyla hata ihtimali her zaman mevcuttur.
- İşletmeler doğaları gereği dinamik bir yapıya sahiptir, dolayısıyla mevcut sistemin sürekli olarak yeni istek ve gereksinimleri ortaya çıkar.
- Ortaya çıkan talepler ve değişiklikler aynı disiplin içerisinde sisteme eklenmelidir.

“Değişik bilgisayar bilimi teknolojilerinin ve kişilerin bir bilgi yada yazılım sistemi oluşturmak amacıyla bir araya getirilmesinde bir bütünleştirici gibi çalışır.”

“Yazılım mühendisi bir programcı değildir, ancak programcının tüm yeteneklerine sahiptir. “

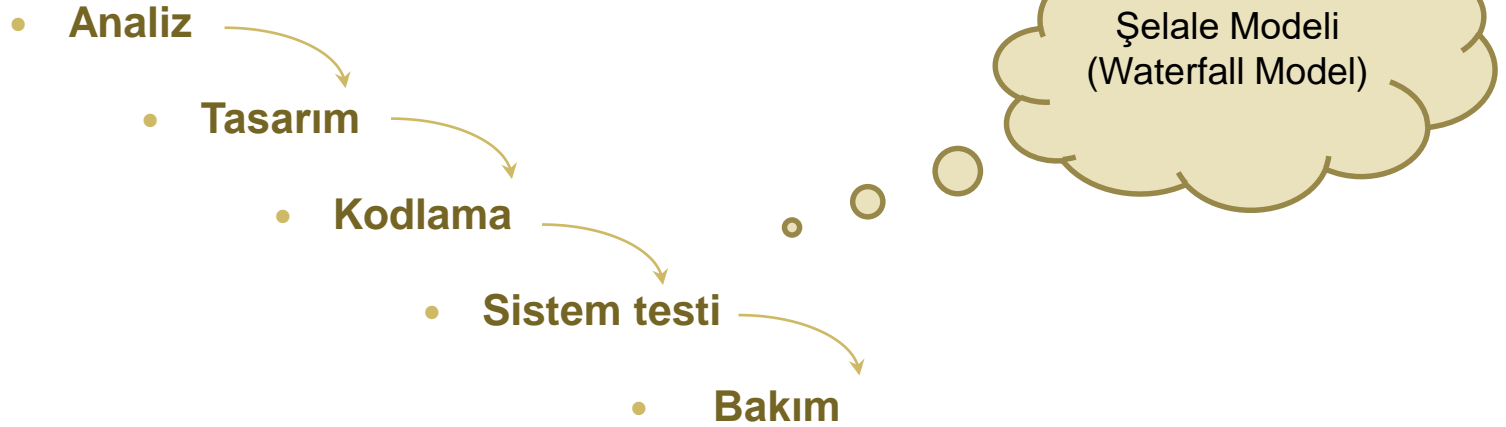
“Yazılım projelerinde temel hedefi, söz konusu üretimin az maliyet, yüksek nitelik ve talebe uygun yapılmasıdır.”

Yazılım Gelişim Süreci :



Yazılım geliştirme adımları

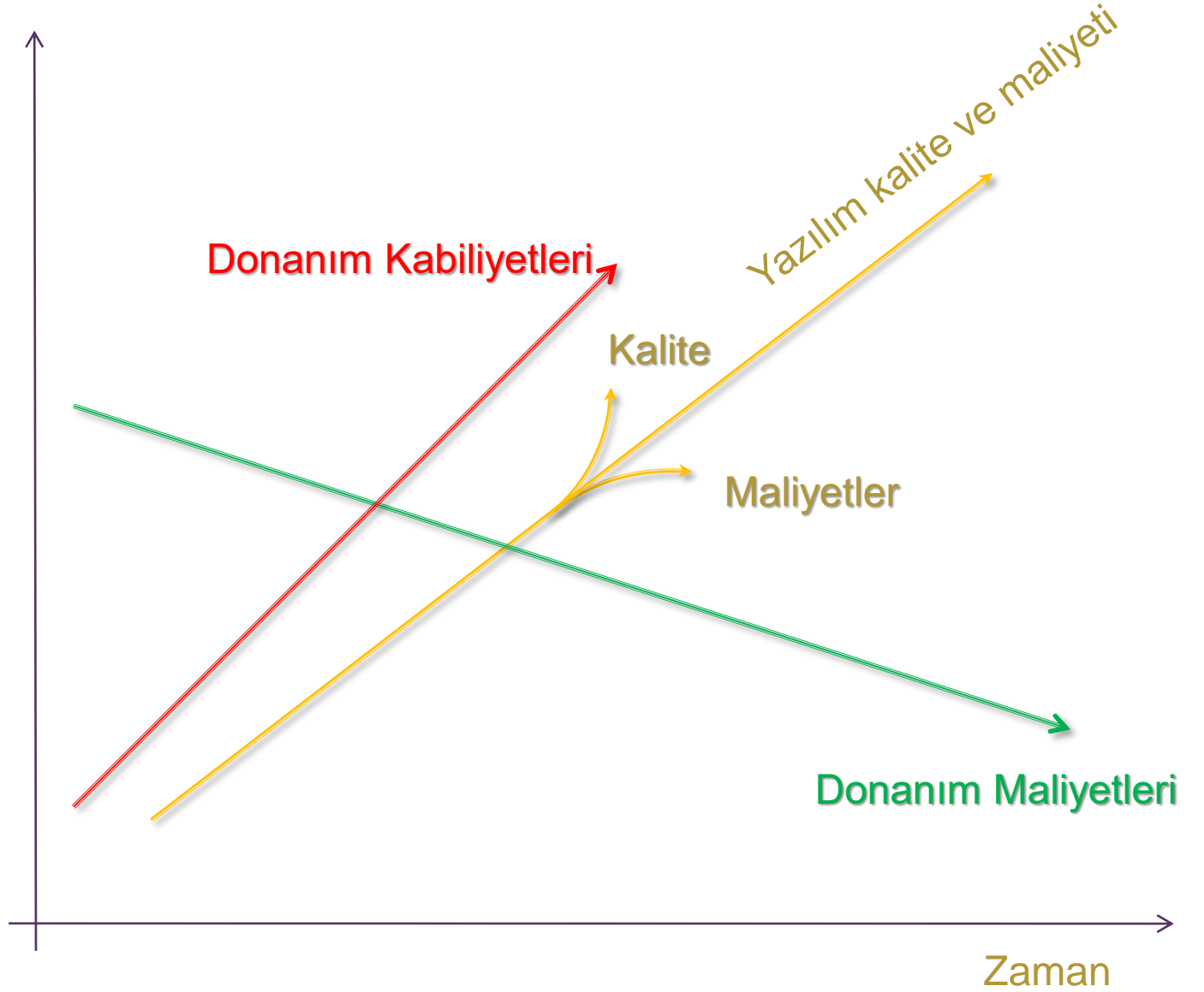
- Gereksinim analizi (requirements analysis)
 - Fonksiyonel ve fonksiyonel-olmayan gereksinimler (functional & non-func. reqs.)
- Tasarım (design)
 - Sistem tasarımı (system design): subsystems
 - Detaylı tasarım (detailed design)
- Kodlama ve birim testi (unit testing)
 - Birleşenlerin ayrı ayrı gerçekleştirilmesi ve birim testi
- Bütünleme testi ve sistem testi
 - (integration & system testing)
- Bakım ve güncelleme (maintenance)
 - Hataların giderilmesi, performans iyileştirme,
 - servislerin geliştirilmesi, değişikliklere uyum, vs.



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

- Yöntemler
- Araçlar
- Teknikler

“Yazılım üretiminde (Yaşam döngüsü) belirtilen aşamaların sistematik olarak izlenmesi ve gerçekleştirilebilmesi yazılım mühendisliği için ön şarttır.”



Problemler !

Büyük çaptaki uygulamalar ne kadar test edilsede, hiçbir zaman %100 hatadan arındırılamaz.

Yazılım geliştirme (çoğunlukla) ;

- Proje temellidir
- Yoğundur (bilgi yoğun)
- Genellikle bütçelerini aşar
- Öngörülen zamandan uzun süren
- Zorluklarla doludur
- Zaman alıcıdır
- Hesapta olmayan masraflar gerektirir
- Sürekli devam eden süreçtir.
- ...

Problemler !

- Değişik yetenekte bir çok personel (raportör, programcı, test sorumlusu, çözümleyici),
- Yeniliğe ve değişime tepki gösteren kullanıcı ve yöneticiler,
- Standart ve yöntem eksiklikleri,
- Yeterince tanımlanmamış, oldukça karmaşık kullanıcı beklentileri,
- Personel sirkülasyonunun fazla olması,
- Yüksek eğitim maliyetleri,
- Dışsal ve içsel kısıtlar (maliyet, zaman, işgücü),
- Verimsiz kaynak kullanımı,
- Mevcut yazılımlardaki yetersizlik ve kalitesizlik,
- Üretim maliyetlerinin yüksek olması,

Hatalar

Yazılım üretiminde Hataların Dağılımı :

- Mantıksal Tasarım %25
- İşlevsel Tasarım %15
- Kodlama %35
- Belgeleme ve Diğer işlevler %25

Yazılım Üretiminde Hata Düzeltme Maliyetleri :

- Çözümleme %1
- Tasarım %5
- Kodlama %15
- Test %30
- Kabul Testi %40
- Uygulama %100

Mühendislik !

Yazılımın mühendislik olarak ifade edilmesi demek ;

Belirli standartlara uyması ve **ölçülebilir olması** gerekir.

- Kalite standartları,
- Kalitenin ölçülebilmesi
- Verimin standartları,
- Verimin ölçülebilmesi

Kalite

- Hata sayısının düşük düzeyde olması
- Kullanıcı isterlerine cevap oluşturabilme (tamamını gerçekleştirebilme)
- Arızalar arası zamanın uzunluğu
- Destek ve gelişme.

Yazılımlar

- Sistem yazılımı
- Gerçek-zamanlı yazılım
- İş yazılımı
- Mühendislik ve bilimsel yazılım
- Gömülü yazılım
- Kişisel bilgisayar
- Yapay zeka yazılımı
- ...

Sınıflama

- İşlevlerine göre
- Zamana dayalı ve uygulama alanlarına göre
- Boyutlarına göre

Örnekler

- Hesaplama (Nümerik Çözümleme)
- süreç temelli (Gömülü sistemler)
- Veri isleme (finans sektörü)
- CAD (Sinyal işleme)
- Kural Temelli (Robotik, Yapay Zeka)

Yazılım Mühendisliğinde İçerik Bilgileri



Bilgisayar Temelleri

- Algoritmalar, veri yapıları, programlama dilleri
- İşletim sistemleri, matematik



Yazılım Mühendisliği

- Mühendislik disiplini, çözümleme, tasarım,
- Gerçekleştirim(kodlama), test, bakım, geliştirme



Yazılım Yönetimi

- Süreç yönetimi, Risk yönetimi, Kalite yönetimi
- Geliştirme yönetimi, Kazanım yönetimi



Yazılım Alanları

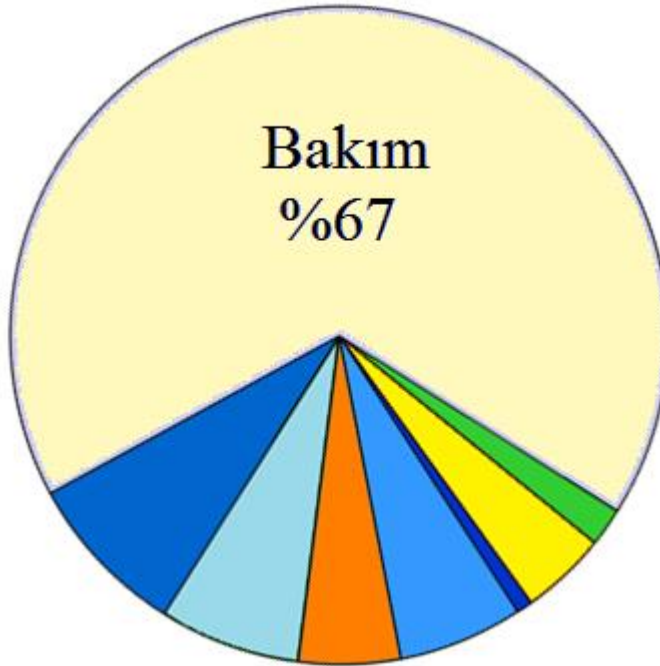
- Veri tabanı, mühendislik, Gömülü sistemler
- Yapay zeka, ...

Süreç bilgi alanları

- Üretimi ve işlemleri içine alan mühendislik disiplini
- Yazılım Mühendisliği Yönetimi
- Yazılım ihtiyaç Analizi
- Yazılım Yönetimi
- Yazılım Tasarımı
- Yazılım Yapılandırılması
- Yazılım Testi
- Yazılım Mühendisliği Altyapısı
- Yazılım Mühendisliği işlemi
- Yazılım Değerlendirme ve Bakımı
- Yazılım Kalite Analizi

Örnek meslekler ...

- Sistem Analisti
- Yazılım Mimarı
- Yazılım Proje Yöneticisi
- Yazılım Programcısı
- Yazılım Sistem Yöneticisi
- Yazılım Veri Tabanı Yöneticisi
- Yazılım Test elemanı
- Yazılım Ağ Uzmanı
- Yazılım Güvenlik Mühendisi
- Yazılım Konfigürasyon Yöneticisi
- Yazılım Kalite Yöneticisi
- ...



Göreceli maliyetlere göre yazılım yaşam döngüsü aşamaları.

İhtiyaç belirleme	% 2
Şartname belirleme	% 4
Planlama	% 1
Tasarım oluşturma	% 6
Gerçekleştirim(kodlama)	% 5
Test süreci	% 7
Entegrasyon süreci	% 8
Bakım süreci	% 67

TOP TECHNOLOGY TRENDS THAT YOU CAN LEARN



AI & ML



Web 3.0



Blockchain



Game Dev.



IOT



Cyber Security



5G



Edge Computing



Virtual Reality



Analytics



Big Data



Voice Tech



RPA



**Web/App
Development**



DevOps



Cloud Computing



Uygulama ...

Kaynak :

- Roger S. Pressman, Software Engineering – A Practitioner’s Approach, 6th Ed., McGraw Hill, International Edition, 2004,
- Prof. Dr. Ş.Sağiroğlu ders notları
- N.Y. Topaloğlu (Makale)
- Wikipedia