

Yazılım Mühendisliği

Prof. Dr. Oya Kalıpsız
Yıldız Teknik Üniversitesi
Bilgisayar Mühendisliği Bölüm Başkanı

Bilgisayar Mühendisliği Eğitiminin Amaçları

Bilgisayar mühendisliği eğitimi öğrencilere aşağıdaki bilgi ve becerileri kazandırmayı amaçlar:

- Temel bilimleri Bilgisayar Mühendisliği alanında kullanabilme
- Analitik düşünce ile mevcut sistemleri inceleme, iyileştirme ve geliştirmeye yönelik algoritmik çözümler üretebilme

Bilgisayar Mühendisliği Eğitiminin Amaçları

- İstenilen gereksinimleri karşılayacak sistemleri tasarlayabilme
- Tasarımları, deneysel yöntemler ile destekleyerek uygulayabilme
- Küresel ve toplumsal boyutlarda mühendislik alanındaki gelişmeleri takip edebilme

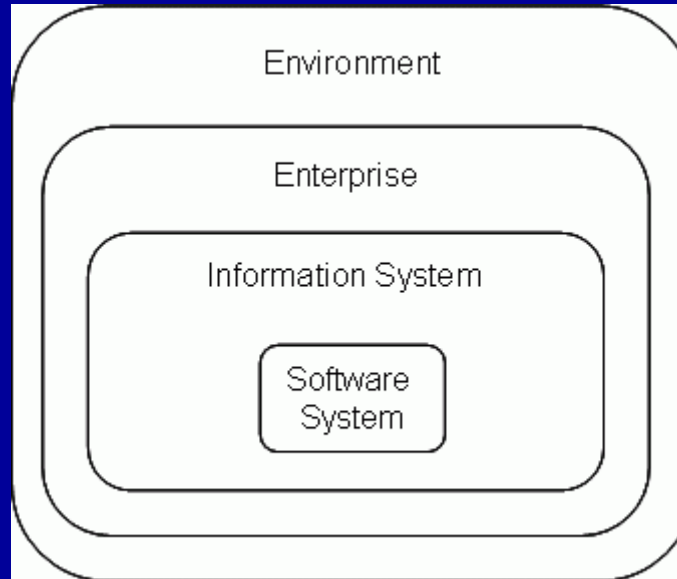
Bilgisayar Mühendisliği Eğitiminin Amaçları

- Yaşam boyu öğrenme gereğini algılama ve kendi kendine öğrenebilme
- Takım çalışması yapabilme
- Mesleki ve etik sorumluluklara sahip olma, yetki alabilme ve gereğini yerine getirebilme

Yazılım Mühendisliği

- Yazılım mühendisliği, bilgisayar bilimi, yönetim bilimi, ekonomi ve bilişim bilim dallarından yararlanmakta ve sorun çözümünde mühendislik yöntemlerini kullanmaktadır.

Yazılım Sisteminin Kurum İçindeki Yeri



Yazılım Mühendisliği Dersinin Hedefleri

- Yazılım mühendisliğinin önemi ve gereği
- Yazılım geliştirme süreci
- Yazılım Yeterlilik Olgunluk Modeli'nin (Capability Maturity Model, CMM) yapı ve düzeyleri
- Yazılım standartlarının yapısı ve tipleri

Yazılım Mühendisliği Dersinin Hedefleri

- CASE Teknolojisi
- Yazılım proje maliyeti tahmin yöntemleri, yazılım ölçümü kriterleri, yazılım proje planı hazırlama
- Yazılım kalite özellikleri ve yazılım kalite yönetimi teknikleri
- Yazılım bakımı kavramı

Yazılım Mühendisliği Dersinin Hedefleri

- Yazılım mühendisliğinde konfigürasyon yönetiminin yeri ve önemi, yazılım değişim kontrolü ve versiyon kontrolü yolları
- Yazılım gereksinim analizi kavramı, yazılım gereksinim spesifikasyonu hazırlama teknikleri
- Yazılım tasarım süreci ve tasarım notasyonları
- Nesne yönelimli (object-oriented) yaklaşımın özellikleri ve analiz ve tasarım yöntemleri

Yazılım Mühendisliği Dersinin Hedefleri

- Yazılım tasarım süreci ve tasarım notasyonları
- Nesne yönelimli (object-oriented) yaklaşımın özellikleri ve analiz ve tasarım yöntemleri

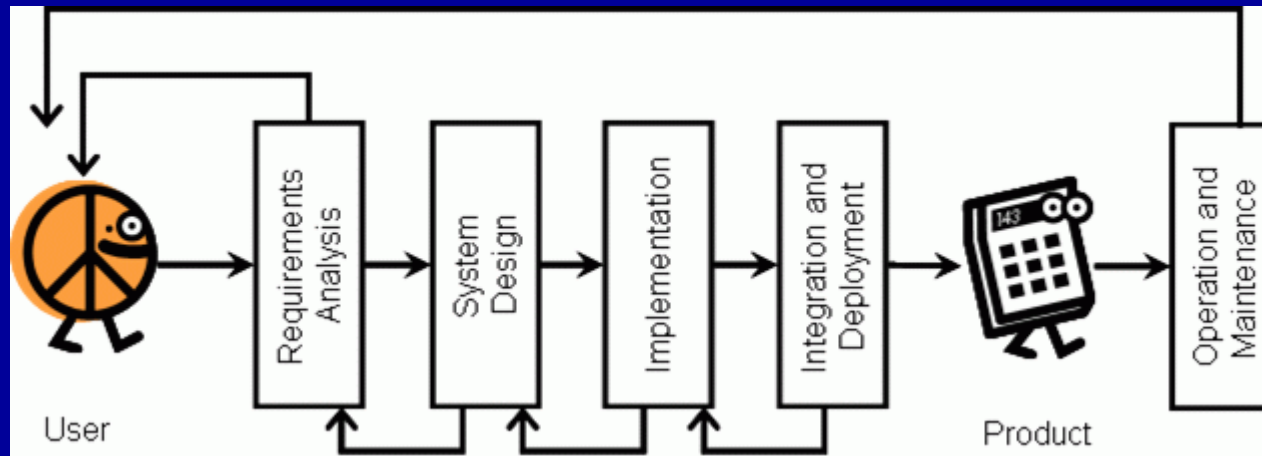
Yazılım geliştirme sürecinde gerçekleştirilen işlemler

- Esas işlemler
 - Gereksinim Analizi
 - Kodlama
 - Test
 - Bakım

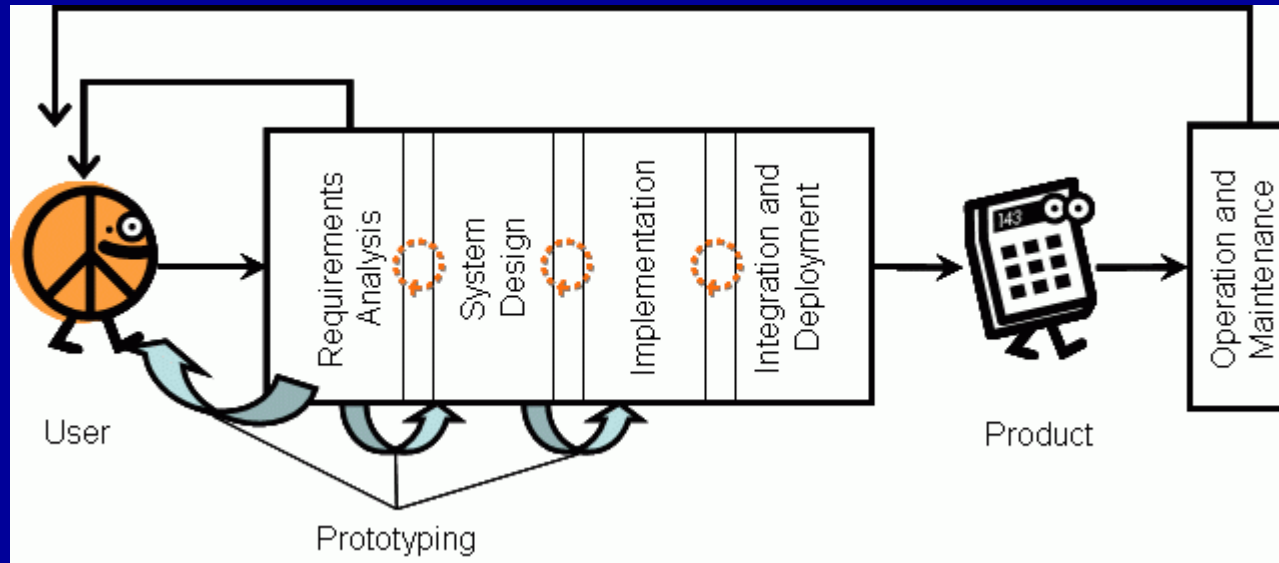
Yazılım geliştirme sürecinde gerçekleştirilen işlemler

- Destek İşlemler
 - Proje Yönetimi
 - Kalite Yönetimi
 - Yazılım Konfigürasyon Yönetimi
 - Risk Yönetimi

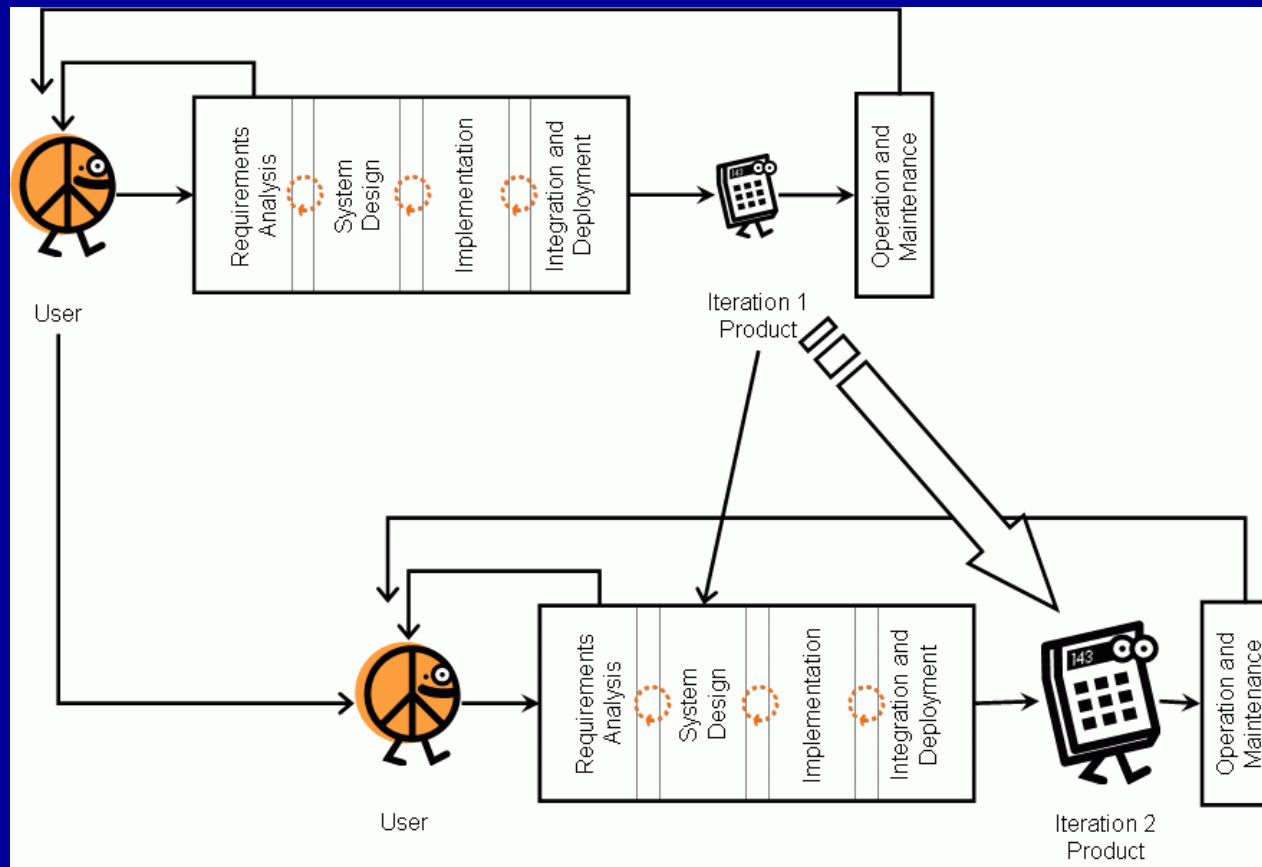
Yazılım Geliştirmede Klasik (Waterfall) Süreç



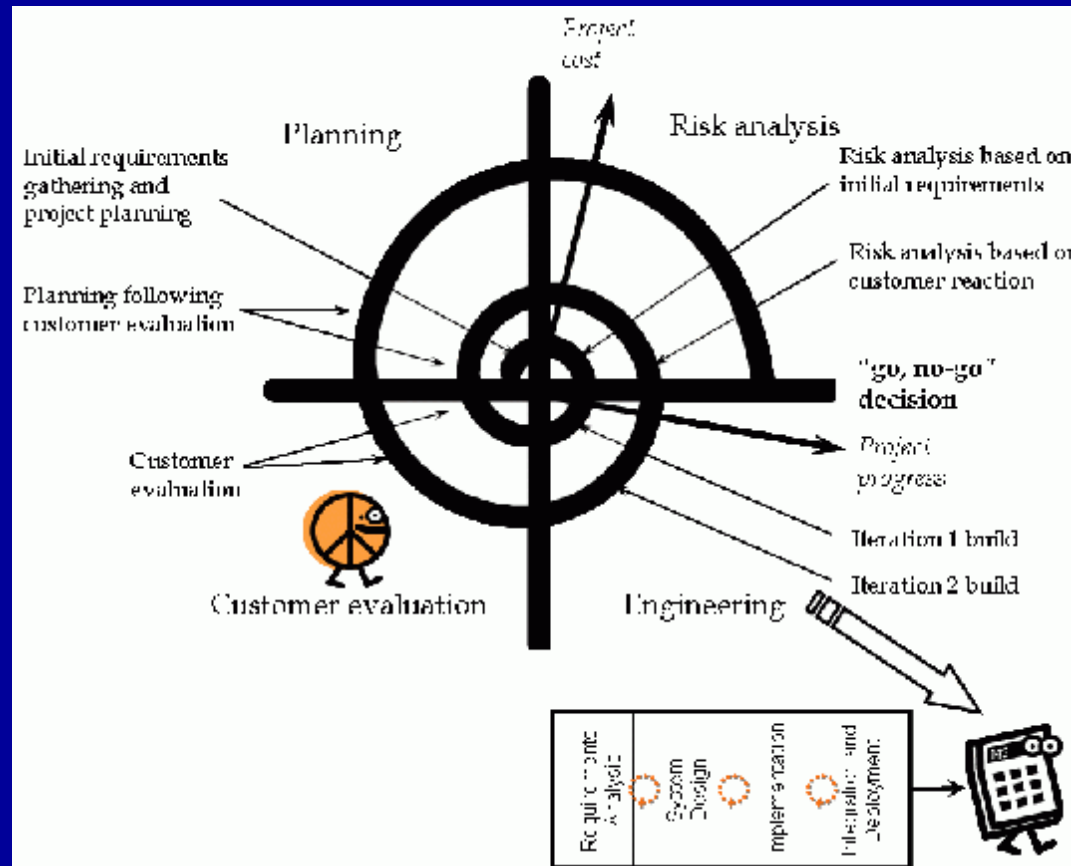
Yazılım Geliřtirmede Prototip Kullanımı



Artımlı Yazılım geliştirme



Yazılım Geliştirmede Spiral Yöntem



Yazılım Proje Yönetimi

- Yazılım proje yönetimi, yazılım projelerinde kaynakların en etkin biçimde kullanılmasını sağlamaya yöneliktir.



- Etkin proje yönetimi gerçekleştirmek için, insan kaynakları, problem ve yazılım geliştirme sürecindeki işlemler göz önünde bulundurulmalıdır.

Yazılım Proje Yönetimi Hedefleri

- Yazılım proje yönetimi elemanları
- Yazılım ölçümü kriterleri
- Yazılım ölçüm yöntemleri
- Yazılım proje maliyeti tahmin yöntemleri
- Risk analizine yaklaşımlar
- Yazılım proje planı hazırlama

Yazılım Geliştirme Planı

Yazılım geliştirme planının:

- Ne?
- Ne kadar?
- Ne zaman?

sorularını yanıtlaması beklenmektedir.

Yazılım Geliştirme Planı Aşamaları

- Sorunun tanımlanması
- Çözüm yolunun karşılaştırılması
- Geliştirme sürecinin plânlanması
- Proje organizasyonu
- Proje zamanlaması

Yazılım Kalite Yönetimi

- Yazılım kalite özellikleri
- Yazılım kalite ölçümü
- Kalitenin sağlanması
- Yazılımın sinanması
- Yazılım sinama teknikleri

Yazılım Kalite özellikleri

- **Doğruluk:** spesifikasyonlara uygunluk ve müşteri isteklerini karşılama derecesi
- **Güvenirlilik:** tasarlanan işlevleri istenilen duyarlılıkla yerine getirme olanağı
- **Bağlanabilirlik:** bir sistemin diğerine bağlanabilme olanağı.

Yazılım Kalite özellikleri

- **Verimlilik:** programın işlevlerini yerine getirebilmesi için gerekli bilgi-işlem kaynaklarının ve kodlamanın gideri
- **Güvenlik:** yetkisiz kişilerin yazılıma ya da veriye girişi önleme olanağı
- **Kullanışlılık:** öğrenme, işletme, girdi hazırlama ve çıktı yorumlamada kolaylık derecesi

Yazılım Kalite özellikleri

- **Hata bulma kolaylığı:** hatanın yerini bulma ve düzeltme olanağı
- **Esneklik:** programda değişiklik yapma kolaylığı
- **Sinama kolaylığı:** programın doğruluğunu sınamadaki kolaylık

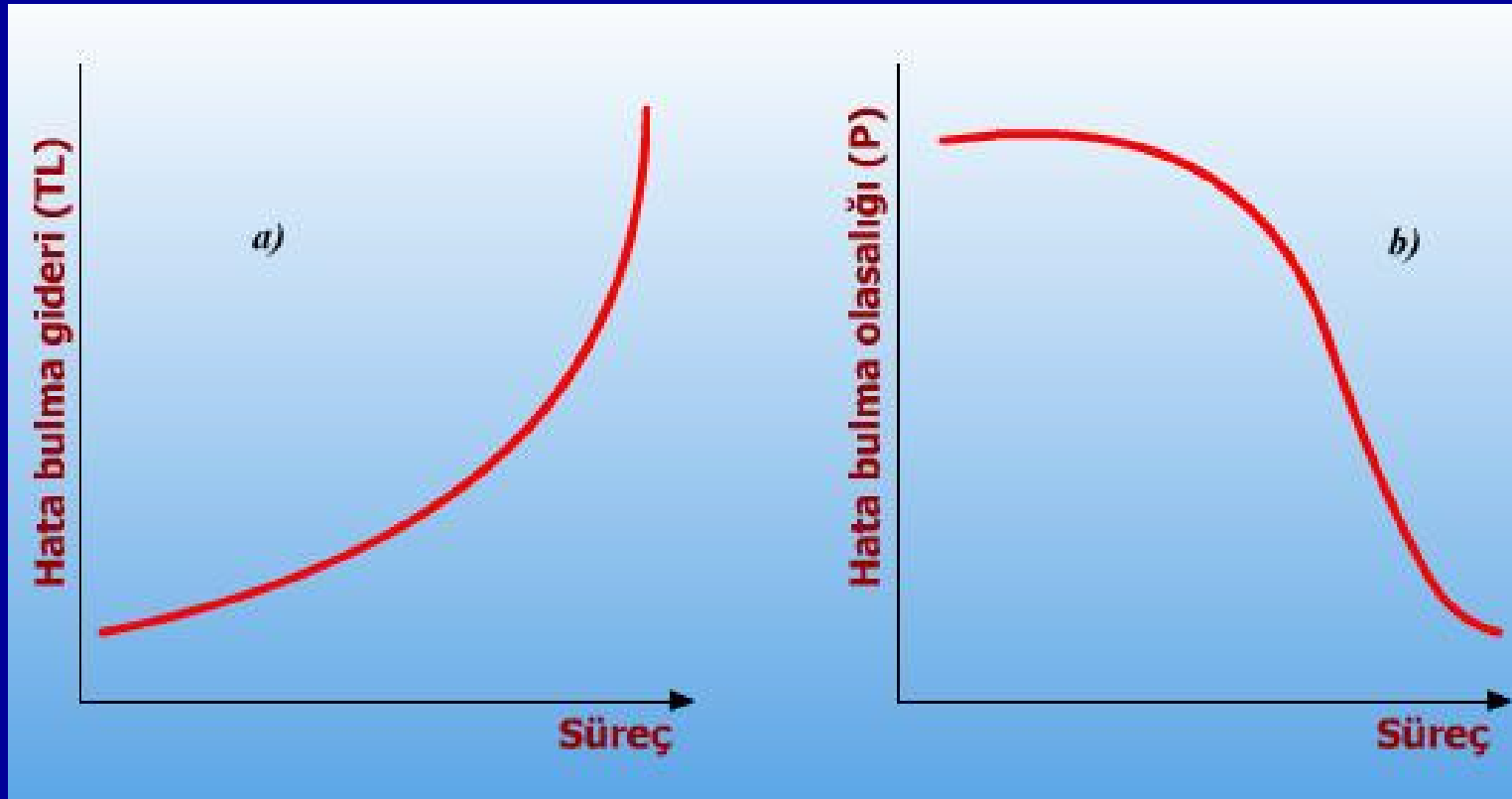
Yazılım Kalite özellikleri

- **Taşınabilirlik:** programın farklı donanımlarda ve değişik yazılım sistemi ortamlarında kullanılma olanağı
- **Tekrar kullanılabilme:** programın veya bir bölümünün başka bir uygulamada kullanılma olanağı

Yazılım Geliştirme Sürecinde Kaliteyi Sağlama Örnekleri

- Gereksinim analizi aşamasında: sistemin kolayca kullanılıp öğrenilir olması, istenmeyen hallerin plânlı olarak değiştirilmesine çaba gösterilmesi
- Modüllere ayırmada: taşınabilirlik ve bağlanabilirlik özelliklerine önem verilmesi
- Ayrıntılı tasarımda: doğruluk faktörüne ağırlık verilmesi
- Sınama aşamasında: doğruluk ve olduğunca performans vb. diğer faktörlerin ölçümüne çaba gösterilmesi

Yazılım Kalitesini Sağlamak İçin, Hata Ve Eksiklerin Anında Bulunması Ve Düzeltilmesi



Yazılım Kalitesi Aşamaları

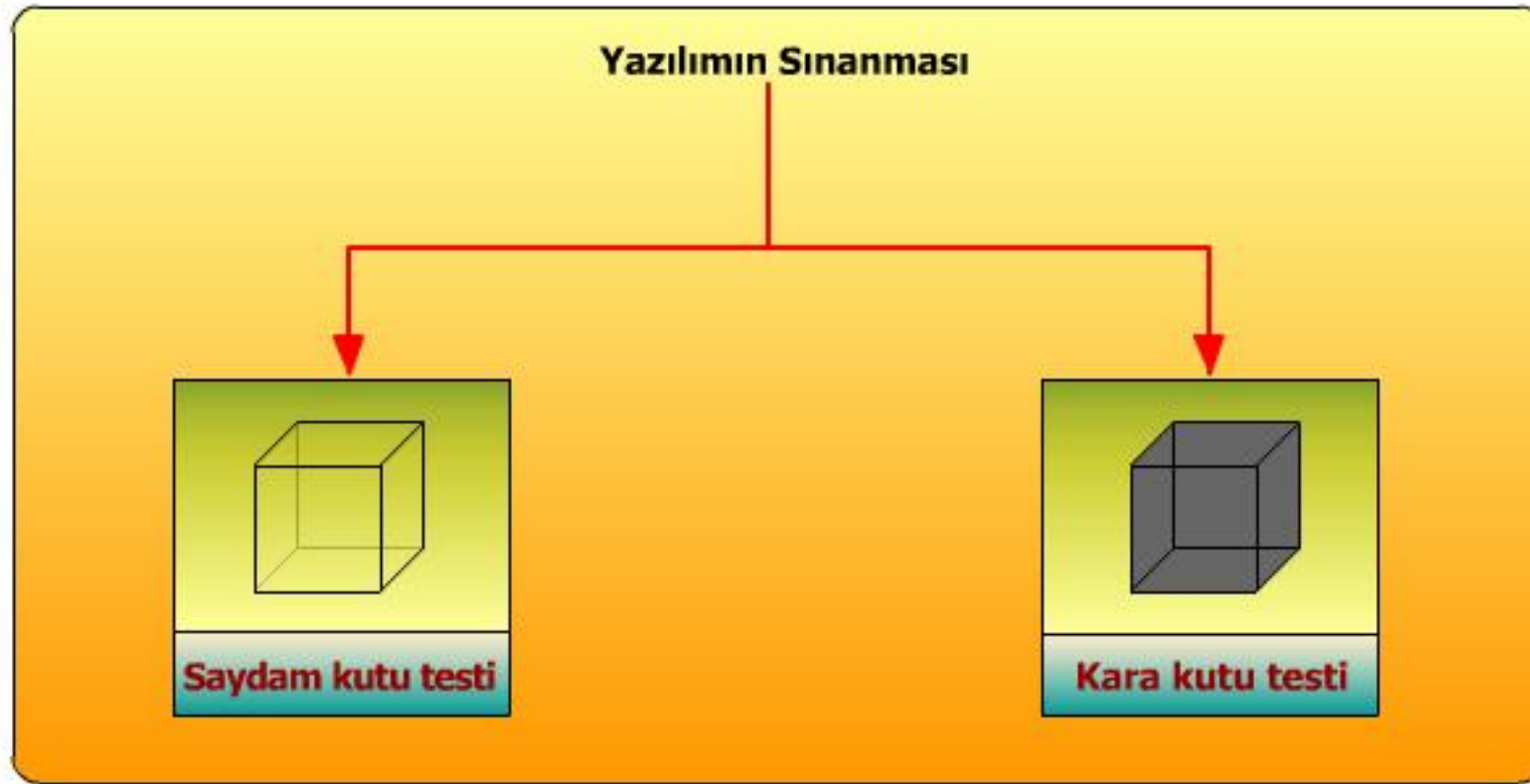
- Geliştirmenin plânlanması aşamasında kalite kontrolü yöntem ve araçlarının belirlenmesi
- Geliştirme sürecinin durak noktalarında yapılanların gözden geçirilmesi
- Kaynak programın sinanması

YAZILIMI SINAMA TEKNİKLERİ

Yazılımın sinanması;

- Saydam kutu testi,
- Kara kutu testi

olarak iki ayrı yoldan yapılmaktadır.



KONFIGÜRASYON YÖNETİMİ SİSTEMİ

