**Proje Nedir**

Belirli bir yerde belirli bir zaman ve bütçe çerçevesinde bir başlama ve bitiş noktasına sahip öncede belirlenmiş amaçlara ulaşılmasını sağlayacak olan faaliyetler topluluğudur.

* Belirli bir yerde uygulanmalı
* Takvimi olmalı
* Bütçesi olmalı
* Tanımlanmış faaliyetleri olmalıdır
* Amaç Nedir
* Sorun analizi ve hedef analizi yapılmalıdır
* -Proje ile ortadan kaldırmayı iyileştirmeyi önerdiğiniz sorunlara işaret etmeli
* Araştırma Süreci
* Literatür taraması
* Alandaki çalışmalar iyi okunmalı
* Literatür taraması sonunda araştırma sorusu belirlenmeli
* Analiz yöntemi belirlenmeli
* Veriler toplanmalı
* Analiz yapılmalı
* Analiz sonucuna elde edilen bulgular yorumlanmalı
* Proje raporu yazılmalıdır

**Sınıf Diyagramı**

**Sınıf Tanımlaması**

|  |
| --- |
| Araba |
| + TekerlekSayisi : byte = 4  - Model : String |
| + ArizaKontrol ( ) : bool  + ArizaKontrol ( içeriArac : A ) : bool  + Init ( ) |

+ işareti public seviyesinde olduğunu gösterir.

# işareti protected seviyesinde olduğunu gösterir

- işareti private

**Sınıflar Arası İlişki**

Satın Alma

|  |
| --- |
| Müşteri |
|  |

|  |
| --- |
| Ürün |
|  |

|  |
| --- |
| Sipariş |
|  |

**Türetme**

|  |
| --- |
| Kullanıcı |

|  |
| --- |
| Yönetici |

|  |
| --- |
| Müşteri |

**Arayüz**

|  |
| --- |
| Sınıf 1 |
| - Özellik 1 |
| + Do 1 ( ) : bool  + Do 2 ( ) : bool |

|  |
| --- |
| Arayüz |
| + Do 1 ( ) : bool  + Do 2 ( ) : bool |

**- Sistem Çözümleme –**

=> Sistem çözümleme üretim sürecinin başlangıcıdır.

Mevcut sistemin incelenmesi sırasında temel hedef gereksinimlerin saptanmasıdır

Bu işlemden sonra sistem için modelleme yapılır. Mantıksal model

=> **Gereksinim Nedir**

Bir sistem geliştirilirken kullanıcın sistemin işlevleri ile ilgili beklentileri sistemin amaçlarını oluştur.

Gereksinim sistemin amaçlarını yerine getirme yeteneği olan bir özellik ya da belirtim olarak tanımlanmaktadır.

=> **Gereksinim Türleri**

* **İşlevsel Gereksinimler**: Sistem ile çevresi arasındaki iletişimi belirleyen gereksinimlerdir.
* **İşlevsel Olmayan Gereksinimler**: Kullanıcının sorunundan bağımsız olarak çözülmesi gereken sorunlar işlevsel olmayan gereksinimlerdir.

Gereksinimler 9’a ayrılır

* Fiziksel Çevre
* Arayüzler
* Kullanıcı ve İnsan etmeni
* İşlevsellik
* Belgeleme
* Veri
* Kaynaklar
* Güvenlik
* Kalite Güvencesi

Gereksinimler 3 amaca hizmet eder

* Geliştiricilerin, müşterilerin sistemin nasıl çalışmasını istediklerini anlamalarını sağlar.
* Tasarımcılara, sonuç sistemin ne işlevsellik ve özellikte olacağını söyler.
* Gereksinimler, sınama ekibine, kullanıcıya sunulan sistemin istenen sistem olduğuna ikna etmek için neler göstermeleri gerektiğini söyler.

Gantt Diyagramı  
–**Gantt çizelgesi**, Henry Gantt tarafından geliştirilen ve süreç yönetiminin planlı olması amacıyla oluşturulmuş bir çizelgedir.  
-Çubuk şeması olarak da adlandırılır.  
-Projelerde planlanan görevlerin zaman ölçekli bir hat üzerinde grafiksel olarak gösterilmesi temeline dayanmaktadır.  
-Dünyada en yaygın proje yönetim aracı olma özelliğini sürdürmektedir (yaklaşık %80).  
-Birinci Dünya Savaşında büyük kargo gemisi inşa projelerinin planlamasında kullanılmıştır.

Yazılım Geliştirme Aşamaları

Karmaşık yazılımları geliştirmek ve bakımını yapmak çok masraflı ve zordur. Bu yüzden, yazılımlar yazılım mühendisleri tarafından nizami olarak planlı bir [proje](https://tr.wikipedia.org/wiki/Proje) şeklinde geliştirilmektedir. Bu nizami geliştirme planına "yazılım geliştirme süreci" (İngilizce: *software development process*) adı verilmektedir. Yazılım geliştirme süreci, zamanlamaya dayalı, içerik olarak bölünmüş ve görselleştirilmiş aşamalardan oluşmaktadır. Bu sayede yazılım adım adım ve planlı bir şekilde geliştirilmektedir. Bu aşamalar birbirleri ile bağlantılı olarak geliştirilmektedir.

Başlıca yazılım geliştirme aşamaları şunlardır:

Çekirdek aşamalar:

* Planlama
* Analiz
* Dizayn
* Programlama
* Test

Destekleyici aşamalar:

* Talep yönetimi
* Proje yönetimi
* Kalite yönetimi
* Yapılandırma yönetimi
* Yazılım sunumu
* Dökümantasyon
* Ayrık yapılandırma

Doğrulama ürünü belirtilen koşullara göre doğru geliştiriyor muyuz’a bakıyoruz? Geçerlemede ise doğru ürünü geliştirdik mi?

Doğrulama genel olarak planlar, gereksinimler, tasarım, kod ve test adımlarını değerlendirilirken, geçerlemede ise yazılımın kendisi değerlendirilir,

Doğrulama genel olarak gözden geçirmeler, muayene ve statik analiz aktiviteleri içerirken, geçerleme fonksiyonel veya kabul testleri gibi daha dinamik aktivitleri içerir,

Doğrulama genelde hiç ya da az kod koşumuna ihtiyaç duyarken, geçerleme kod muhakkak koşulur,

Doğrulama daha alt seviye ve geliştirme ortamında yapılan bir aktivite olsa da geçerleme daha üst seviye ve paydaşların da dahil olduğu bir aktivitedir.