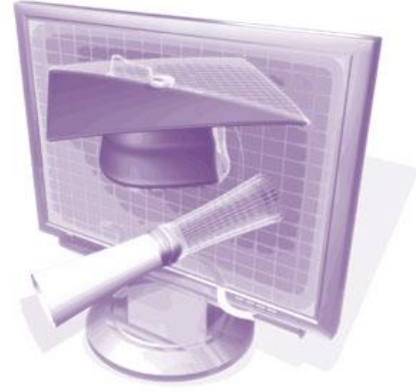


# Yazılım Mühendisliği Eğitimi



## Bir yazılım mühendisliği (lisans) mezununun sahip olması gereken yetenekler şunlardır :

1. Yazılım ürünleri geliştirmek için **bir takımın parçası** olarak çalışmak,
2. **Kullanıcı gereksinimlerini** belirlemek ve onları **yazılım gereksinimlerine** çevirmek,
3. Çelişen amaçları düzenlemek, maliyet, zaman, bilgi ve organizasyon kısıtlamaları içinde kabul edilebilir uzlaşmalar bulmak.
4. Bir veya daha çok uygulama alanı için, etik, sosyal, yasal ve ekonomik ilgileri bütünleştiren mühendislik yaklaşımlarını kullanarak uygun çözümler tasarlamak.
5. Yazılım tasarımı, geliştirilmesi, gerçekleştirimi ve doğrulanması için bir temel sağlayan mevcut teorileri, modelleri ve teknikleri anlamak ve uygulayabilmek.
6. Tipik bir yazılım geliştirme ortamında etkin olarak çalışmak, gerekli olduğunda liderlik yapabilmek ve kullanıcılarla iyi iletişim kurabilmek.
7. Yeni modelleri, teknikleri ve teknolojileri öğrenebilmek.

## Yazılım Mühendisliği Eğitimi Bilgi Alanları ;

1. Temeller
2. Profesyonel Uygulama
3. Gereksinimler
4. Tasarım
5. Yazılım oluşturma
6. Yazılım sına ve doğrulama
7. Yazılım gelişimi
8. Yazılım Süreci
9. Yazılım Kalitesi
10. Yazılım Yönetimi

## Yazılım Mühendisliği Eğitimi Bilgi Alanları

### **Temeller:**

Yazılım mühendisliğinin ürettiği ürünlerin niteliklerini anlatan teorik ve bilimsel temellerden, bu ürünleri modellemeyi ve tanımlamayı kolaylaştıran matematiksel temellerden ve öngörülebilir sonuçlar üreten ana ilkelerden oluşur.

Buradaki ana nokta, kaynakları belirlenmiş bir amaca dönüştürmek için mühendislik tasarımı ve mühendislik biliminin uygulanmasıdır.

## Yazılım Mühendisliği Eğitimi Bilgi Alanları

### **Profesyonel Uygulama:**

Yazılım mühendislerinin, yazılım mühendisliğini profesyonel ve etiğe uygun olarak uygulayabilmeleri için sahip olmaları gereken bilgi, beceri ve davranışlarla ilgilidir.

**Profesyonel uygulamalar bilgisi teknik iletişim, psikoloji ve sosyal ve mesleki sorumlulukları içerir.**

## Yazılım Mühendisliği Eğitimi Bilgi Alanları

### **Gereksinimler:**

Bir sistemin amacını ve hangi içerikte kullanılacağını tanımlar.

Gereksinimler, kullanıcıların gerçek gereksinimleri ile yazılım ve diğer bilgisayar teknolojileri arasında köprü oluşturur.

Gereksinimlerin belirlenmesi, sistemin fizibilite çalışmasını, kullanıcıların gereksinimlerinin analizi, sistemin ne yapacağını ve ne yapmayacağını kısıtlamalar göz önünde alınarak belirlenmesini ve bu bilginin kullanıcılar tarafından doğrulanmasından oluşur.

## Yazılım Mühendisliği Eğitimi Bilgi Alanları

### **Tasarım:**

Bir bileşenin veya bir sistemin nasıl gerçekleştirileceğini belirlemek için kullanılan teknikler, stratejiler, gösterimler ve desenlerle ilgilidir.

Tasarım, kaynaklar, performans, güvenilirlik ve güvenlik gibi kısıtlamalar göz önüne alınarak işlevsel gereksinimlere uygun olmalıdır.

Ayrıca, yazılım bileşenleri arasındaki içsel ara yüzler, mimari tasarım, veri tasarımı, kullanıcı ara yüzü tasarımı, tasarım araçları ve tasarımın değerlendirilmesi de bu alanın kapsamındadır.

## Yazılım Mühendisliği Eğitimi Bilgi Alanları

### **Yazılım Oluşturma:**

Bu alan, tasarımda belirlenmiş yazılım bileşenlerinin geliştirilmesiyle ilgili bilgileri içermektedir.

Bu kapsamda, bir tasarımın bir gerçekleştirim diline çevrilmesi, bileşen sınamaları ve program belgelemeleri incelenmektedir.



## Yazılım Mühendisliği Eğitimi Bilgi Alanları

### **Yazılım Sınama ve Doğrulama:**

Elde edilen programın hem belirlenen gereksinimleri sağladığını hem de gerçekleştirimin beklenenlere uygun olduğunu kontrol etmek için statik ve dinamik sınama teknikleri kullanır.

Statik teknikler, yazılımın tüm yaşam döngüsü boyunca elde edilen gösterimlerin analizi ve kontrolüyle ilgilenirken, dinamik teknikler sadece gerçekleştirilmiş sistemi içerir.

## **Yazılım Gelişimi:**

Yazılımın kullanıma verilmesinin öncesindeki ve sonrasındaki aşamalarda etkin bir maliyetle desteklenmesini sağlar.

Bu destek, gelişen sistemi oluşturan versiyonların veya sürümlerin her biri için hazırlık aktivitelerine gerek duyar.

Bu aktiviteler, planlama, ölçüt desteği, regrasyon sınama ve karmaşıklık kontrolünü içermektedir.

Bu aktiviteleri desteklemek için kullanılan teknikler, program anlama, sürüm planlaması, değişiklik tanımlaması, yeniden mühendislik, tersine mühendislik, bakım, sistemin kullanımına son verilmesini içerir.

## Yazılım Mühendisliği Eğitimi Bilgi Alanları

### **Yazılım Süreci:**

Yaygın olarak kullanılan yazılım yaşam döngüsü süreç modellerinin tanımlanmasıyla ilgili bilgileri ve kurumsal süreç standartlarını;

**yazılım süreçlerinin tanımlanmasını,**

**gerçekleştirilmesini,**

**ölçülmesini,**

**bakımını,**

**yönetimini,**

**değiştirilmesi ve iyileştirilmesini ;**

ve yazılım geliştirme ve bakımı için gereken teknik ve yönetsel aktiviteleri gerçekleştirmek için tanımlı bir süreç kullanımını kapsamaktadır.

## **Yazılım Yönetimi:**

Tüm yazılım yaşam döngüsü aşamalarının planlanması, düzenlenmesi ve izlenmesiyle ilgili bilgileri içermektedir.

Yazılım geliştirme projelerinin başarısı için, farklı organizasyonel birimlerdeki işlerin koordinasyonu için, yazılım versiyonlarının bakımı için, kaynakların gerekli oldukları zaman var olabilmesi için, projedeki işlerin uygun olarak bölünebilmesi için, iletişimin kolaylaşması için kritik önemdedir.

**Yazılım Kalitesi:**

Yazılım geliştirmenin ve bakımın tümünü etkileyen ve tümünden etkilenen bir kavramdır.

Hem geliştirilen ürünlerin kalitesini hem de bu ürünleri geliştirmek için kullanılan süreçlerin kalitelerini içerir.

**Ürün kalite nitelikleri,**

1. kullanılabilirlik,
2. güvenilebilirlik,
3. güvenlik,
4. bakıma uygunluk,
5. esneklik,
6. Etkinlik,
7. performans

gibi kriterleri kapsamaktadır.

## Temel Bileşenleri

1. Ekip olarak çalışmak için gerekli bilgiyi sağlayan **Etkin İletişim ve Grup Çalışması Yetenekleri.**
2. Öğrencileri gerçek hayat problemlerine hazırlayan **Bir Uygulama Alanında Deneyim.**
3. Öğrencileri gereksinimlerin değişmesi, proje yönetimi, konfigürasyon yönetimi, araç kullanımı gibi konulara hazırlayan **Bir Ekip Projesi.**
4. Öğrencileri gerçek hayat ortamına hazırlayan **Çalışma Ortamında Deneyim.**
5. Ders notu veya ders gibi belirli bir düzende sağlanmayan bilgiyi aramak, değerlendirmek ve kullanmak için gerekli beceriyi kazandırmayı amaçlayan **Yaşam boyu Öğrenme Araçları.**

6. Yazılım ve donanım ürünleri hakkında temel teknik bilgi ve becerileri sağlayan **Bilgisayar Bilimi Temelleri**. Bunlar, programlama dilleri, modelleme, veritabanları, işletim sistemleri, ağ sistemleri, algoritmalar olarak sayılabilir.
7. Yazılımı ve ilgili belgelemeyi oluşturmak ve bakımı yapmak için gerekli teknik bilgi, yetenek ve araçları sağlayan **Yazılım Mühendisliği Temelleri**. Bunlar arasında, yazılım süreçleri, yaşam döngüsü modelleri, yazılım ölçütleri, mimari ve tasarım yöntemleri bulunmaktadır.
8. Genel sistem ilkelerinin, ekonominin ve mühendislerin görev ve sorumluluklarının anlaşılmasını sağlayan **Mühendislik Uygulamaları ve Etiği**

## Yazılım mühendisliği

6 seneye yayılmış bir Eğitim programıdır.

Bu program 2 bölüme ayrılır:

- Temel Yazılımcılık
- Uzman Yazılımcılık



## Temel yazılımcılık :

Temel yazılımcılığın süresi 4 Dönemdir.

Bu dört dönem içerisinde yazılımcı adayı bilgisayar yazılımcılığı hakkında temel bilgiler edinir.

- Algoritma geliştirme,
- Sistem analizi,
- Veri tabanı analizi,

## Yazılım dilleri hakkında genel bilgiler....

- C ----- C++ ----- Visual C ---- Borland C#
- Assembly (8/16/32/64)
- Sistem analizi
- Veri Tabanı analizi
- Delphi ----- Pascal----Delphi400
- Java --- JavaScript---- Perl ---- Web Tasarım --- CQI --- CGI --- ClientSERVER
- Grafik
- Animasyon
- Projeksiyon --Promasyon --- Prodüksiyon
- . net
- AS400 ----- RPG400
- Ağ yönetimi
- Cracker
- Hacker
- ...

### **Uzman Yazılımcılık:**

Yazılım Mühendisi adayı yukarıda belirtilen konulardan herhangi birini seçer. Daha sonra bu konu hakkında ayrıntılı bilgi edinirler ve uzmanlaşırlar.

**“Böylece üretilen işin kalitesi gün geçtikçe artar.”**

**Orta büyüklükte bir Yazılım şirketi  
en az şu yazılımcıları bulundurmak zorunda kalacak.**

- Sistem Analisti Yazılım Mühendisi
- Hacker Yazılım Mühendisi
- Cracker Yazılım Mühendisi
- Algoritma Yazılım Mühendisi
- Veri Tabanı Analisti Yazılım Mühendisi
- Grafiker Yazılım Mühendisi
- Animasyon Yazılım Mühendisi
- Assembly Yazılım Mühendisi
- Herhangi bir dil kullanan yazılım Mühendisi

Listelerin eksik kaldığı en önemli noktalardan birisi fikirlerimizin birbirleriyle olan bağlantılarının net bir şekilde görülememesidir.

Özellikle belirli bir konuda kafa yorarken veya bilgi toplarken “zihin haritası” (Mind Map) kullanmak işinize yarayabilir.

Zihin haritalarının faydalarını ve kullanılabilecekleri alanları aşağıda listeledim.

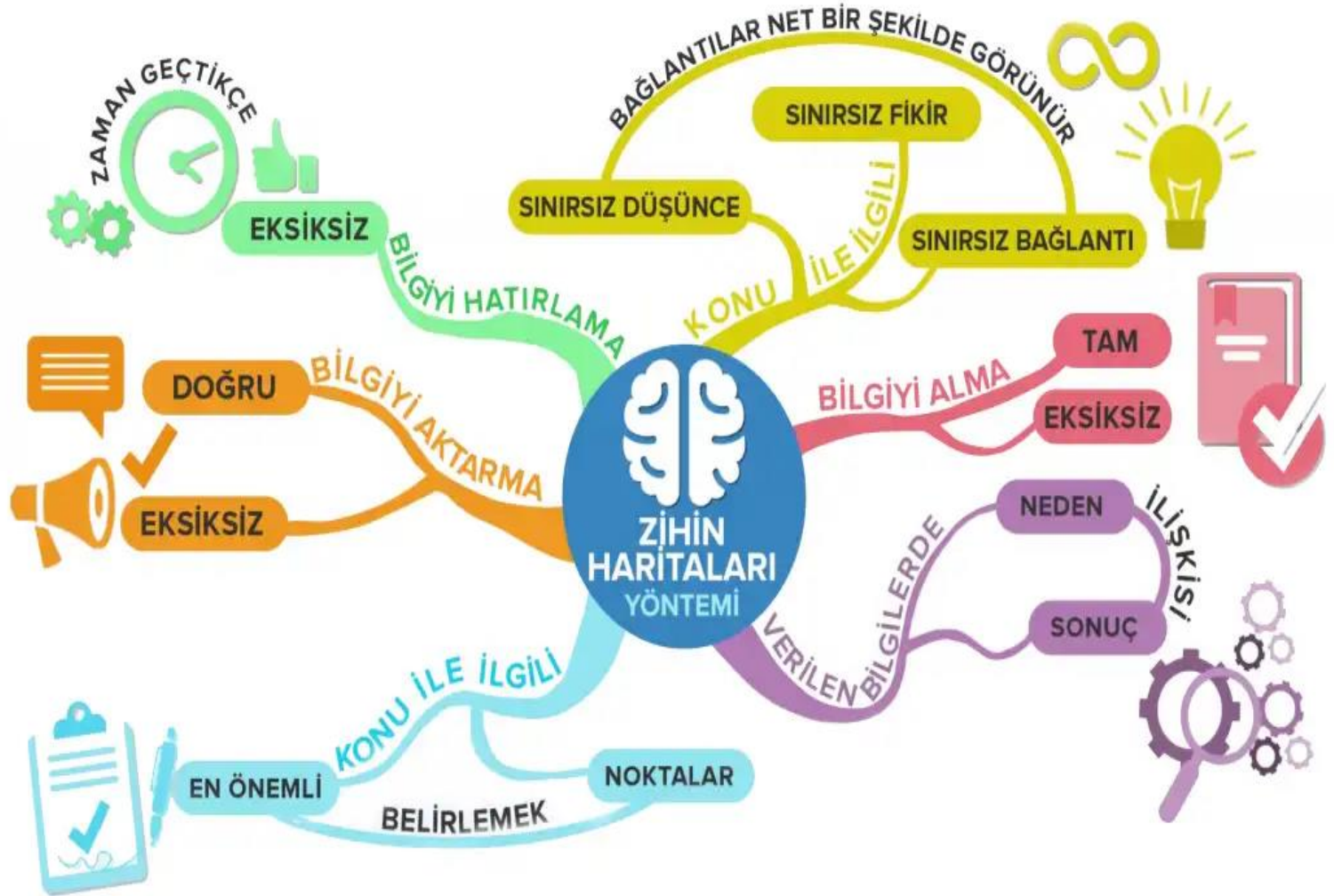
### **Zihin haritalarının faydaları:**

Verimliliği artırmak  
Yaratıcı düşünme yeteneğini geliştirmek  
Düşüncelerinizi yapılandırmak  
Bilgileri görselleştirmek  
Veriyi daha etkin biçimde yönetmek

### **Zihin haritalarının kullanılabileceği alanlar:**

Kişisel bilgileri yapılandırmak  
Bilgiyi kayıt altına almak  
Hayatınızı yönetmek  
Beyin fırtınası yapmak için  
Planlama yapmak için  
Yönetim becerilerini geliştirmek  
Toplantıları daha verimli hale getirmek için  
Sınavlara çalışırken  
Ders içeriği hazırlarken  
Ezber yaparken











## Uygulama ...

Kaynak :

N.Y. Topaloğlu

Ege Üniversitesi

Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü