

Yazılım Mühendisliği

Dr. Öğr. Üyesi

Bilge Kağan Dedetürk

Yazılım Mühendisliği

Ders Linkleri

- bilgededeturk@erciyes.edu.tr
- <https://github.com/kagandedeturk/YazilimMuhendisligi>
- <https://avesis.erciyes.edu.tr/bilgededeturk/documents>

Software Requirements

Ders Kitabı

- Yazılım Mühendisliği (10. Baskı) – Ian Sommerville

Yazılım Mühendisliğine Giriş

- “Our civilization runs on software” (Bjarne Stroustrup)
- Yazılım, cep telefonları, kişisel bilgisayarlar, elektronik cihazlar gibi çok çeşitli cihazları kontrol eder.

Bu derste neler öğreneceğiz?

- Her yazılım ile ilgili karakterin rolü: geliştirici, test uzmanı, tasarımcı, yönetici
- Karşılaşılabilecekleri sorunlar ve bu sorunları nasıl çözebilecekleri
- Biz onlardan ne beklemeliyiz?
- Onlar bizden ne bekliyor?
- En iyi uygulamaları, araçları ve teknikleri nelerdir?
- İyi bir iş çıkardıklarını nasıl anlarız?
- Çalışmalarının kalitesini ölçebilir miyiz?
- Günlük kullandıkları dil.

Yazılım Mühendisliğini Neden Öğrenmeliyiz?



Yazılım Mühendisliğini Neden Öğrenmeliyiz?

Yeni teknikleri, en iyi uygulamaları ve yeni araçları öğrenmek



Daha verimli



Daha çok zaman



Daha çok yazılım



Daha çok kazanç

Yazılım Mühendisliğini Neden Öğrenmeliyiz?



VS



Yazılım Mühendisliğini Neden Öğrenmeliyiz?

Yazılım Krizi

- Bu terim, 60'lar ve 70'lerde büyük yazılım sistemleri geliştirirken karşılaşılan birçok zorluğu ifade eder.
- Yazılım projelerinin başarısızlıkları, olması gerekenden daha sık meydana gelir.
- Bugün hâlâ geçerlidir.

Tamamlanamayan sistemler

Kaçırılan son teslim tarihleri

Aşılan bütçeler

Gereken tüm işlevleri yerine getirmeyen bir sistem

Çalışan bir sistem fakat kullanımı zor

Değiştirmesi zor bir sistem

Kullanıcıların güven kaybı

Yazılım Mühendisliğini Neden Öğrenmeliyiz?

- Küçük yazılım sistemleri geliştirirken kullanılan tekniklerin, daha büyük ve karmaşık sistemleri yönetmek için ölçeklenememesi.
- Yazılım geliştirme projelerinin planlanması ve organizasyonu için herhangi bir çerçevenin olmaması.

Yazılım Mühendisliğini Neden Öğrenmeliyiz?

Mevcut zorluklar:

- Güvenilir, yüksek kaliteli yazılımlar geliştirme ihtiyacı
- Konseptten dağıtım ve operasyona kadar hızlı geri dönüş talebinin artması
- Artan yazılım karmaşıklığı
- Yazılım sistemlerinin çeşitliliği
- Bileşen tabanlı yeniden kullanımda ve otomatik kod üretiminde verimliliğin artırılması ihtiyacı
- Değişimi yönetmek

Yazılım Mühendisliğini Neden Öğrenmeliyiz?

- Büyük/karmaşık sistemler inşa etmek için yapılandırılmış bir yaklaşım
- Sorunun yönetilebilir bölümlere ayrılması için bir yöntem gereklidir.
- Görevle ilgili ortak bir anlayış ve doğru iletişim esastır.
- Büyük sistemler inşa etmek geniş kapsamlı grup çalışmasını içerir.
- Her grup üyesinin kendi görevini ve bu görevin diğer görevlerle nasıl etkileşime geçtiğini anlaması gerekir.
- Gruplar ve bireyler, üzerinde mutabık kalınan ortak bir dilde iletişim kurmalıdır.



Mühendislik Nedir?

- **Mühendislik:** Bir amacı gerçekleştirmek için makineler, yapılar ve diğer öğeleri tasarlamak ve inşa etmek amacıyla bilimsel ilkelerin kullanılması.
 - Yazılım mühendisliği ise bir amacı gerçekleştirmek için yazılımı tasarlamak ve inşa etmek amacıyla bilimsel ilkeleri kullanır.
-

Yazılım Mühendisliği Nedir?

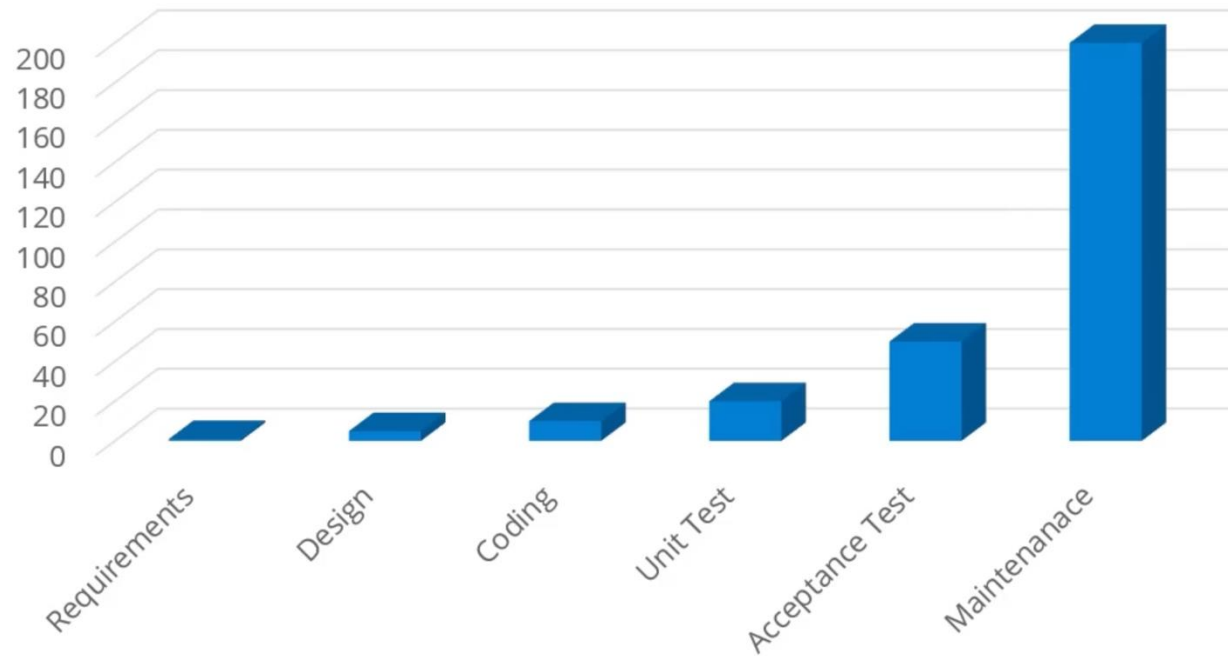
IEEE'nin yazılım mühendisliği tanımı:

- (1) The application of a **systematic, disciplined, quantifiable** approach to the development, operation, and maintenance of software; that is, the application of engineering to software.
- (2) The study of approaches as in (1).

Yazılım Mühendisliğinin Faydaları

- Projelerde günlük görevleri tamamlarken güçlü ve kabul görmüş teknikler kullanmak.
- Projelerimizi tamamlamak için maliyet ve zaman çizelgesini doğru bir şekilde tahmin etmek.
- Ürünlerimize sürdürülebilirlik, güvenilirlik vb. gibi istenen özellikler kazandırmak.

Cost of Repair



Life Cycle Phase	Relative Cost of Repair
Requirements	1
Design	5
Coding	10
Unit Test	20
Acceptance Test	50
Maintenance	200 - 1000