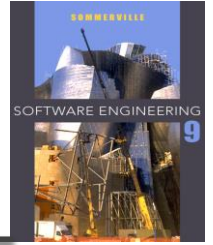


# Bölüm 1- Giriş

## Ders 1

# Konular

---



## ✧ Profesyonel Yazılım Geliştirme

- Yazılım Mühendisliği ne demek?

## ✧ Yazılım Mühendisliği Etikleri

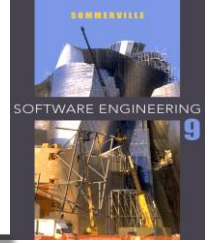
- Yazılım Mühendisliğini etkileyen etik konulara kısa bir giriş

## ✧ Örnek Yazılım Projeleri

- Ders kapsamında kullanılacak olan yazılım projelerinin özetleri

# Yazılım Mühendisliği

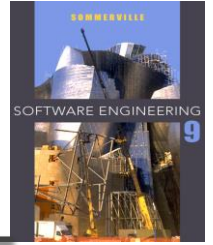
---



- ✧ Bütün gelişmiş ulusların ekonomileri yazılıma bağlı
- ✧ Her gün daha fazla sistem yazılım ile kontrol ediliyor
- ✧ Yazılım mühendisliği profesyonel yazılımlar geliştirmek için gerekli olan teoriler, yöntemler ve araçlar ile ilgilenir
- ✧ Gelişmiş ülkelerin milli hasıllarında yazılım harcamalarının büyük payı vardır.

# Yazılım Maliyetleri

---



- ✧ Yazılım maliyetleri genelde donanım maliyetlerini geride bırakır. Bir PC üzerindeki yazılımların maliyeti genelde PC'nin donanım maliyetinden yüksektir.
- ✧ Bir yazılımın bakım maliyeti geliştirme maliyetinden yüksektir. Uzun ömürlü sistemler için, bakım/onarım maliyetleri geliştirme maliyetlerinin birkaç katıdır.
- ✧ Yazılım Mühendisliği, maliyet-etkin yazılım geliştirme ile ilgilenir.



## ✧ Genel Ürünler

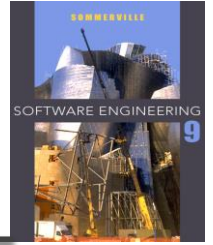
- İlgilenen herhangi bir alıcıya satılan bağımsız ürünler
- Örnek – grafik programları, proje yönetim araçları; bilgisayar destekli tasarım araçları; diş hekimi randevu yazılımı gibi.

## ✧ Özelleşmiş Ürünler

- Belirli bir müşteri tarafından kendi ihtiyaçlarını karşılaması için ısmarlanmış yazılımlar.
- Örnek – gömülü kontrol sistemleri, hava trafik kontrol yazılımı, Trafik gözlem sistemi.

# Ürün Tanımlama (Ürünün neler yapabileceğine karar verme)

---



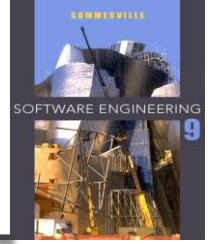
## ✧ Genel Ürünler

- Yazılımın neler yapabileceği yazılım geliştirici tarafından belirlenir ve yazılım üzerindeki değişikliklere (güncellemelere) geliştirici karar verir.

## ✧ Özelleşmiş Ürünler

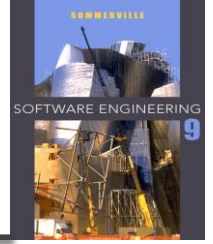
- Yazılımın neler yapabileceği müşteri tarafından belirlenir ve yazılım üzerindeki değişikliklere (güncellemelere) müşteri karar verir.

# Yazılım Mühendisliği ile İlgili Sıkça Sorulan Sorular



Soru	Cevap
Yazılım nedir?	Bilgisayar programları ve ilgili dokümanlar.
İyi bir yazılımın özellikleri nelerdir?	İyi bir yazılım, istenen fonksiyonelliği ve performansı kullanıcıya sunmalıdır. Bakımı yapılabilir, güvenilir ve kullanışlı olmalıdır.
Yazılım Mühendisliği nedir?	Yazılım Mühendisliği, yazılım üretiminin bütün yönleri ile ilgilenen bir mühendislik alanıdır.
Temel Yazılım Mühendisliği aktiviteleri nelerdir?	Yazılım tanımlama, yazılım geliştirme, yazılım doğrulama ve yazılım evrimi.
Bilgisayar Bilimi ile Yazılım Mühendisliği arasındaki fark nedir?	Bilgisayar Bilimi teori ve temellere odaklıdır. Yazılım Mühendisliği ise kullanışlı bir yazılımın geliştirilmesi ve kullanıcıya teslimi konusundaki pratik uygulamalarla ilgilenir.
Sistem Mühendisliği ile Yazılım Mühendisliği arasındaki fark nedir?	Sistem Mühendisliği; donanım, yazılım ve süreç mühendisliğini içeren bilgisayar tabanlı sistemlerin bütün yönleri ile ilgilenir. Yazılım Mühendisliği, bu genel süreçlerin bir parçasıdır.

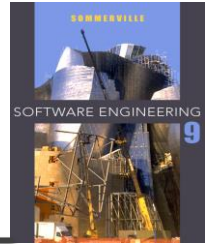
# Yazılım Mühendisliği ile İlgili Sıkça Sorulan Sorular



Soru	Cevap
Yazılım Mühendisliğinde karşılaşılan ana zorluklar nelerdir?	Çeşitliliğin artması, daha kısa teslim sürelerinin istenmesi ve güvenilir yazılım geliştirme talepleri.
Yazılım Mühendisliğinin maliyetleri nelerdir?	Kabaca %60 geliştirme maliyetleri, %40 test maliyetleridir. Özel yazılımlar için yazılım güncelleme ve değişikliklerinin maliyeti çoğu zaman geliştirme maliyetlerini aşar.
En iyi Yazılım Mühendisliği yöntemi nedir?	Bütün yazılım projelerinin profesyonel olarak yönetilmesi ve geliştirilmesi için değişik tipteki sistemlere değişik teknikler uygun düşer. Örneğin, oyunlar prototipleme ile geliştirilirken; güvenliği önemli olan kontrol sistemleri tam olarak analiz edildikten sonra geliştirilirler. Bu nedenle bir yöntemin diğer bir yöntemden iyi olduğunu söyleyemeyiz.
Web, Yazılım Mühendisliği üzerinde ne gibi değişiklikler yapmıştır?	Web, yazılım servislerinin ulaşılabilirliğini arttırmış ve dağıtık servis tabanlı yazılımlar geliştirilmesini sağlamıştır. Web tabanlı sistemler programlama dillerinde önemli ilerlemelere neden olmuş ve yazılımların yeniden kullanılabilirliğini arttırmıştır.

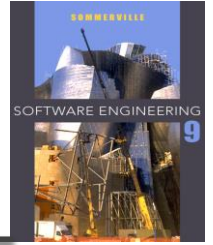


# İyi bir yazılımın temel özellikleri



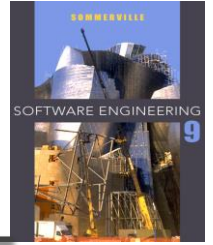
Ürün karakteristiği	Açıklama
Sürdürülebilirlik/Bakım	Yazılım, müşterinin ihtiyaçları doğrultusunda gerekli değişikliklerin yapılabilmesine olanak tanıyacak şekilde geliştirilmelidir. Değişen iş ortamlarının kaçınılmaz ihtiyacı olduğu için bu özellik kritik bir özelliktir.
Güvenilirlik ve Güvenlik	Güvenilir bir yazılım fiziksel veya ekonomik bir zarara yol açacak hatalara neden olmamalıdır. Kötü amaçlı kullanıcılar sisteme ulaşamamalı ve zarar verememelidir.
Etkinlik	Yazılım hafıza ve işlem gücü gibi sistem kaynaklarını gereksiz yere kullanmamalıdır. Etkinlik kavramı cevap verme süresi, işlem süresi, hafıza kullanım verimliliği gibi kavramları barındırır.
Kabul edilebilirlik	Yazılım, kimin için geliştirildiyse o kişiler için kabul edilebilir olmalıdır. Yani, yazılım anlaşılabilir, kullanışlı ve kullanıcılar tarafından hali hazırda kullanılan sistemler ile uyumlu olmalıdır.

# Yazılım Mühendisliği



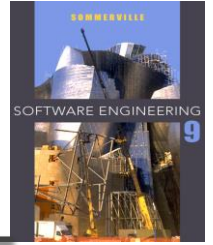
- ✧ Yazılım Mühendisliği yazılım üretiminin ilk aşamalarından kullanıma giriş sonrasındaki bakım/güncelleme aşamasına kadarki bütün yönleri ile ilgilenen bir mühendislik disiplini.
- ✧ Mühendislik disiplini
  - Organizasyonel ve finansal kısıtları göz önünde bulundurarak problemleri çözmek için gerekli teorileri ve yöntemleri kullanmak.
- ✧ Yazılım üretiminin bütün yönleri
  - Yalnızca geliştirme aşamasının teknik işlemleri değildir. Aynı zamanda yazılım üretimini destekleyecek proje yönetim ve geliştirme araçları, yöntemleri gibi şeylerdir.

# Yazılım Mühendisliğinin Önemi



- ✧ Her geçen gün çok daha fazla sayıda insan ve grup ileri yazılım sistemlerine ihtiyaç duymakta. Bu nedenle güvenilir sistemleri ekonomik olarak ve hızlı bir biçimde geliştirmek zorundayız.
- ✧ Genellikle uzun vadede yazılım mühendisliği yöntemlerini kullanarak yazılım geliştirmek, kişisel amaçlar için program yazarken yaptığımız gibi doğrudan kodlama yapmaktan daha ucuzdur. Çoğu sistemdeki maliyetlerin büyük kısmı, yazılım kullanıma girdikten sonra yazılım üzerinde yapılması istenen değişikliklerden kaynaklanır.

# Yazılım geliştirme aşama aktiviteleri



- ✧ **Yazılım tanımlama**, müşterilerin ve mühendislerin üretilecek yazılımın ne yapacağını kararlaştırdıkları ve yazılımın çalışmasındaki sınırları (kısıtları) belirledikleri aşamadır.
- ✧ **Yazılım geliştirme**, yazılımın tasarlandığı ve geliştirildiği aşamadır.
- ✧ **Yazılım doğrulama**, yazılımın müşterinin istediği gibi olup olmadığının kontrol edildiği aşamadır.
- ✧ **Yazılım evrimi** (bakım/güncelleme), yazılımın değişen müşteri istekleri ve piyasa koşulları doğrultusunda değiştirildiği aşamadır.

# Çoğu yazılımı etkileyen genel konular



## ✧ Heterojenlik

- Artan bir biçimde sistemlerin, değişik tipteki bilgisayarlar ve mobil cihazlardan oluşan bir ağ üzerinde çalışmaları gerekliliği

## ✧ İş ve sosyal değişiklikler

- Gelişen ekonomi ve yeni teknolojiler ile birlikte iş dünyası ve toplum değişmekte ve var olan yazılımlarını değiştirerek hızlı bir biçimde yeni yazılımların geliştirilmesine ihtiyaç duymaktalar.

## ✧ Güvenlik ve güven

- Yazılım bütün hayatımızı her yönü ile kuşatmış olduğu için yazılıma güvenebilmemiz önemlidir.

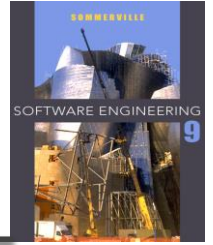
# Yazılım mühendisliği çeşitlilikleri

---



- ✧ Bir çok yazılım sistemi vardır be bu yazılım sistemlerine uygulanabilecek evrensel bir yazılım teknikleri kümesi bulunmamaktadır.
- ✧ Yazılım mühendisliği yöntemleri ve araçları, geliştirilmekte olan uygulamanın tipine, müşteri isteklerine ve yazılım geliştirme ekibinin bilgi ve tecrübesine bağlıdır.

# Uygulama Tipleri



## ✧ Tek başına çalışan (stand-alone) uygulamalar

- Bu uygulamalar tek bir cihaz üzerinden çalışan uygulamalardır. Gereksinim duydukları bütün fonksiyonelliğe sahiptirler ve bir ağa bağlanma ihtiyaçları yoktur.

## ✧ İnteraktif işleme dayalı uygulamalar

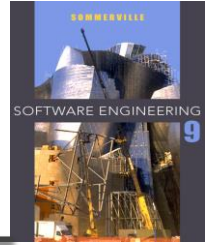
- Uzaktaki bir bilgisayarda çalışan ve kullanıcı tarafından bir bilgisayar veya terminal aracılığı ile ulaşılan uygulamalardır. E-ticaret uygulamaları gibi web uygulamaları örnek olarak gösterilebilir.

## ✧ Gömülü kontrol sistemleri

- Donanımsal cihazları kontrol eden ve yöneten yazılım kontrol sistemleridir. Sayıca muhtemelen diğer tipteki sistemlerden daha çok gömülü sistem vardır.

# Uygulama tipleri

---



## ✧ Toplu işlem sistemleri

- Büyük miktarlardaki verileri işlemek için tasarlanan iş sistemleridir. Anlamlı sonuçlar üretmek için büyük sayılardaki giriş verisini işlerler.

## ✧ Eğlence sistemleri

- Birincil olarak kişisel amaçlar için üretilmiş ve insanları eğlendirmeye yarayan sistemlerdir.

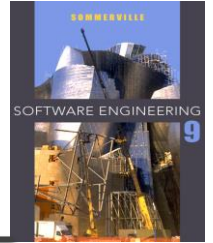
## ✧ Modelleme ve simülasyon sistemleri

- Bilim adamları ve mühendisler tarafından geliştirilen ve fiziksel olguları, nesne etkileşimlerini modellemeye yarayan sistemlerdir.



# Uygulama tipleri

---



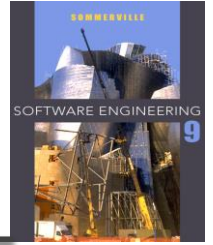
## ✧ Veri toplama sistemleri

- Bulundukları çevrelerden sensörler yardımı ile veri toplayan ve bu veriler işlenmek üzere başka sistemlere gönderen sistemlerdir.

## ✧ Sistemlerin sistemleri

- Diğer yazılım sistemlerinin bileşimi olan sistemlerdir.

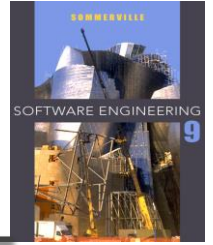
# Yazılım mühendisliğinin temelleri



- ✧ Bazı temel prensipler, geliştirme tekniğine bakılmaksızın bütün yazılım sistem tiplerine uygulanır:
- Sistemler, yönetilebilir ve anlaşılabilir geliştirme süreçlerini kullanarak geliştirilmelidirler. Elbette, değişik tip yazılımlar için değişik süreçler kullanılabilir.
  - Güvenilirlik ve performans bütün yazılım sistemleri için çok önemlidir.
  - Yazılım tanımının ve ihtiyaçların (yazılımın ne yapması gerektiği) anlaşılır ve kontrol edilebilir olması çok önemlidir.
  - Mümkün olduğunda, yeni yazılım geliştirmeden önce, zaten geliştirilmiş olan yazılımların kullanılması.

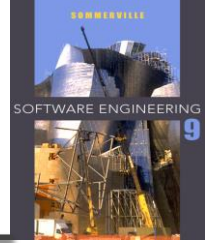
# Yazılım mühendisliği ve web

---



- ✧ Web, lokal sistemlerden daha çok yazılımın çalıştırıldığı ve organizasyonların kullanımına yöneldiği bir platform haline geldi.
- ✧ Web servisleri (daha sonra açıklanacak) uygulamaların fonksiyonelliğine web üzerinden erişim imkanı sunar.
- ✧ Bulut bilişim, uzaktan çalışan uygulamaların sunulduğu bir bilgisayar hizmetidir.
  - Kullanıcılar yazılım satın almaz; kullanımına göre para öderler.

# Web yazılım mühendisliği



- ✧ Web tabanlı sistemlerin oluşturulmasında yazılımın yeniden kullanılabilirliği baskın bir yaklaşımdır.
  - Bu sistemleri geliştirirken, daha önceden var olan yazılım bileşenlerinin ve sistemlerinin nasıl kullanılabileceğini düşünürsünüz.
- ✧ Web tabanlı uygulamalar parça parça geliştirilmeli ve uygulamaya konulmalı
  - Günümüzde genel olarak kabul gören yaklaşım bu tip sistemlerin bütün ihtiyaçlarının önceden belirlenerek geliştirilmesinin pratik olmadığı yönündedir.
- ✧ Kullanıcı ara yüzleri web tarayıcıların yetenekleri ile sınırlıdır.
  - AJAX gibi teknolojiler zengin arayüzler oluşturmak için kullanılabilir ancak kullanımları halen zordur. Yerelde çalışan scriptler kullanarak web formları oluşturmak daha yaygındır.

# Web tabanlı yazılım mühendisliği

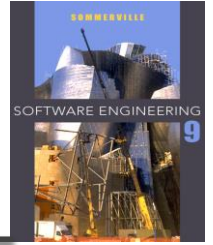
---



- ✧ Web tabanlı sistemler karmaşık dağıtık sistemlerdir. Fakat, daha önceden bahsettiğimiz yazılım mühendisliğinin temel prensipleri diğer sistemlere olduğu gibi bu tip sistemlere de uygulanabilir.

# Önemli noktalar

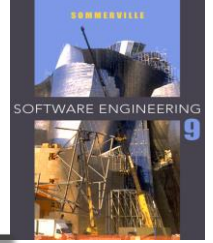
---



- ✧ Yazılım Mühendisliği, yazılım üretiminin bütün yönleri ile ilgilenen bir mühendislik alanıdır.
- ✧ Temel yazılım ürün özellikleri sürdürülebilirlik, güvenilirlik ve güvenlik, etkinlik ve kabul edilebilirliktir.
- ✧ Tanımlama, geliştirme, doğrulama ve değiştirme (evrim) bütün yazılım süreçlerinin parçalarıdır.
- ✧ Yazılım mühendisliğinin temel kavramları bütün sistem geliştirme tiplerine uygulanabilir.

# Önemli noktalar

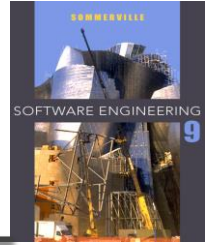
---



- ✧ Çok sayıda değişik sistem tipi vardır ve her biri doğru yazılım mühendisliği araçları ve tekniklerine ihtiyaç duyar.

# Yazılım mühendisliği etiği

---

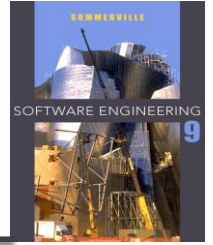


- ✧ Yazılım mühendisliği, basitçe teknik becerilerin kullanılmasından daha geniş sorumlulukları gerektirir.
- ✧ Yazılım mühendisleri eğer işlerine saygı duyuyorsa dürüst ve ahlaklı davranmak zorundadırlar.
- ✧ Etik davranmak yalnızca kanunlara uymak değildir. Aynı zamanda ahlak kurallarına uymayı gerektirir.



# Profesyonel sorumluluk konuları

---



## ✧ Gizlilik (mahremiyet)

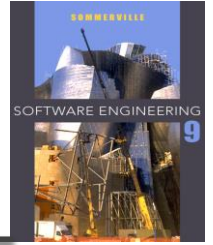
- Mühendisler, bir etik sözleşme imzalanmamış olsa bile iş verenlerinin ve müşterilerin gizliliklerine (özellerine) saygı duymalıdır.

## ✧ Yeterlilik (kabiliyet)

- Mühendisler kabiliyet seviyelerini yanlış sunmamalıdır. Kabiliyetlerini aşan işleri kabul etmemelidir.

# Profesyonel sorumluluk konuları

---



## ✧ Fikri hak mülkiyeti

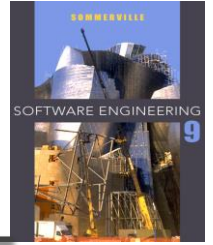
- Mühendisler, patent gibi fikri mülkiyet haklarının kanunlar ile korunduğundan haberdar olmalıdır. İşverenin ve müşterilerin fikri mülkiyet haklarının korunduğundan emin olmalıdır.

## ✧ Bilgisayarların kötüye kullanımı

- Yazılım mühendisleri bilgilerini diğer kişilerin bilgisayarlarını kötü amaçla veya amaç dışı kullanmak için kullanmamalıdır. Kötüye kullanım, işverenin bilgisayarında oyun oynamak gibi önemsiz bir konu da olabilir, bilgisayara virüs bulaştırmak gibi ciddi bir konu da olabilir.

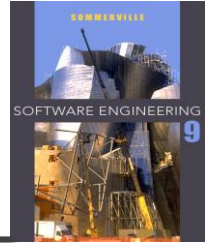
# ACM/IEEE Etik Yönetmeliği

---



- ✧ A.B.D.'deki bazı organizasyonlar bir etik yönetmelik oluşturmak için bir araya gelmiştir.
- ✧ Yönetmelik 8 temel prensip barındırır. Bu prensipler yazılım mühendisleri ile ilgili olduğu kadar programcılar, eğitimciler, yöneticiler, karar vericiler, stajyerler ve öğrencilerle ilgilidir.

# ACM/IEEE Etik Yönetmeliği



## Software Engineering Code of Ethics and Professional Practice

ACM/IEEE-CS Joint Task Force on Software Engineering Ethics and Professional Practices

### PREAMBLE

The short version of the code summarizes aspirations at a high level of the abstraction; the clauses that are included in the full version give examples and details of how these aspirations change the way we act as software engineering professionals. Without the aspirations, the details can become legalistic and tedious; without the details, the aspirations can become high sounding but empty; together, the aspirations and the details form a cohesive code.

Software engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession. In accordance with their commitment to the health, safety and welfare of the public, software engineers shall adhere to the following Eight Principles:

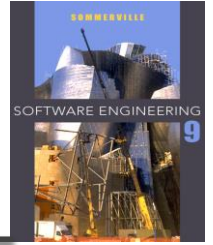
# Etik Kurallar



1. **PUBLIC (KAMU)** – Yazılım mühendisleri kamu menfaatine uygun davranmalıdır.
2. **MÜŞTERİ VE İŞ VEREN** – Yazılım mühendisleri kamu menfaatine uygun olacak şekilde müşteri ve iş veren menfaatine uygun davranmalıdır.
3. **ÜRÜN-** Yazılım mühendisleri ürünlerinin ve ilgili değişikliklerinin mümkün olan en üst düzey profesyonel standartları karşıladığını garanti etmelidirler.
4. **JUDGMENT (Profesyonelce karar verme)** – Yazılım mühendisleri verecekleri kararlarda dürüstlük ve bağımsızlıklarını korumalıdır.
5. **YÖNETİM** – Yazılım mühendisliği yöneticileri ve liderleri yazılım geliştirme ve sürdürme için bir etik yaklaşım benimsemelidirler.
6. **PROFESSION (MESLEK)** – Yazılım mühendisleri mesleklerinin saygınlığını kamu yararı doğrultusunda arttırmalıdır.
7. **İŞ ARKADAŞLARI** – Yazılım mühendisleri iş arkadaşlarına karşı adaletli ve destekleyici olmalıdırlar.
8. **SELF (ŞAHSI)** – Yazılım mühendisleri kendi mesleklerine bir saygı olarak hayat boyu öğrenmeyi benimsemeli ve mesleklerini uygularken bir etik yaklaşım belirlemelidirler.

# Etik ikilemler

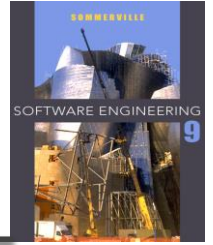
---



- ✧ Üst düzey yöneticilerin politikaları ile prensipte anlaşmazlık.
- ✧ İş vereniniz etik kurallara aykırı davranarak hayati tehlike arz eden bir sistemi, testleri tamamlamadan kullanıcıya teslim edebilir.
- ✧ Askeri silah veya nükleer sistemlerin geliştirilmesine katılmak.

# Örnek Yazılım Projeleri

---



## ✧ İnsülin pompası kontrol sistemi

- Diyabet hastaları tarafından kan şekeri seviyesini kontrol etmek için kullanılan, gömülü sistem insülin pompası

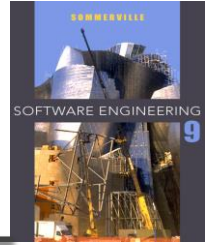
## ✧ Akıl sağlığı hastaları için hasta yönetim sistemi

- Akıl sağlığı problemleri yaşayan insanların kayıtlarını tutan bir sistem

## ✧ Sahradaki bir hava tahmin merkezi

- Uzak alanlardaki hava durumu koşulları ile ilgili bilgileri toplayan bir sistem

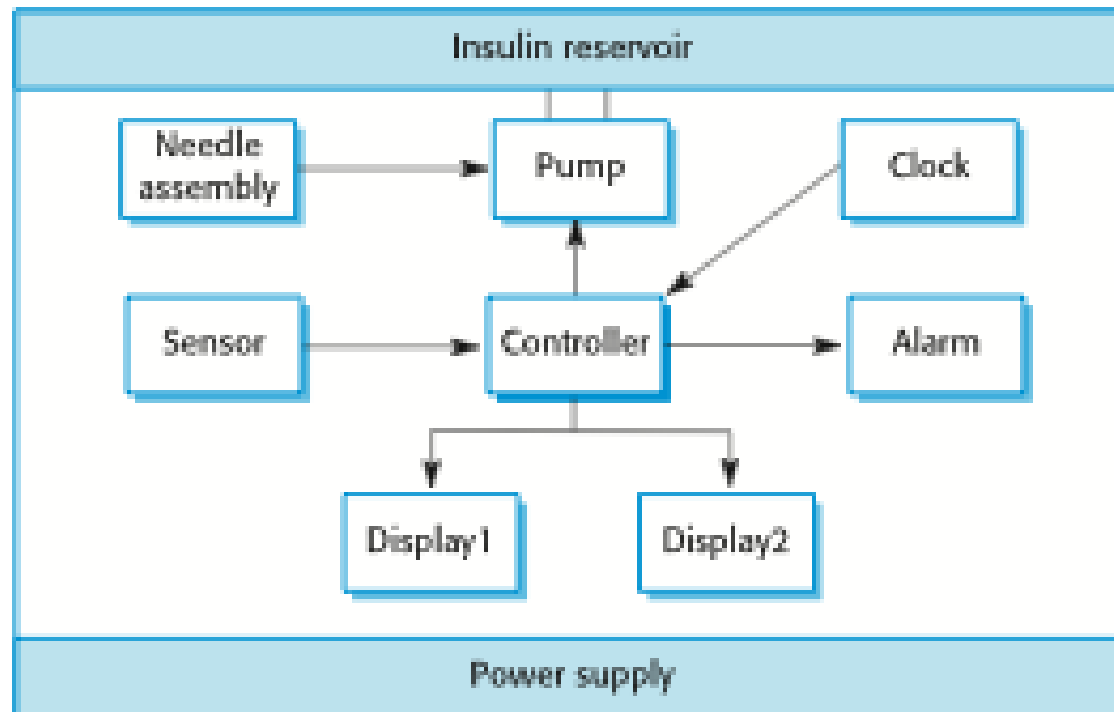
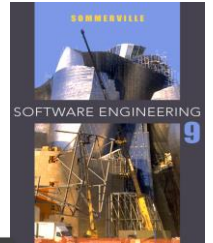
# İnsülin pompası kontrol sistemi



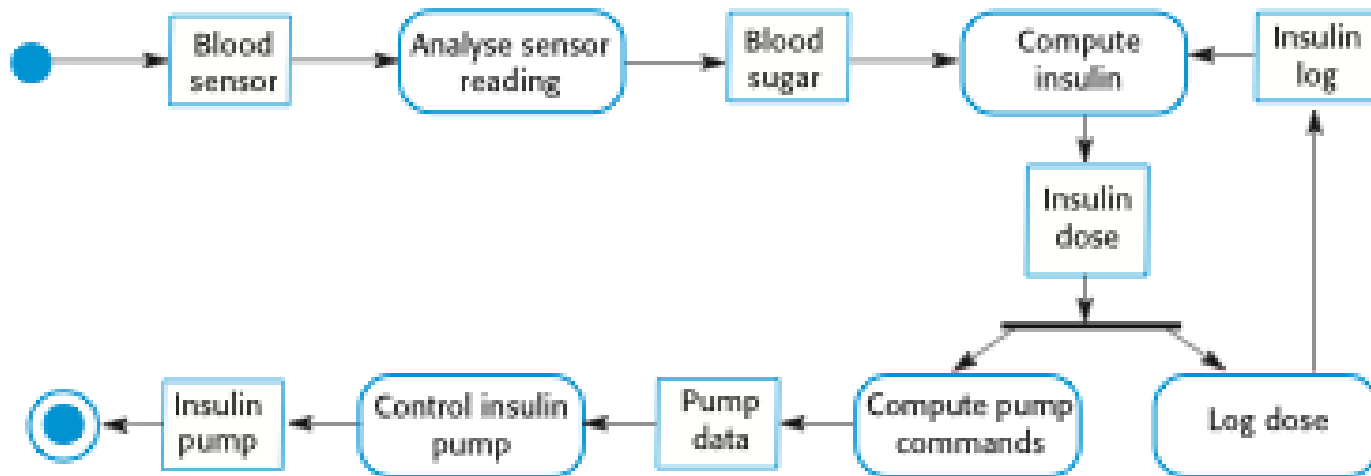
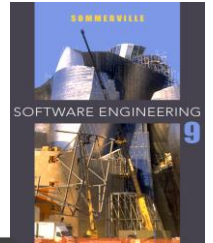
- ✧ Bir kan şekeri sensöründen verileri alır ve enjekte edilmesi gereken insülin miktarını hesaplar.
- ✧ Hesaplama, kan şekeri seviyesindeki değişikliklere göre yapılır
- ✧ Mikro düzeydeki bir pompaya sinyal göndererek gereken dozdaki insülinin enjekte edilmesini sağlar
- ✧ Hayati tehlike arz edecek kadar düşük kan şekeri seviyeleri beyin hasarına, komaya ve ölüme neden olabilir. Yüksek kan şekeri seviyeleri uzun vadede böbrek ve göz hasarlarına neden olabilir.



# İnsülin pompasının donanım mimarisi

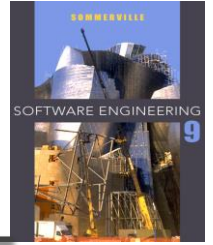


# İnsülin pompasının çalışması



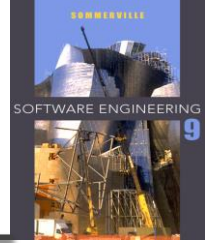
# Temel yüksek seviyeli gereksinimleri

---



- ✧ İnsülin enjekte edilmesi gerektiği anda sistem bunu yapabiliyor olmalı
- ✧ Sistem, mevcut kan şekeri seviyesini dengelemek için gerekli olan doğru miktar insülini enjekte edebilmeli
- ✧ Sistem bu ihtiyaçları her zaman karşılayabilecek şekilde tasarlanmalı ve gerçekleştirilmeli

# Akıl sağlığı hastaları için hasta yönetim sistemi



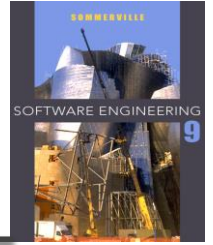
- ✧ Akıl sağlığı ile ilgili problem yaşayan hastaların bilgilerini tutan ve hasta bakım sürecini destekleyen bir hasta bilgi sistemi.
- ✧ Çoğu akıl sağlığı problemi yaşayan hastaların sürekli olarak bir hastanede tutulmalarına gerek yoktur. Bunun yerine, hastayı iyi tanıyan bir doktorun bulunduğu bir kliniğe periyodik olarak gitmeleri yeterlidir.
- ✧ Hastaların katılımlarını kolaylaştırmak için bu klinikler yalnızca hastanelerde bulunmazlar. Aynı zamanda bölgesel sağlık kuruluşları ile topluluk (dernek) merkezlerinde bulunurlar.



- ✧ MHC-PMS (Mental Health Care-Patient Management System) sistemi, kliniklerde kullanılmak üzere tasarlanmış bir bilgi sistemidir.
- ✧ Sistem, hastaların bilgilerini tutmak için merkezi bir veritabanı kullanır ve bu veritabanına bir PC aracılığı ile ulaşılabilir. Bundan dolayı, güvenli bir ağ bağlantısı olmayan yerlerden de ulaşılabilir.
- ✧ Lokal sistemler güvenli bir ağ bağlantısına sahip oldukları zaman veritabanına bağlanarak hasta bilgilerini kullanabilirler. Ağ bağlantıları olmadığı zamanlarda, indirdikleri hasta kayıtları ile çalışabilirler.

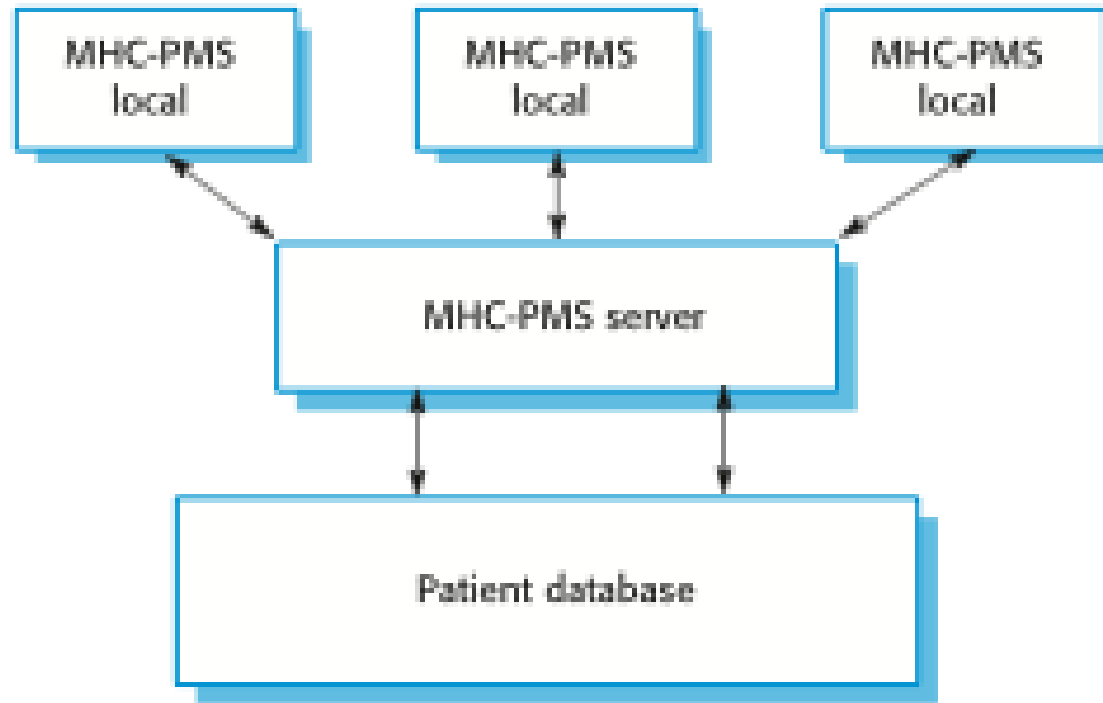
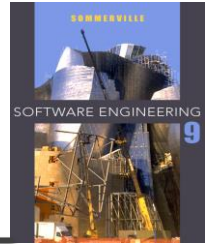
# MHC-PMS'in amaçları

---

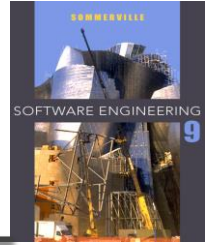


- ✧ Bilgi yönetimi, sağlık hizmetleri yöneticilerine, yerel ve ulusal hedeflere göre performans analizi yapabilmeyi sağlar.
- ✧ Hastalar ile ilgili güncel bilgileri sağlık görevlilerine ulaştırarak hastanın tedavi sürecini desteklemek

# MHC-PMS'nin organizasyon şeması



# MHC-PMS'nin temel özellikleri



## ✧ Bireysel hasta yönetimi

- Klinik uzmanları hastalar için kayıtlar oluşturabilir, bilgileri değiştirebilir, hastanın geçmiş kayıtlarını görebilir vb. Sistem, verileri özet olarak uzmana sunabilir. Bu sayede uzman, hastanın temel problemlerini ve sorunlarını hızlıca görebilir.

## ✧ Hasta gözlemleme

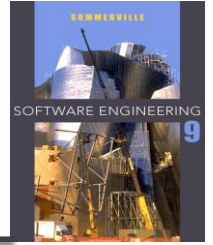
- Sistem, hastaların sorunları ile ilgili kayıtlarını inceleyerek eğer mümkünse muhtemel problemleri tespit eder ve uyarı verir.

## ✧ Yönetimsel raporlama

- Sistem aylık raporlar üreterek her bir klinikte tedavi gören hasta sayısı, hasta bakım sistemine girmiş ve sistemden çıkmış hasta sayısı, farklı kliniklere yönlendirilen hasta sayısı, yazılan ilaç sayısı ve maliyeti gibi konularda bilgi sağlar.



# MHC-PMS'de dikkat edilmesi gereken noktalar



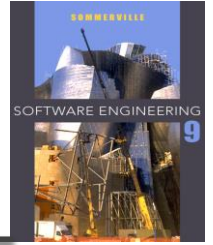
## ✧ Gizlilik

- Hastaların bütün bilgileri gizlidir. Hastanın kendisi ve yetkili kişiler haricinde hiç kimsenin bu bilgilere ulaşmasına izin verilmez.

## ✧ Güvenlik

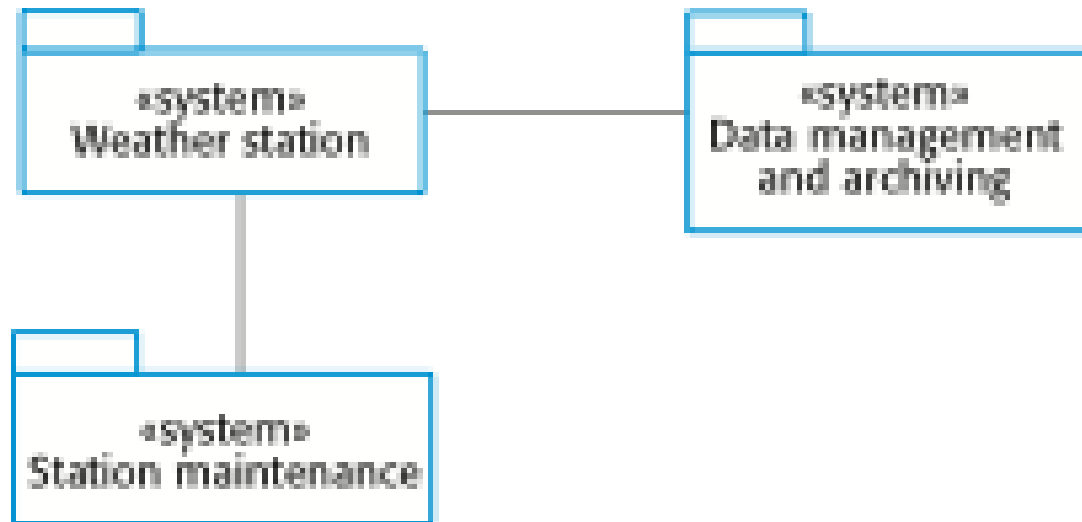
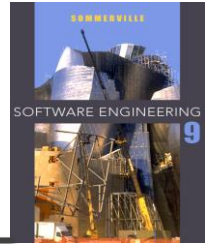
- Bazı akıl hastalıklarında hasta intihara meyillidir ya da diğer insanlara zarar verme potansiyeli vardır. Mümkün olduğu zaman, sistem bu tip durumlar hakkında yetkili kişileri uyarmalıdır.
- Sistem, her ihtiyaç duyulduğunda erişilebilir olmalıdır. Aksi taktirde güvenlikle ilgili sıkıntılar oluşabilir, hastaya doğru ilacın verilmesi mümkün olmayabilir.

# Sahradaki bir hava tahmin merkezi

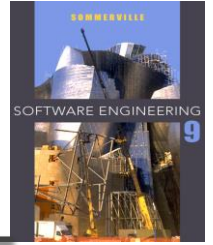


- ✧ Büyük miktarda yerleşim dışı arazisi olan bir ülkenin yöneticileri uzak yerlerde birkaç yüz tane hava tahmin istasyonu kurmaya karar veriyor.
- ✧ Hava tahmin istasyonları cihazlardan bazı bilgileri toplar. Bu bilgiler sıcaklık, basınç, gün doğumu, yağış miktarı, rüzgar hızı ve yönü bilgileridir.
  - Hava tahmin istasyonu bu bilgileri 24 saat boyunca ölçerler. Bütün ölçüm cihazları bir yazılım tarafından yönetilir. Bu yazılım, okunan ölçümleri alır ve toplanan verileri yönetir.

# Hava tahmin istasyonunun şeması



# Hava tahmin istasyonu bilgi sistemi



- ✧ The weather station system  
(Hava durumu istasyon sistemi)
  - Bu sistem hava durumu ile ilgili bilgileri toplar; bazı başlangıç veri işleme süreçlerini gerçekleştirir ve verileri veri yönetim sistemine iletir.
- ✧ The data management and archiving system (Veri yönetimi ve arşivleme sistemi)
  - Bu sistem, arazideki bütün istasyonlardan verileri toplar, bazı veri işleme süreçlerini gerçekleştirir, veriyi analiz eder ve arşivler.
- ✧ The station maintenance system (İstasyon bakım sistemi)
  - Bu sistem uydu üzerinden arazideki bütün istasyonlar ile haberleşerek istasyonların sağlıklı çalışıp çalışmadıklarını gözlemler ve problemlerle ilgili raporlar üretir.

# Ek Yazılım İşlevselliği



- ✧ Ölçüm cihazlarının, güç ve haberleşme sistemlerinin gözlemlenmesi ve hataların yönetim sistemine raporlanması.
- ✧ Sistem gücünün yönetilmesi, hava durumu uygun olduğunda bataryaların şarj edilmesi ve aynı zamanda tehlikeli hava koşulları (şiddetli rüzgar) oluştuğunda jeneratörlerin kapatılması.
- ✧ Yazılım güncelleştirmelerini ve konfigürasyonlarını dinamik olarak destekler ve sistemde bir hata meydana geldiğinde yedek ölçüm cihazlarını devreye alır.