

## Ders 1- Giriş

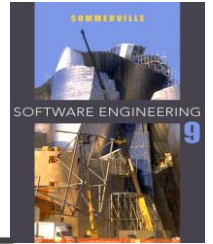
### Part 1



Source : <https://ifs.host.cs.st-andrews.ac.uk/Books/SE9/Presentations/index.html>

# Topics covered

---



- ✧ Professional software development
  - What is meant by software engineering.
- ✧ Software engineering ethics
  - A brief introduction to ethical issues that affect software engineering.
- ✧ Case studies
  - An introduction to three examples that are used in later chapters in the book.

# Software engineering

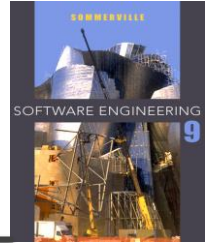
---



- ✧ The economies of ALL developed nations are dependent on software.
- ✧ More and more systems are software controlled
- ✧ Software engineering is concerned with theories, methods and tools for professional software development.
- ✧ Expenditure on software represents a significant fraction of Gross National Product (GNP) in all developed countries.

# Software costs

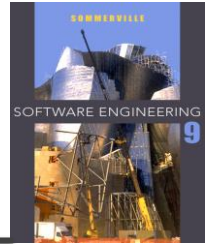
---



- ✧ Software costs often dominate computer system costs. The costs of software on a PC are often greater than the hardware cost.
- ✧ Software costs more to maintain than it does to develop. For systems with a long life, maintenance costs may be several times development costs.
- ✧ Software engineering is concerned with cost-effective software development.

# Software products

---



## ✧ Generic products

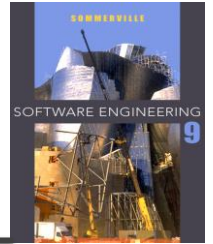
- Stand-alone systems that are marketed and sold to any customer who wishes to buy them.
- Examples – PC software such as graphics programs, project management tools; computer-aided design (CAD) software; software for specific markets such as appointments systems for dentists.

## ✧ Customized products

- Software that is commissioned by a specific customer to meet their own needs.
- Examples – embedded control systems, air traffic control software, traffic monitoring systems.

# Product specification

---



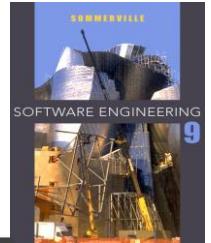
## ✧ Generic products

- The specification of what the software should do is owned by the software developer and decisions on software change are made by the developer.

## ✧ Customized products

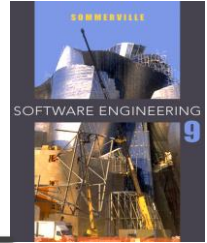
- The specification of what the software should do is owned by the customer for the software and they make decisions on software changes that are required.

# Frequently asked questions about software engineering



Question	Answer
What is software?	Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.
What are the attributes of good software?	Good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable.
What is software engineering?	Software engineering is an engineering discipline that is concerned with all aspects of software production.
What are the fundamental software engineering activities?	Software specification, software development, software validation and software evolution.
What is the difference between software engineering and computer science?	Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.
What is the difference between software engineering and system engineering?	System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this more general process.

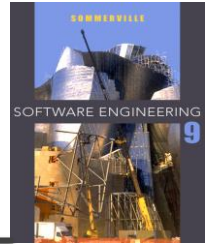
# Frequently asked questions about software engineering



Question	Answer
What are the key challenges facing software engineering?	Coping with increasing diversity, demands for reduced delivery times and developing trustworthy software.
What are the costs of software engineering?	Roughly 60% of software costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs.
What are the best software engineering techniques and methods?	While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of system. For example, games should always be developed using a series of prototypes whereas safety critical control systems require a complete and analyzable specification to be developed. You can't, therefore, say that one method is better than another.
What differences has the web made to software engineering?	The web has led to the availability of software services and the possibility of developing highly distributed service-based systems. Web-based systems development has led to important advances in programming languages and software reuse.



# Essential attributes of good software



Product characteristic	Description
Maintainability	Software should be written in such a way so that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment.
Dependability and security	Software dependability includes a range of characteristics including reliability, security and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system.
Efficiency	Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, memory utilisation, etc.
Acceptability	Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable and compatible with other systems that they use.

# Software engineering

---



- ✧ Software engineering is an engineering discipline that is concerned with all aspects of software production from the early stages of system specification through to maintaining the system after it has gone into use.
- ✧ Engineering discipline
  - Using appropriate theories and methods to solve problems bearing in mind organizational and financial constraints.
- ✧ All aspects of software production
  - Not just technical process of development. Also project management and the development of tools, methods etc. to support software production.

# Importance of software engineering

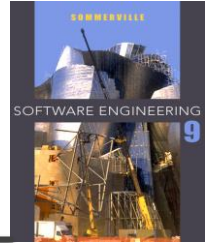
---



- ✧ More and more, individuals and society rely on advanced software systems. We need to be able to produce reliable and trustworthy systems economically and quickly.
- ✧ It is usually cheaper, in the long run, to use software engineering methods and techniques for software systems rather than just write the programs as if it was a personal programming project. For most types of system, the majority of costs are the costs of changing the software after it has gone into use.

# Software process activities

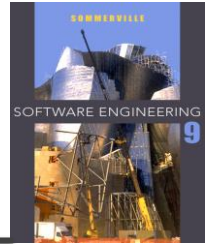
---



- ✧ Software specification, where customers and engineers define the software that is to be produced and the constraints on its operation.
- ✧ Software development, where the software is designed and programmed.
- ✧ Software validation, where the software is checked to ensure that it is what the customer requires.
- ✧ Software evolution, where the software is modified to reflect changing customer and market requirements.

# General issues that affect most software

---



## ✧ Heterogeneity

- Increasingly, systems are required to operate as distributed systems across networks that include different types of computer and mobile devices.

## ✧ Business and social change

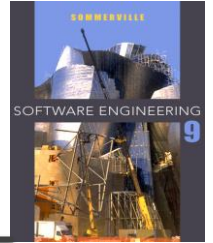
- Business and society are changing incredibly quickly as emerging economies develop and new technologies become available. They need to be able to change their existing software and to rapidly develop new software.

## ✧ Security and trust

- As software is intertwined with all aspects of our lives, it is essential that we can trust that software.

# Software engineering diversity

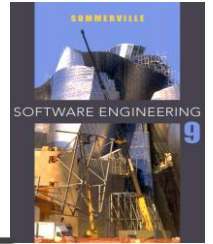
---



- ✧ There are many different types of software system and there is no universal set of software techniques that is applicable to all of these.
- ✧ The software engineering methods and tools used depend on the type of application being developed, the requirements of the customer and the background of the development team.

# Application types

---



## ✧ Stand-alone applications

- These are application systems that run on a local computer, such as a PC. They include all necessary functionality and do not need to be connected to a network.

## ✧ Interactive transaction-based applications

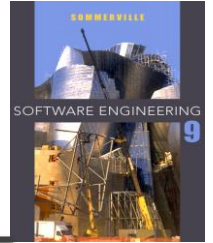
- Applications that execute on a remote computer and are accessed by users from their own PCs or terminals. These include web applications such as e-commerce applications.

## ✧ Embedded control systems

- These are software control systems that control and manage hardware devices. Numerically, there are probably more embedded systems than any other type of system.

# Application types

---



## ✧ Batch processing systems

- These are business systems that are designed to process data in large batches. They process large numbers of individual inputs to create corresponding outputs.

## ✧ Entertainment systems

- These are systems that are primarily for personal use and which are intended to entertain the user.

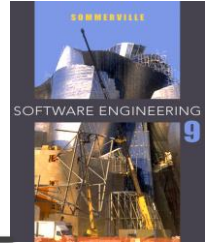
## ✧ Systems for modelling and simulation

- These are systems that are developed by scientists and engineers to model physical processes or situations, which include many, separate, interacting objects.



# Application types

---



## ✧ Data collection systems

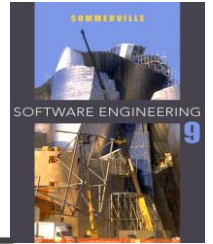
- These are systems that collect data from their environment using a set of sensors and send that data to other systems for processing.

## ✧ Systems of systems

- These are systems that are composed of a number of other software systems.

# Software engineering fundamentals

---



- ✧ Some fundamental principles apply to all types of software system, irrespective of the development techniques used:
  - Systems should be developed using a managed and understood development process. Of course, different processes are used for different types of software.
  - Dependability and performance are important for all types of system.
  - Understanding and managing the software specification and requirements (what the software should do) are important.
  - Where appropriate, you should reuse software that has already been developed rather than write new software.

# Software engineering and the web

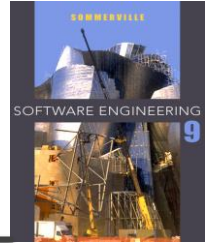
---



- ✧ The Web is now a platform for running application and organizations are increasingly developing web-based systems rather than local systems.
- ✧ Web services (discussed in Lecture 19) allow application functionality to be accessed over the web.
- ✧ Cloud computing is an approach to the provision of computer services where applications run remotely on the 'cloud'.
  - Users do not buy software but pay according to use.

# Web software engineering

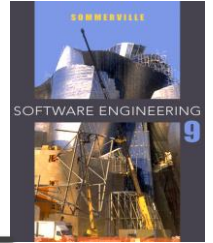
---



- ✧ Software reuse is the dominant approach for constructing web-based systems.
  - When building these systems, you think about how you can assemble them from pre-existing software components and systems.
- ✧ Web-based systems should be developed and delivered incrementally.
  - It is now generally recognized that it is impractical to specify all the requirements for such systems in advance.
- ✧ User interfaces are constrained by the capabilities of web browsers.
  - Technologies such as AJAX allow rich interfaces to be created within a web browser but are still difficult to use. Web forms with local scripting are more commonly used.

# Web-based software engineering

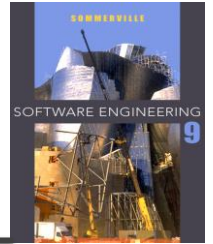
---



- ✧ Web-based systems are complex distributed systems but the fundamental principles of software engineering discussed previously are as applicable to them as they are to any other types of system.
- ✧ The fundamental ideas of software engineering, discussed in the previous section, apply to web-based software in the same way that they apply to other types of software system.

# Key points

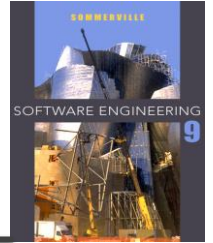
---



- ✧ Software engineering is an engineering discipline that is concerned with all aspects of software production.
- ✧ Essential software product attributes are maintainability, dependability and security, efficiency and acceptability.
- ✧ The high-level activities of specification, development, validation and evolution are part of all software processes.
- ✧ The fundamental notions of software engineering are universally applicable to all types of system development.

# Key points

---



- ✧ There are many different types of system and each requires appropriate software engineering tools and techniques for their development.
- ✧ The fundamental ideas of software engineering are applicable to all types of software system.

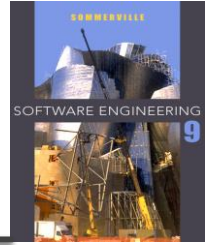
## Ders 1 - Giriş

## Bölüm 2



# Yazılım mühendisliği etiği

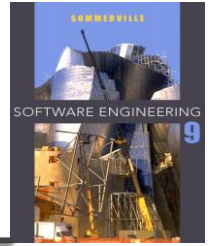
---



- ✧ Yazılım mühendisliği, sadece teknik becerilerin uygulanmasından daha geniş sorumluluklar içerir.
- ✧ Profesyoneller olarak saygı görmeleri için yazılım mühendisleri dürüst ve etik açıdan sorumlu bir şekilde davranmalıdır.
- ✧ Etik davranış, sadece yasayı korumaktan daha fazlasıdır, ancak ahlaki olarak doğru olan bir dizi ilkeyi takip etmeyi içerir.

# Mesleki sorumluluk konuları

---



## ✧ Gizlilik

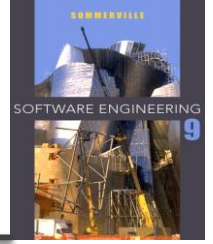
- Mühendisler, resmi bir gizlilik sözleşmesi imzalanmış olsun ya da olmasın normalde işverenlerinin veya müşterilerinin gizliliğine saygı göstermelidir.

## ✧ Yetkinlik

- Mühendisler, yetkinlik seviyelerini yanlış beyan etmemelidir. Yetkinliklerinin dışında kalan işleri bilerek kabul etmemelidirler.

# Mesleki sorumluluk konuları

---



## ✧ Fikri Mülkiyet Hakları

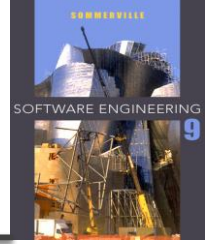
- Mühendisler, patent, telif hakkı vb. gibi fikri mülkiyetin kullanımını düzenleyen yerel yasaların farkında olmalıdırlar. İşverenlerin ve müşterilerin fikri mülkiyetinin korunduğundan emin olmak için dikkatli olmalıdırlar.

## ✧ Bilgisayarın kötüye kullanımı

- Yazılım mühendisleri, teknik becerilerini başkalarının bilgisayarlarını kötüye kullanmak için kullanmamalıdır. Bilgisayarın kötüye kullanımı, görece önemsizden (örneğin bir işverenin makinesinde oyun oynama) son derece ciddiye (virüslerin yayılması) kadar çeşitlilik gösterir.

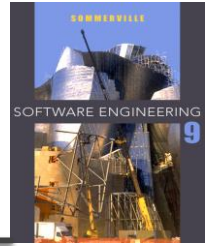
# ACM (Bilgisayar Makineleri Derneği) /IEEE (Elektrik ve Elektronik Mühendisleri Enstitüsü) Etik Kurallar

---



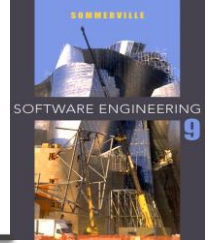
- ✧ ABD'deki profesyonel topluluklar, bir etik uygulama kodu üretmek için işbirliği yaptılar.
- ✧ Bu kuruluşların üyeleri, katıldıklarında uygulama kurallarına kaydolurlar.
- ✧ Kurallar, uygulayıcılar, eğitimciler, yöneticiler, süpervizörler ve politika yapıcıların yanı sıra mesleğin stajyerleri ve öğrencileri dahil olmak üzere profesyonel yazılım mühendislerinin davranışları ve aldıkları kararlarla ilgili sekiz İlke içerir.

# Etik kuralların gerekçesi



- *Bilgisayarlar, ticaret, endüstri, hükümet, tıp, eğitim, eğlence ve genel olarak toplumda merkezi ve büyüyen bir role sahiptir. Yazılım mühendisleri, yazılım sistemlerinin analizine, spesifikasyonuna, tasarımına, geliştirilmesine, sertifikasyonuna, bakımına ve test edilmesine doğrudan katılımla veya öğreterek katkıda bulunan kişilerdir.*
- *Yazılım sistemleri geliştirmedeki rollerinden dolayı, yazılım mühendisleri, başkalarının iyilik yapmasını veya zarar vermesini sağlamak veya başkalarını iyilik yapmaları veya zarar vermeleri için etkilemek için iyilik yapma veya zarar verme konusunda önemli fırsatlara sahiptir. Yazılım mühendisleri, çabalarının iyilik için kullanılmasını mümkün olduğunca sağlamak için, kendilerini yazılım mühendisliğini faydalı ve saygın bir meslek haline getirmeye adanmıştır.*

# ACM (Bilgisayar Makineleri Derneği) /IEEE (Elektrik ve Elektronik Mühendisleri Enstitüsü) Etik Kuralları



## Yazılım Mühendisliği Etik Kuralları ve Mesleki Uygulama

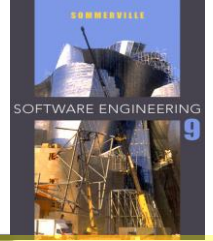
ACM/IEEE-CS Yazılım Mühendisliği Etik ve Mesleki Uygulamalar Ortak Görev Gücü

### GİRİŞ

Kodun kısa versiyonu, yüksek bir soyutlama düzeyindeki istekleri özetlemektedir; tam sürüme dahil edilen maddeler, bu isteklerin yazılım mühendisliği uzmanları olarak davranış şeklimizi nasıl değiştirdiğine dair örnekler ve ayrıntılar verir. Özlemler olmadan ayrıntılar yasal ve sıkıcı hale gelebilir; ayrıntılar olmadan, özlemler kulağa yüksek gelebilir ama boş olabilir; birlikte, özlemler ve ayrıntılar uyumlu bir kod oluşturur.

Yazılım mühendisleri, yazılımın analizini, özelliklerini, tasarımını, geliştirilmesini, test edilmesini ve bakımını yararlı ve saygın bir meslek haline getirmeye kendilerini adanmalıdır. Halkın sağlığı, güvenliği ve refahına yönelik taahhütlerine uygun olarak, yazılım mühendisleri aşağıdaki Sekiz İlkeye bağlı kalacaktır:

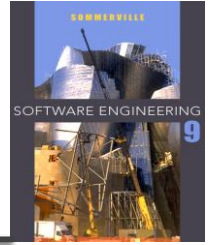
# Etik ilkeler



1. KAMU - Yazılım mühendisleri, kamu çıkarı için tutarlı bir şekilde hareket etmelidir.
2. MÜŞTERİ VE İŞVEREN - Yazılım mühendisleri, müşterileri ve işverenleri için kamu menfaatine uygun şekilde en yüksek menfaat olacak şekilde hareket edeceklerdir.
3. ÜRÜN - Yazılım mühendisleri, ürünlerinin ve ilgili değişikliklerinin mümkün olan en yüksek profesyonel standartları karşılamasını sağlayacaktır.
4. KARAR - Yazılım mühendisleri mesleki muhakemelerinde bütünlük ve bağımsızlığı koruyacaklardır.
5. YÖNETİM - Yazılım mühendisliği yöneticileri ve liderleri, yazılım geliştirme ve bakım yönetimine etik bir yaklaşıma abone olacak ve bunu teşvik edeceklerdir.
6. MESLEK - Yazılım mühendisleri, mesleğin bütünlüğünü ve itibarını kamu yararına uygun şekilde geliştireceklerdir.
7. MESLEKTAŞLAR - Yazılım mühendisleri meslektaşlarına karşı adil ve onları destekleyici olmalıdır.
8. KİŞİSEL - Yazılım mühendisleri, mesleklerinin uygulanmasına ilişkin yaşam boyu öğrenmeye katılacak ve mesleğin uygulanmasına etik bir yaklaşım geliştireceklerdir.

# Etik ikilemler

---

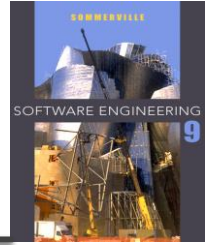


- ✧ Üst yönetimin politikaları ile prensipte anlaşmazlık.
- ✧ İşvereniniz etik olmayan bir şekilde davranır ve sistemin testini bitirmeden güvenlik açısından kritik bir sistemi yayınlar.
- ✧ Askeri silah sistemlerinin veya nükleer sistemlerin geliştirilmesine katılım.



# Örnek olaylar

---



## ✧ Kişisel bir insülin pompası

- Şeker hastaları tarafından kan şekeri kontrolünü sağlamak için kullanılan bir insülin pompasına gömülü bir sistem.

## ✧ Bir akıl sağlığı hasta yönetim sistemi

- Akıl sağlığı sorunları için bakım alan kişilerin kayıtlarını tutmak için kullanılan bir sistem.

## ✧ Bir doğa hava istasyonu

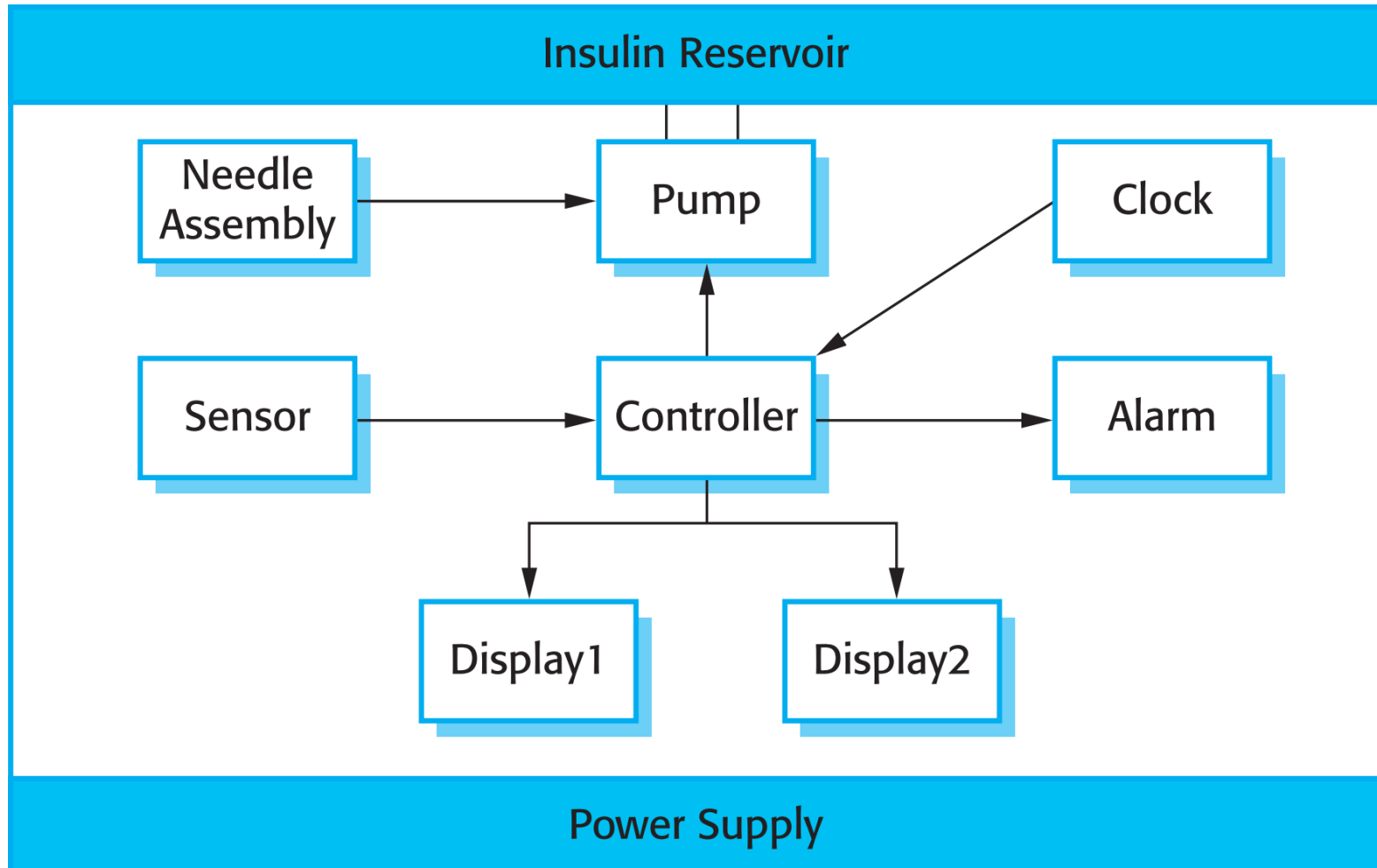
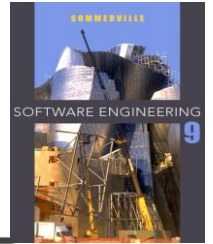
- Uzak bölgelerdeki hava koşulları hakkında veri toplayan bir veri toplama sistemi.

# İnsülin pompası kontrol sistemi

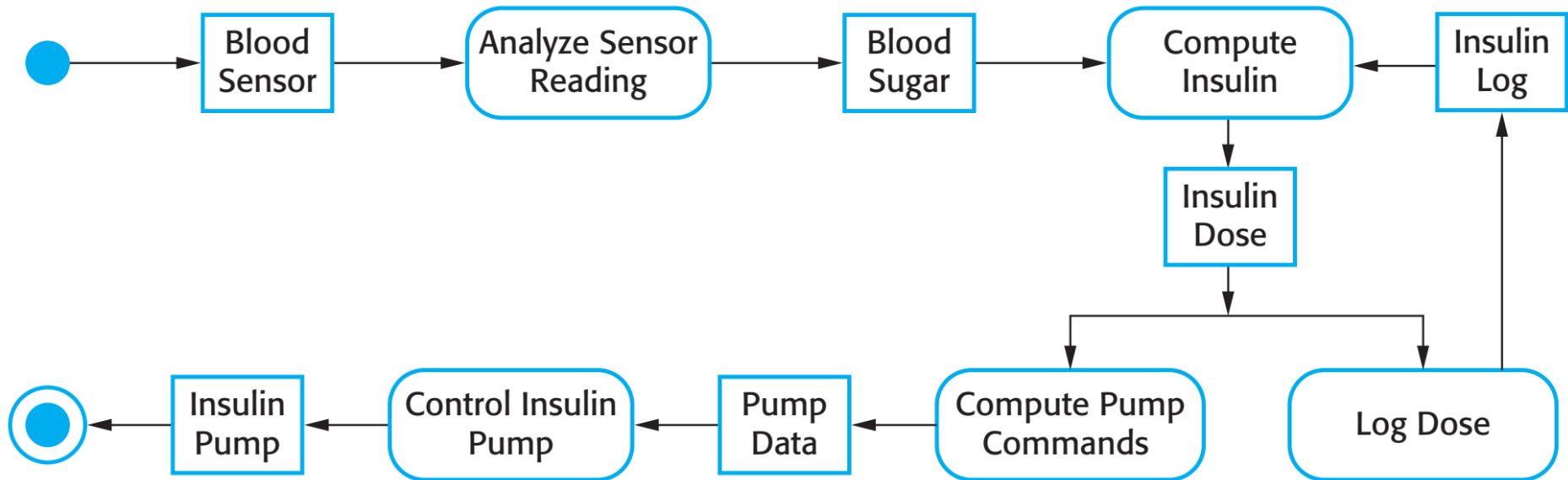
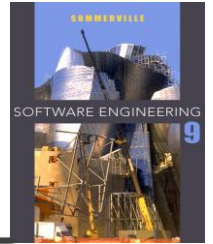


- ✧ Bir kan şekeri sensöründen veri toplar ve enjekte edilmesi gereken insülin miktarını hesaplar.
- ✧ Kan şekeri seviyelerinin değişim oranına göre hesaplama.
- ✧ Doğru insülin dozunu iletmek için bir mikro pompaya sinyal gönderir.
- ✧ Düşük kan şekeri beyin hasarına, komaya ve ölüme yol açabileceğinden güvenlik açısından kritik sistem; yüksek kan şekeri seviyelerinin göz ve böbrek hasarı gibi uzun vadeli sonuçları vardır.

# İnsülin pompası donanım mimarisi

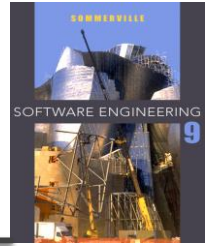


# İnsülin pompasının aktivite modeli



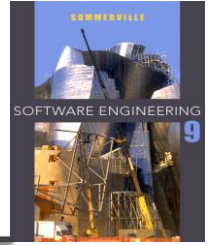
# Temel üst düzey gereksinimler

---

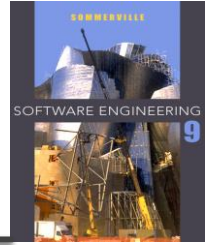


- ✧ Sistem, gerektiğinde insülin verebilecek durumda olacaktır.
- ✧ Sistem güvenilir bir şekilde çalışacak ve mevcut kan şekeri düzeyini dengelemek için doğru miktarda insülin gönderecektir.
- ✧ Bu nedenle sistem, sistemin her zaman bu gereksinimleri karşıladığından emin olmak için tasarlanmalı ve uygulanmalıdır.

# Akıl sağlığı bakımı için bir hasta bilgi sistemi



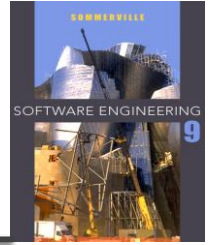
- ✧ Akıl sağlığı hizmetlerini desteklemek için bir hasta bilgi sistemi: akıl sağlığı sorunları olan hastalar ve gördükleri tedaviler hakkında bilgi tutan tıbbi bir bilgi sistemidir.
- ✧ Çoğu akıl sağlığı hastası, özel bir hastanede tedaviye ihtiyaç duymaz, ancak sorunları hakkında ayrıntılı bilgiye sahip bir doktorla görüşebilecekleri uzman kliniklere düzenli olarak gitmeleri gerekir.
- ✧ Hastaların katılmasını kolaylaştırmak için, bu klinikler sadece hastanelerde işletilmiyor. Yerel tıbbi muayenahanelerde veya toplum merkezlerinde de bu klinikler bulunabilir.



- ✧ AS-HYS (Akıl Sağlığı Hasta Yönetim Sistemi) kliniklerde kullanılmak üzere tasarlanmış bir bilgi sistemidir.
- ✧ Hasta bilgilerinin tutulduğu merkezi bir veri tabanını kullanır, ancak aynı zamanda bir PC üzerinde çalışacak şekilde tasarlanmıştır, böylece güvenli ağ bağlantısı olmayan cihazlardan erişilebilir ve kullanılabilir.
- ✧ Yerel sistemler güvenli ağ erişimine sahip olduklarında, merkezi veri tabanındaki hasta bilgilerini kullanırlar, ancak bağlantıları kesildiklerinde hasta kayıtlarının yerel kopyalarını indirebilir ve kullanabilirler.

# AS-HYS hedefleri

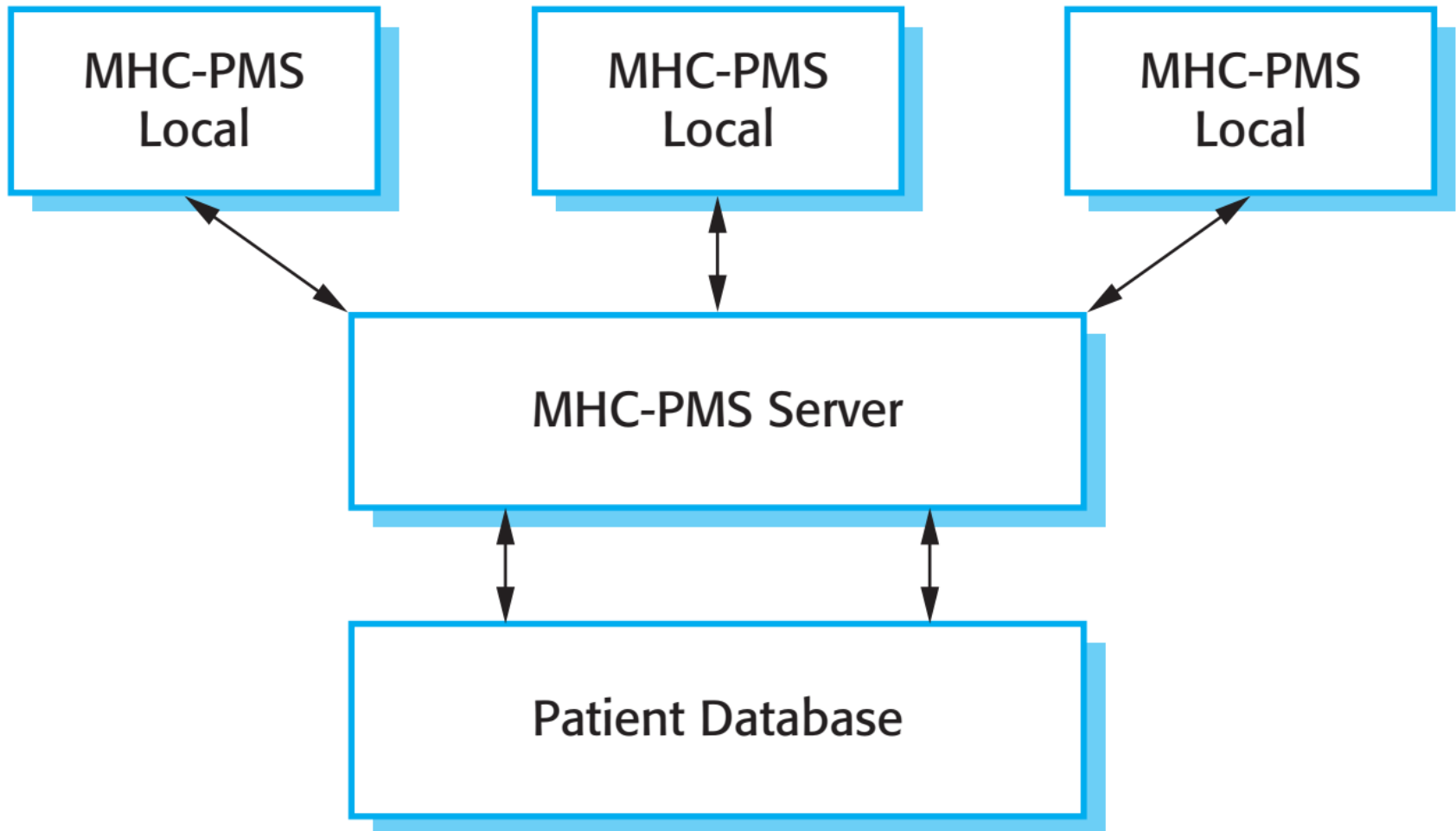
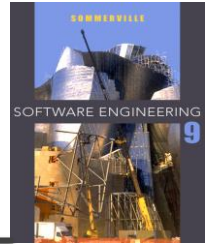
---



- ✧ Sağlık hizmeti yöneticilerinin, performans durumunu, yerel ve hükümet hedeflerine göre değerlendirmesine olanak tanıyan yönetim bilgilerini oluşturmak.
- ✧ Hastaların tedavisini desteklemek için sağlık personeline zamanında bilgi sağlamak.



# AS-HYS'nin organizasyonu



# AS-HYS'nin anahtar özellikleri



## ✧ Bireysel bakım yönetimi

- Klinisyenler hastalar için kayıt oluşturabilir, sistemdeki bilgileri düzenleyebilir, hasta geçmişini görüntüleyebilir, vb. Sistem veri özetlerini destekler, böylece doktorlar reçete edilen temel sorunlar ve tedaviler hakkında hızlı bir şekilde bilgi edinebilir.

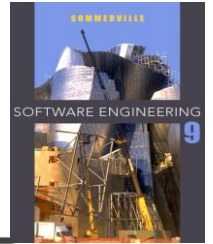
## ✧ Hasta izleme

- Sistem, tedaviye dahil olan hastaların kayıtlarını takip eder ve olası sorunlar tespit edildiğinde uyarı verir.

## ✧ İdari raporlama

- Sistem, her klinikte tedavi edilen hasta sayısını, bakım sistemine giren ve çıkan hasta sayısını, bölümlere ayrılan hasta sayısını, reçete edilen ilaçları ve bunların maliyetlerini vb. gösteren aylık yönetim raporları oluşturur.

# AS-HYS'nin endişeleri



## ✧ Gizlilik

- Hasta bilgilerinin gizli olması ve yetkili sağlık personeli ve hastanın kendisi dışında asla kimseye ifşa edilmemesi esastır.

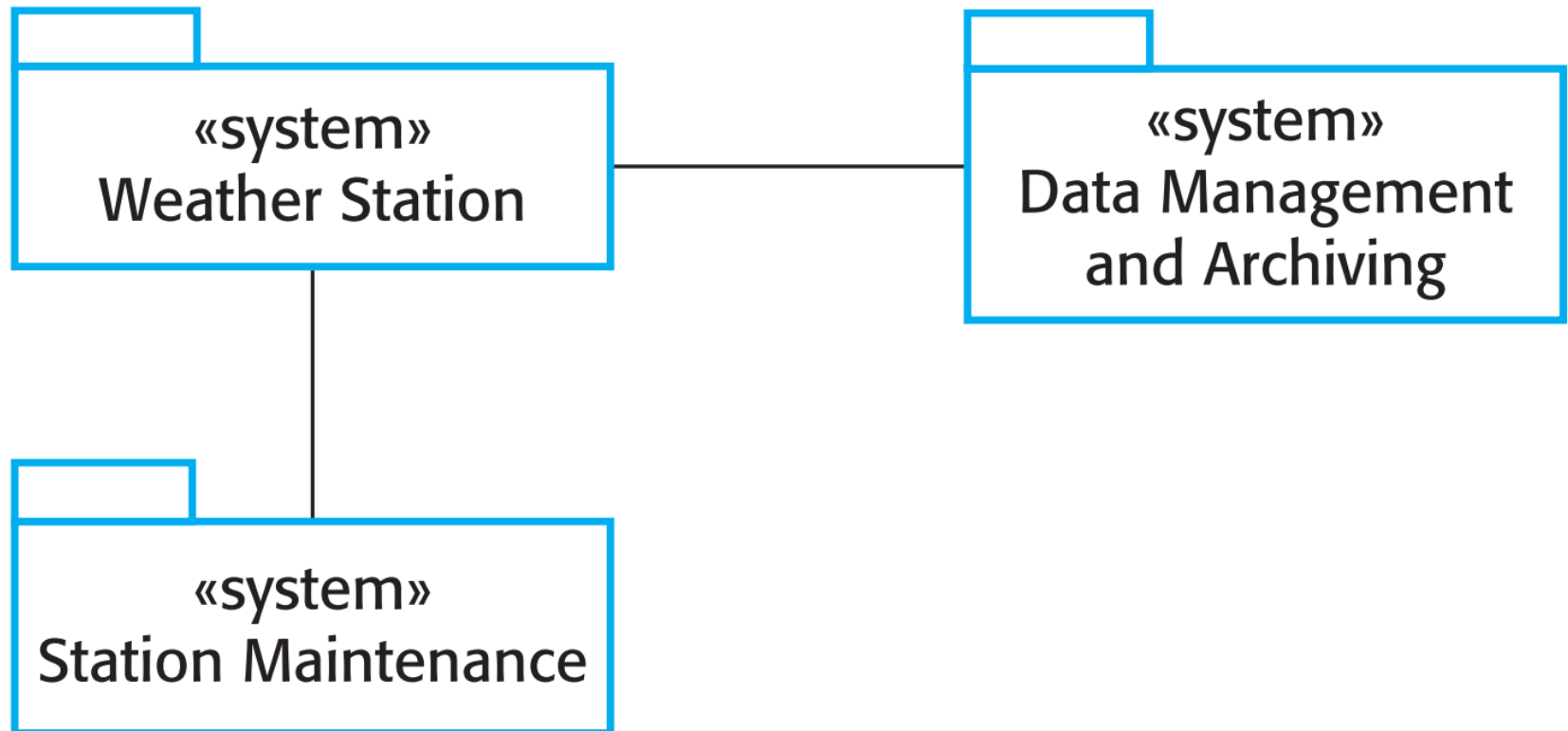
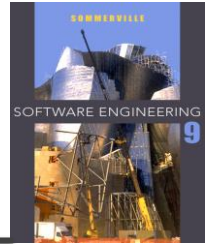
## ✧ Güvenlik

- Bazı akıl hastalıkları, hastaların intihar etmesine veya başkaları için tehlike oluşturmalarına neden olur. Mümkün olan her yerde, sistem tıbbi personeli intihara meyilli veya tehlikeli hastalar konusunda uyarmalıdır.
- Sistem ihtiyaç duyulduğunda hazır olmalıdır, aksi takdirde güvenlik tehlikeye girebilir ve hastalara doğru ilacı reçete etmek imkansız olabilir.

# Doğa hava istasyonu

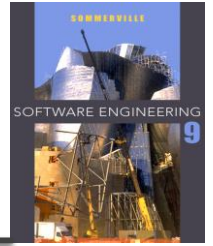
- ✧ Geniş vahşi alanlara sahip bir ülkenin hükümeti uzak bölgelere birkaç yüz meteoroloji istasyonu yerleştirmeye karar verir.
- ✧ Hava durumu istasyonları, sıcaklığı ve basıncı, güneş ışığını, yağmuru, rüzgar hızını ve rüzgar yönünü ölçen bir dizi cihazdan veri toplar.
  - Hava durumu istasyonu, rüzgar hızı ve yönü, yer ve hava sıcaklıkları, barometrik basınç ve 24 saatlik bir süre boyunca yağış miktarı gibi hava parametrelerini ölçen bir dizi araç içerir. Bu cihazların her biri, periyodik olarak parametre okumaları alan ve cihazlardan toplanan verileri yöneten bir yazılım sistemi tarafından kontrol edilir.

# Hava durumu istasyonunun çevresi



# Hava durumu bilgi sistemi

---



## ✧ Meteoroloji istasyonu sistemi

- Bu sistem, hava durumu verilerinin toplanmasından, bazı başlangıç verilerinin işlenmesinden ve veri yönetim sistemine iletilmesinden sorumludur.

## ✧ Veri yönetimi ve arşivleme sistemi

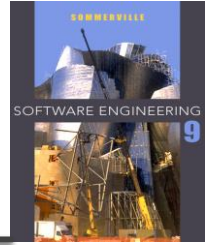
- Bu sistem, tüm doğa hava durumu istasyonlarından verileri toplar, veri işleme ve analizi gerçekleştirir ve verileri arşivler.

## ✧ İstasyon bakım sistemi

- Bu sistem, doğa hava durumu sistemlerinin sağlığını izlemek ve sorunların raporlarını elde etmek için tüm doğa hava istasyonlarıyla uydu aracılığıyla iletişim kurabilir.

# Ek yazılım işlevi

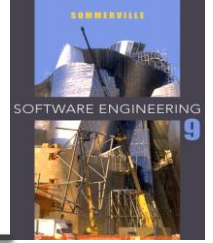
---



- ✧ Cihazları, güç ve iletişim donanımını izle ve hataları yönetim sistemine bildir.
- ✧ Sistem gücünü yönetin: çevresel koşullar izin verdiğinde pilleri şarj edin ve aynı zamanda kuvvetli rüzgar gibi potansiyel olarak zararlı hava koşullarında jeneratörlerin kapatılmasını sağlayın.
- ✧ Dinamik yeniden yapılandırmayı destekleyin: yazılımın belirli kısımları yeni sürümler ile değiştirilebilmeli ve sistem arızası durumunda yedekleme araçları sisteme alınabilmeli.

# Anahtar noktalar

---



- ✧ Yazılım mühendislerinin mühendislik mesleğine ve topluma karşı sorumlulukları vardır. Sadece teknik konularla ilgilenmemelidirler.
- ✧ Meslek toplulukları, üyelerinden beklenen davranış standartlarını belirleyen davranış kuralları yayınlar.
- ✧ Bu kitapta üç örnek olay incelemesi kullanılıyor:
  - Gömülü bir insülin pompası kontrol sistemi
  - Akıl sağlığı hasta yönetimi için bir sistem
  - Doğa hava istasyonu sistemi