

IT522 – Yazılım Mühendisliği 2021



PhD Furkan Gözükkara, Toros University

<https://github.com/FurkanGozukara/Yazilim-Muhendisligi-IT522-2021>

Ders 1 Giriş



SOFTWARE ENGINEERING

Ninth Edition

Ian Sommerville

Ders 1'de İşlenen Konular



✧ Profesyonel yazılım geliştirme

- Yazılım mühendisliği ile ne kastedilmektedir.

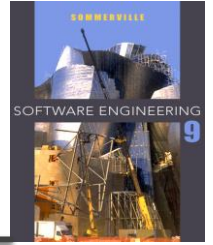
✧ Yazılım mühendisliği etiği

- Yazılım mühendisliğini etkileyen etik konulara kısa bir giriş.

✧ Durum çalışmaları

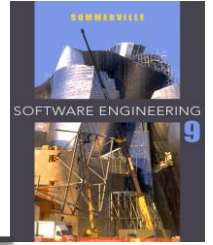
- Kitabın sonraki bölümlerinde kullanılan üç örneğe giriş.

Yazılım Mühendisliği



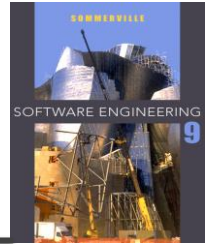
- ✧ TÜM gelişmiş ülkelerin ekonomileri yazılıma bağlıdır.
- ✧ Giderek daha fazla sistem yazılım kontrollü hale gelmektedir.
- ✧ Yazılım mühendisliği, profesyonel yazılım geliştirmeye yönelik teoriler, yöntemler ve araçlarla ilgilenir.
- ✧ Yazılım harcamaları, tüm gelişmiş ülkelerde Gayri Safi Milli Hasıla'nın (GSMH) önemli bir bölümünü temsil etmektedir.

Yazılım Maliyetleri



- ✧ Yazılım maliyetleri genellikle bilgisayar sistemi maliyetlerinin çoğunluğunu oluşturmaktadır. Bir bilgisayardaki yazılımın maliyeti genellikle donanım maliyetinden daha yüksektir.
- ✧ Yazılımın bakım maliyeti genellikle geliştirmekten daha pahalıdır. Uzun ömürlü sistemler için bakım maliyetleri, geliştirme maliyetlerinin birkaç katı olabilir.
- ✧ Yazılım mühendisliği, fiyat-performans odaklı yazılım geliştirme ile ilgilenir.

Yazılım Ürünleri



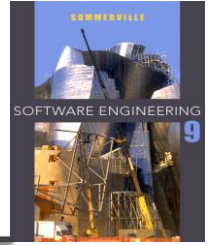
✧ Jenerik ürünler

- Pazarlanan ve satın almak isteyen herhangi bir müşteriye satılan bağımsız sistemler.
- Örnekler - grafik programları, proje yönetimi araçları gibi PC yazılımları; bilgisayar destekli tasarım (CAD) yazılımı; diş hekimleri için randevu sistemleri gibi belirli pazarlar için yazılımlar, Microsoft Windows işletim sistemi, vb.

✧ Özelleştirilmiş ürünler

- Belirli bir müşteri tarafından kendi ihtiyaçlarını karşılamak için yaptırılan yazılım.
- Örnekler - gömülü kontrol sistemleri, hava trafik kontrol yazılımı, trafik izleme sistemleri.

Ürün Özellikleri



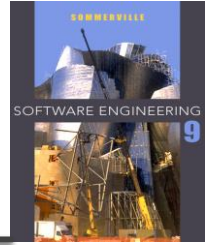
✧ Jenerik ürünler

- Yazılımın ne yapması gerektiğinin özelliği yazılım geliştiriciye aittir ve yazılım değişikliği ile ilgili kararlar geliştirici tarafından verilir.

✧ Özelleştirilmiş ürünler

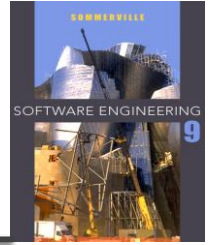
- Yazılımın ne yapması gerektiğinin özellikleri, yazılım yaptırtan müşteriye aittir ve gerekli yazılım değişiklikleri hakkında kararları müşteri belirler.

Yazılım mühendisliği hakkında sıkça sorulan sorular



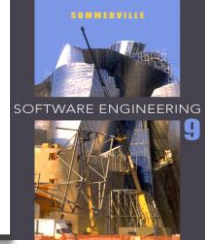
Soru	Cevap
Yazılım nedir?	Bilgisayar programları ve ilgili belgeler. Yazılım ürünleri belirli bir müşteri için geliştirilebilir veya genel bir pazar için geliştirilebilir.
İyi bir yazılımın özellikleri nelerdir?	İyi yazılım, kullanıcıya gerekli işlevselliği ve performansı sağlamalı ve sürdürülebilir, güvenilir ve kullanılabilir olmalıdır.
Yazılım mühendisliği nedir?	Yazılım mühendisliği, yazılım üretiminin tüm yönleriyle ilgilenen bir mühendislik disiplindir.
Temel yazılım mühendisliği faaliyetleri nelerdir?	Yazılım özellikleri, yazılım geliştirme, yazılım doğrulama ve yazılım evrimi.
Yazılım mühendisliği ile bilgisayar bilimi arasındaki fark nedir?	Bilgisayar bilimi teori ve temellere odaklanır; yazılım mühendisliği, kullanışlı yazılım geliştirme ve sunmanın pratiklikleriyle ilgilenir.
Yazılım mühendisliği ile sistem mühendisliği arasındaki fark nedir?	Sistem mühendisliği, donanım, yazılım ve süreç mühendisliği dahil olmak üzere bilgisayar tabanlı sistem geliştirmenin tüm yönleriyle ilgilenir. Yazılım mühendisliği, bu daha genel sürecin bir parçasıdır.

Yazılım mühendisliği hakkında sıkça sorulan sorular



Soru	Cevap
Yazılım mühendisliğinin karşılaştığı temel zorluklar nelerdir?	Artan çeşitlilikle başa çıkmak, daha kısa teslimat süreleri talep etmek ve güvenilir yazılım geliştirmek.
Yazılım mühendisliğinin maliyetleri nelerdir?	Yazılım maliyetlerinin kabaca% 60'ı geliştirme maliyetidir,% 40'ı test maliyetidir. Özel yazılımlar için, geliştirme maliyetleri genellikle geliştirme maliyetlerini aşar.
En iyi yazılım mühendisliği teknikleri ve yöntemleri nelerdir?	Tüm yazılım projelerinin profesyonelce yönetilmesi ve geliştirilmesi gerekirken, farklı sistem türleri için farklı teknikler uygundur. Örneğin, oyunlar her zaman bir dizi prototip kullanılarak geliştirilmelidir, oysa güvenlik açısından kritik kontrol sistemleri geliştirilecek eksiksiz ve analiz edilebilir bir spesifikasyon gerektirir. Bu nedenle, bir yöntemin diğerinden daha iyi olduğunu söyleyemezsiniz.
Web, yazılım mühendisliğinde ne gibi farklılara sebep oldu?	Web, yazılım hizmetlerinin kullanılabilirliğine ve yüksek düzeyde dağıtılmış hizmet tabanlı sistemler geliştirme imkanını sağlamıştır. Web tabanlı sistem geliştirme, programlama dillerinde ve yazılımın yeniden kullanımında önemli gelişmelere yol açmıştır.

İyi Bir Yazılımın Temel Nitelikleri



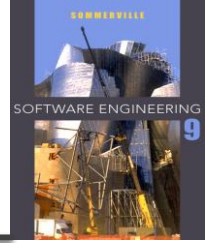
Ürün Karakteristiği	Açıklama
Sürdürülebilirlik	Yazılım, müşterilerin değişen ihtiyaçlarını karşılayacak şekilde gelişebilecek şekilde yazılmalıdır. Bu kritik bir özelliktir çünkü yazılım değişikliği, değişen bir iş ortamının kaçınılmaz bir gereğidir.
Güvenilebilirlik ve güvenlik	Yazılım güvenilirliği, güvenilirlik, güvenlik ve emniyet gibi bir dizi özelliği içerir. Güvenilir yazılım, sistem arızası durumunda fiziksel veya ekonomik hasara neden olmamalıdır. Kötü niyetli kullanıcılar sisteme erişememeli veya sisteme zarar vermemelidir.
Verimlilik	Yazılım, bellek ve işlemci döngüleri gibi sistem kaynaklarını boşa harcamamalıdır. Bu nedenle verimlilik, yanıt verme, işlem süresi, bellek kullanımı vb. içerir.
Kabuledilebilirlik	Yazılım, tasarlandığı kullanıcı türü için kabul edilebilir olmalıdır. Bu, anlaşılır, kullanılabilir ve kullandıkları diğer sistemlerle uyumlu olması gerektiği anlamına gelir.

Yazılım Mühendisliği



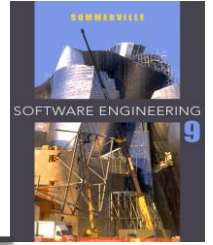
- ✧ Yazılım mühendisliği, sistem spesifikasyonunun ilk aşamalarından sistemin kullanıma girdikten sonra bakımına kadar yazılım üretiminin tüm yönleriyle ilgilenen bir mühendislik disiplini.
- ✧ Mühendislik disiplini
 - Organizasyonel ve finansal kısıtlamaları dikkate alan problemleri çözmek için uygun teorileri ve yöntemleri kullanmak.
- ✧ Yazılım üretiminin tüm yönleri
 - Sadece teknik geliştirme süreci değil. Ayrıca yazılım üretimini desteklemek için proje yönetimi ve araçların, yöntemlerin vb. geliştirilmesi.

Yazılım Mühendisliğinin Önemi



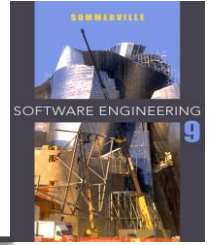
- ✧ Giderek daha çok bireyler ve toplum gelişmiş yazılım sistemlerine güveniyor. Güvenilir ve emin sistemleri ekonomik ve hızlı bir şekilde üretebilmemiz gerekiyor.
- ✧ Uzun vadede, programları kişisel bir programlama projesiymiş gibi yazmaktansa, yazılım sistemleri için yazılım mühendisliği yöntem ve tekniklerini kullanmak genellikle daha ucuzdur. Çoğu sistem türü için, maliyetlerin çoğu, yazılımı kullanıma girdikten sonra değiştirmenin maliyetidir.

Yazılım Süreci Faaliyetleri



- ✧ Müşterilerin ve mühendislerin üretilecek yazılımı ve çalışması üzerindeki kısıtlamaları tanımladığı yazılım spesifikasyonu.
- ✧ Yazılımın tasarlandığı ve programlandığı yazılım geliştirme.
- ✧ Yazılımın, müşterinin ihtiyaç duyduğundan emin olmak için kontrol edildiği yazılım doğrulama.
- ✧ Yazılımın değişen müşteri ve pazar gereksinimlerini yansıtacak şekilde değiştirildiği yazılım evrimi.

Çoğu Yazılımı Etkileyen Genel Sorunlar



✧ Heterojenlik

- Giderek artan bir şekilde, sistemlerin farklı bilgisayar ve mobil cihaz türlerini de içeren ağlar arasında dağıtılmış sistemler olarak çalışması gerekmektedir.

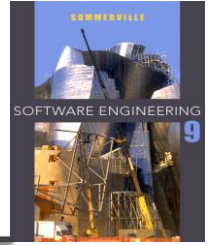
✧ İş ve sosyal değişim

- Yükselen ekonomiler geliştikçe ve yeni teknolojiler kullanılabilir hale geldikçe, iş ve toplum inanılmaz derecede hızlı değişiyor. Mevcut yazılımlarını değiştirebilmeleri ve yeni yazılımları hızla geliştirebilmeleri gerekiyor.

✧ Güvenlik ve itimat

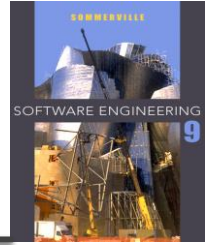
- Yazılım hayatımızın tüm yönleriyle iç içe olduğundan, o yazılıma güvenebilmemiz çok önemlidir.

Yazılım mühendisliği çeşitliliği



- ✧ Birçok farklı yazılım sistemi türü vardır ve bunların tümü için geçerli olan evrensel bir yazılım teknikleri seti yoktur.
- ✧ Kullanılan yazılım mühendisliği yöntemleri ve araçları, geliştirilmekte olan uygulamanın türüne, müşterinin gereksinimlerine ve geliştirme ekibinin tecrübesine bağlıdır.

Uygulama türleri



✧ Bağımsız uygulamalar

- 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.

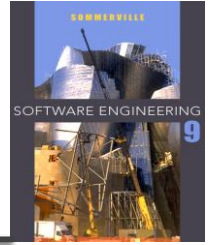
✧ Etkileşimli işlem tabanlı uygulamalar

- Uzak bir bilgisayarda çalışan ve kullanıcılar tarafından kendi bilgisayarlarından veya terminallerinden erişilen uygulamalar. Bunlar, e-ticaret uygulamaları gibi web uygulamalarını içerir.

✧ Gömülü kontrol sistemleri

- Bunlar, donanım cihazlarını kontrol eden ve yöneten yazılım kontrol sistemleridir. Toplam cihaz sayısı olarak, muhtemelen diğer sistem türlerinden daha fazla gömülü sistem vardır.

Uygulama Türleri



✧ Toplu işleme sistemleri

- Bunlar, verileri büyük gruplar halinde işlemek için tasarlanmış iş sistemleridir. Karşılık gelen çıktıları oluşturmak için çok sayıda bağımsız girdiyi işlerler.

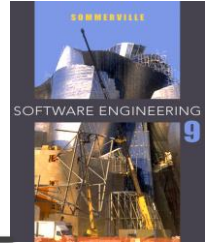
✧ Eğlence sistemleri

- Bunlar, öncelikle kişisel kullanım için olan ve kullanıcıyı eğlendirmeyi amaçlayan sistemlerdir.

✧ Modelleme ve simülasyon sistemleri

- Bunlar, bilim adamları ve mühendisler tarafından, birçok, ayrı, etkileşimli nesneyi içeren fiziksel süreçleri veya durumları modellemek için geliştirilen sistemlerdir.

Uygulama Türleri



✧ Veri toplama sistemleri

- Bunlar, bir dizi sensör kullanarak ortamlarından veri toplayan ve bu verileri işlenmek üzere diğer sistemlere gönderen sistemlerdir.

✧ Sistem sistemleri

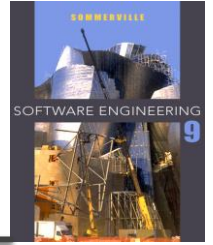
- Bunlar, bir dizi başka yazılım sistemlerinden oluşan sistemlerdir.

Yazılım Mühendisliğinin Temelleri



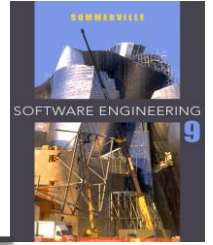
- ✧ Bazı temel ilkeler, kullanılan geliştirme tekniklerine bakılmaksızın her tür yazılım sistemi için geçerlidir:
- Sistemler, yönetilen ve anlaşılan bir geliştirme süreci kullanılarak geliştirilmelidir. Elbette farklı yazılım türleri için farklı süreçler kullanılmaktadır.
 - Güvenilirlik ve performans tüm sistem türleri için önemlidir.
 - Yazılım özelliklerini ve gereksinimlerini (yazılımın ne yapması gerektiği) anlamak ve yönetmek önemlidir.
 - Uygun olduğu durumlarda, yeni yazılım yazmak yerine zaten geliştirilmiş olan yazılımı yeniden kullanmalısınız.

Yazılım Mühendisliği Ve Web



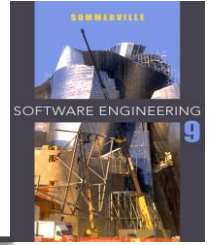
- ✧ Web artık uygulamaları çalıştırmak için bir platformdur ve kuruluşlar yerel sistemler yerine giderek daha fazla web tabanlı sistemler geliştirmektedir.
- ✧ Web hizmetleri (Ders 19'da tartışılmıştır), uygulama işlevselliğine web üzerinden erişilmesine izin verir.
- ✧ Bulut bilişim, uygulamaların uzaktan 'bulut' üzerinde çalıştığı bilgisayar hizmetlerinin sağlanmasına yönelik bir yaklaşımdır.
 - Kullanıcılar yazılım satın almazlar, kullanıma göre ödeme yaparlar.

Web Yazılım Mühendisliği



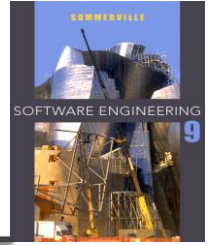
- ✧ Yazılımın yeniden kullanımı, web tabanlı sistemler oluşturmak için baskın yaklaşımdır.
 - Bu sistemleri oluştururken, bunları önceden var olan yazılım bileşenleri ve sistemlerinden nasıl bir araya getirebileceğinizi düşünürsünüz.
- ✧ Web tabanlı sistemler aşamalı olarak geliştirilmeli ve sunulmalıdır.
 - Bu tür sistemler için tüm gereksinimleri önceden belirlemenin pratik olmadığı artık genel olarak kabul edilmektedir.
- ✧ Kullanıcı arayüzleri, web tarayıcılarının yetenekleriyle sınırlıdır.
 - AJAX gibi teknolojiler, bir web tarayıcısı içinde zengin arayüzlerin oluşturulmasına izin verir, ancak yine de kullanımı zordur. Yerel komut dosyası içeren web formları daha yaygın olarak kullanılır.

Web Tabanlı Yazılım Mühendisliği



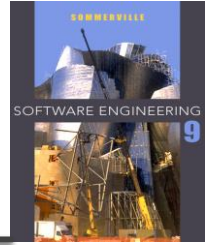
- ✧ Web tabanlı sistemler karmaşık dağıtık sistemlerdir, ancak daha önce tartışılan yazılım mühendisliğinin temel ilkeleri, diğer sistem türlerine olduğu kadar onlar için de geçerlidir.
- ✧ Bir önceki bölümde tartışılan yazılım mühendisliğinin temel fikirleri, diğer yazılım sistemi türlerine uygulandıkları gibi web tabanlı yazılımlar için de geçerlidir.

Anahtar Noktalar



- ✧ Yazılım mühendisliği, yazılım üretiminin tüm yönleriyle ilgilenen bir mühendislik disiplini.
- ✧ Temel yazılım ürünü özellikleri, sürdürülebilirlik, güvenilirlik ve itimat, verimlilik ve kabul edilebilirliktir.
- ✧ Üst düzey spesifikasyon, geliştirme, doğrulama ve güncelleme faaliyetleri, tüm yazılım süreçlerinin bir parçasıdır.
- ✧ Yazılım mühendisliğinin temel kavramları evrensel olarak tüm sistem geliştirme türlerine uygulanabilir.

Anahtar Noktalar

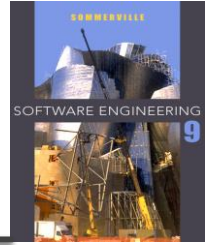


- ✧ Pek çok farklı sistem türü vardır ve bunların her biri, geliştirilmeleri için uygun yazılım mühendisliği araçları ve teknikleri gerektirir.
- ✧ Yazılım mühendisliğinin temel fikirleri her tür yazılım sistemine uygulanabilir.

Ders 1 - Giriş

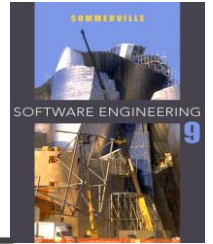
Bölüm 2

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



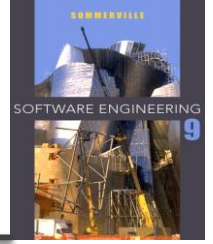
✧ Gizlilik

- 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



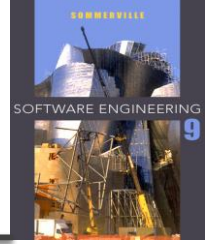
✧ Fikri Mülkiyet Hakları

- 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

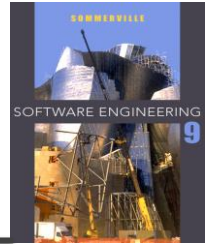
- Yazılım mühendisleri, teknik becerilerini başkalarının bilgisayarlarını kötüye kullanmak için kullanmamalıdır. Bilgisayarın kötüye kullanımı, görece önemsizden (örneğin bir işverenin makinesinde oyun oynama) son derece ciddiye (virüslerin yayılması) kadar çeşitlilik gösterir.

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



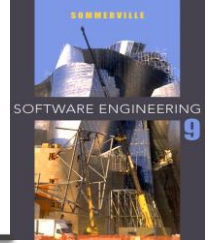
- ✧ ABD'deki profesyonel topluluklar, bir etik uygulama kodu üretmek için işbirliği yaptılar.
- ✧ Bu kuruluşların üyeleri, katıldıklarında uygulama kurallarına kaydolurlar.
- ✧ Kurallar, uygulayıcılar, eğitimciler, yöneticiler, süpervizörler ve politika yapıcıların yanı sıra mesleğin stajyerleri ve öğrencileri 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 adanmışlardır.*

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



Yazılım Mühendisliği Etik Kuralları ve Mesleki Uygulama

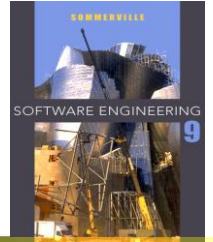
ACM/IEEE-CS Yazılım Mühendisliği Etik ve Mesleki Uygulamalar Ortak Görev Gücü

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.

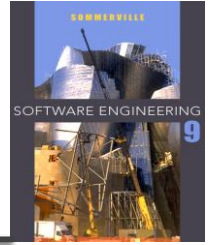
Yazılım mühendisleri, yazılımın analizini, özelliklerini, tasarımını, geliştirilmesini, test edilmesini ve bakımını yararlı ve saygın bir meslek haline getirmeye kendilerini adanmalıdır. Halkın sağlığı, güvenliği ve refahına yönelik taahhütlerine uygun olarak, yazılım mühendisleri aşağıdaki Sekiz İlkeye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

Etik İlkeler



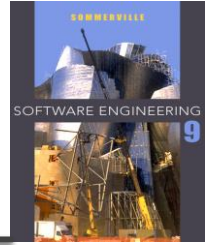
1. KAMU - Yazılım mühendisleri, kamu çıkarı için tutarlı bir şekilde hareket etmelidir.
2. MÜŞTERİ VE İŞVEREN - Yazılım mühendisleri, müşterileri ve işverenleri için kamu menfaatine uygun şekilde en yüksek menfaat olacak şekilde hareket edeceklerdir.
3. ÜRÜN - Yazılım mühendisleri, ürünlerinin ve ilgili 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 İkilemler



- ✧ Üst yönetimin politikaları ile prensipte anlaşmazlık.
- ✧ İşverenin 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

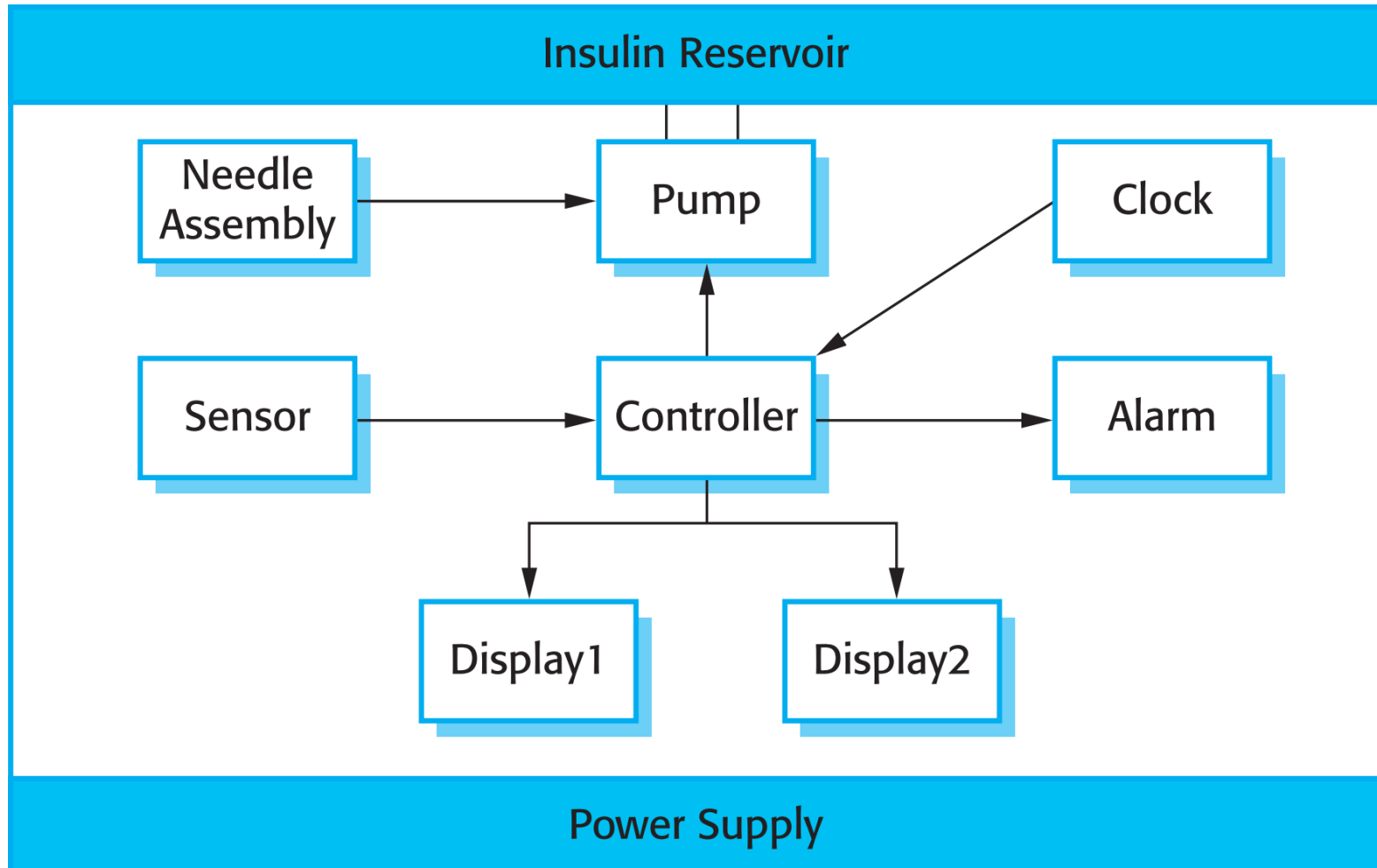
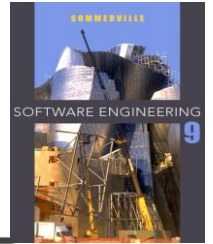
- Uzak bölgelerdeki hava koşulları hakkında veri toplayan bir veri toplama sistemi.

İnsülin Pompası Kontrol Sistemi

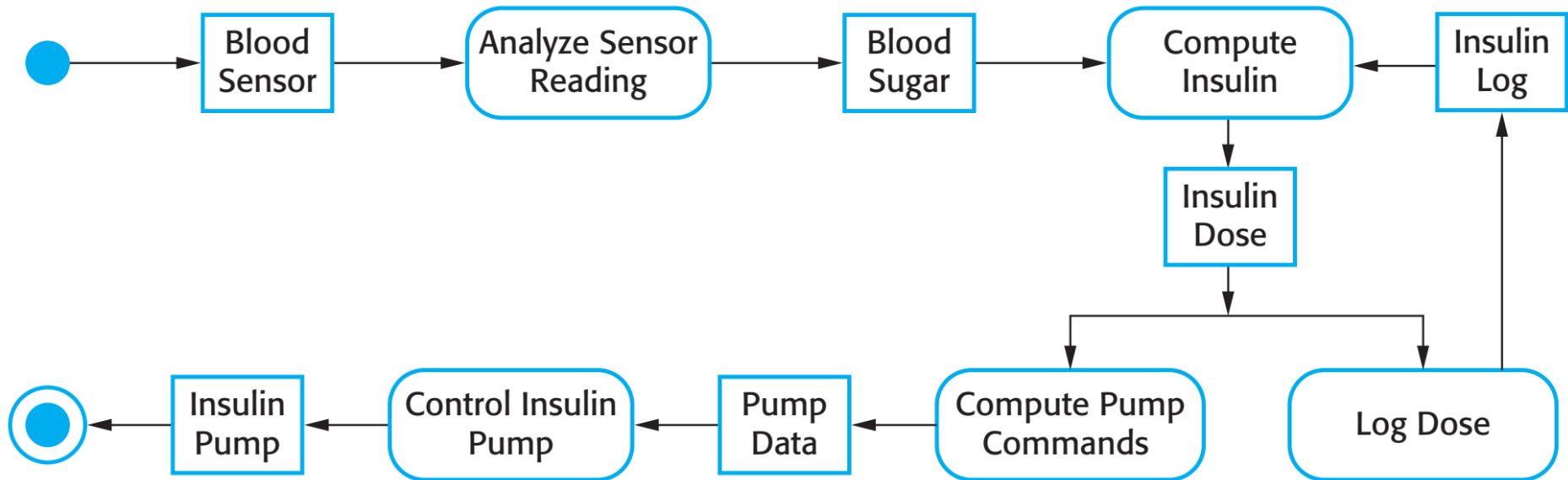
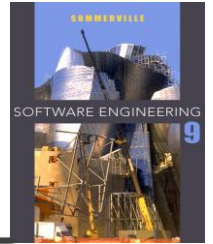


- ✧ Bir kan şekeri sensöründen veri toplar ve enjekte edilmesi gereken insülin miktarını hesaplar.
- ✧ Kan şekeri seviyelerinin değişim oranına göre hesaplama.
- ✧ Doğru insülin dozunu iletmek için bir mikro pompaya sinyal gönderir.
- ✧ Düşük kan şekeri beyin hasarına, komaya ve ölüme yol açabileceğinden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

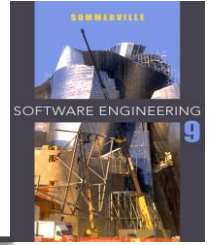
İnsülin Pompası Donanım Mimarisi



İnsülin Pompasının Aktivite Modeli

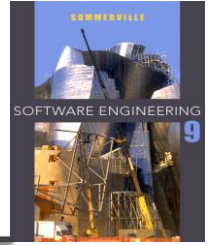


Temel Üst Düzey Gereksinimler

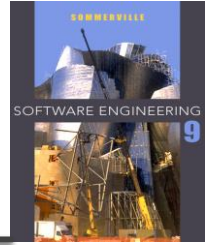


- ✧ Sistem, gerektiğinde insülin verebilecek durumda olacaktır.
- ✧ Sistem güvenilir bir şekilde çalışacak ve mevcut kan şekeri düzeyini dengelemek için doğru miktarda insülin gönderecektir.
- ✧ Bu nedenle sistem, sistemin her zaman bu gereksinimleri karşıladığından emin olmak için tasarlanmalı ve uygulanmalıdır.

Akıl Sağlığı Bakımı İçin Bir Hasta Bilgi Sistemi

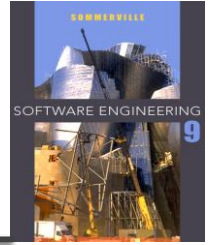


- ✧ Akıl sağlığı hizmetlerini desteklemek için bir hasta bilgi sistemi: akıl sağlığı sorunları olan hastalar ve gördükleri tedaviler hakkında bilgi tutan tıbbi bir bilgi sistemidir.
- ✧ Çoğu akıl sağlığı hastası, özel bir hastanede tedaviye ihtiyaç duymaz, ancak sorunları hakkında ayrıntılı bilgiye sahip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



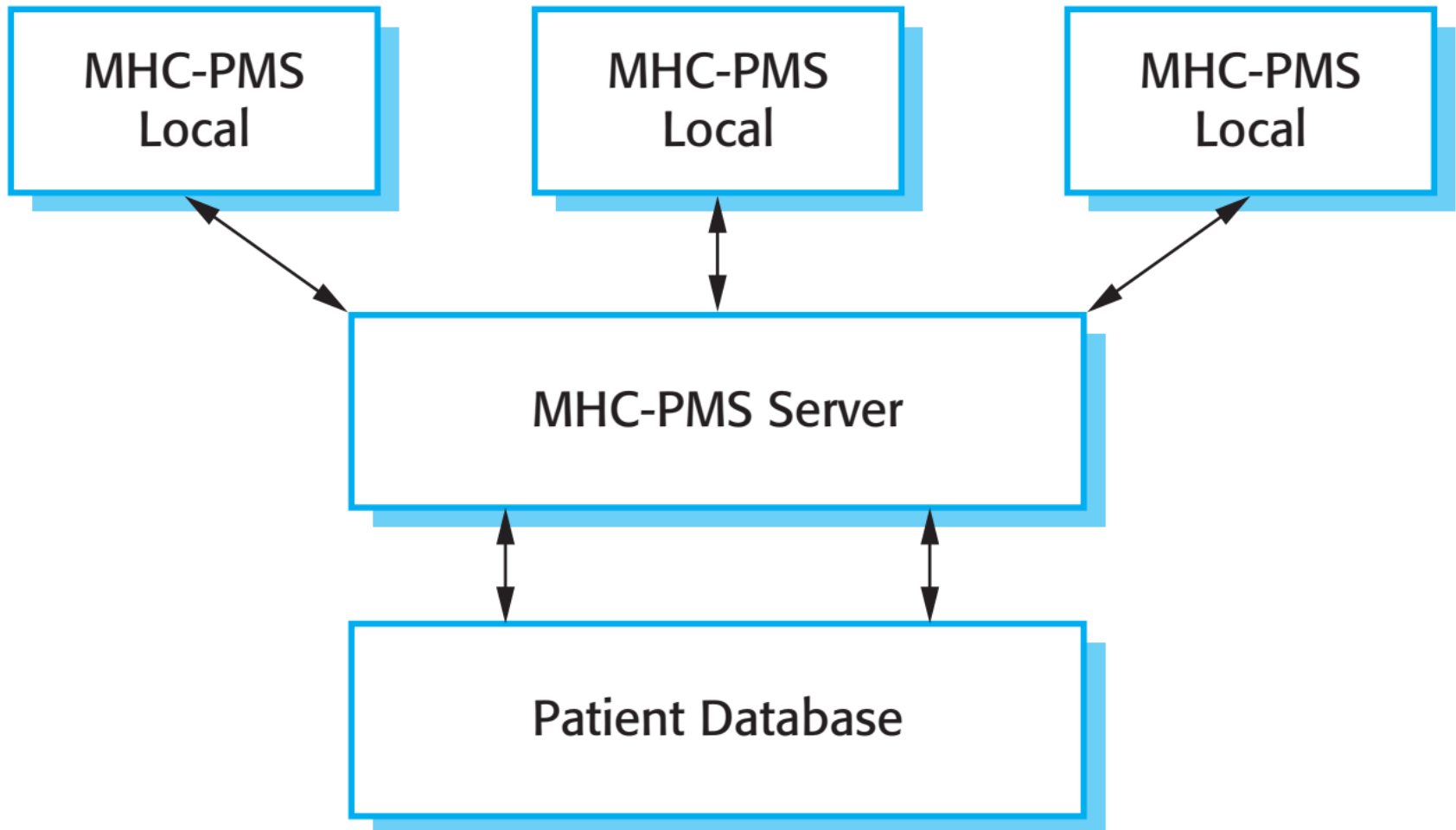
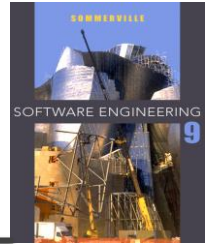
- ✧ AS-HYS (Akıl Sağlığı Hasta Yönetim Sistemi) kliniklerde kullanılmak üzere tasarlanmış bir bilgi sistemidir.
- ✧ Hasta bilgilerinin tutulduğu merkezi bir veri tabanını kullanır, ancak aynı zamanda bir PC üzerinde çalışacak şekilde tasarlanmıştır, böylece güvenli ağ bağlantısı olmayan cihazlardan erişilebilir 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.

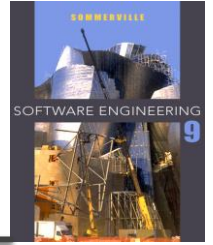
✧ Hasta izleme

- 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



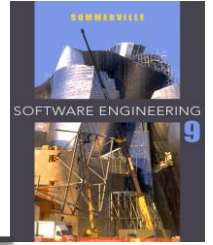
✧ Gizlilik

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

✧ Güvenlik

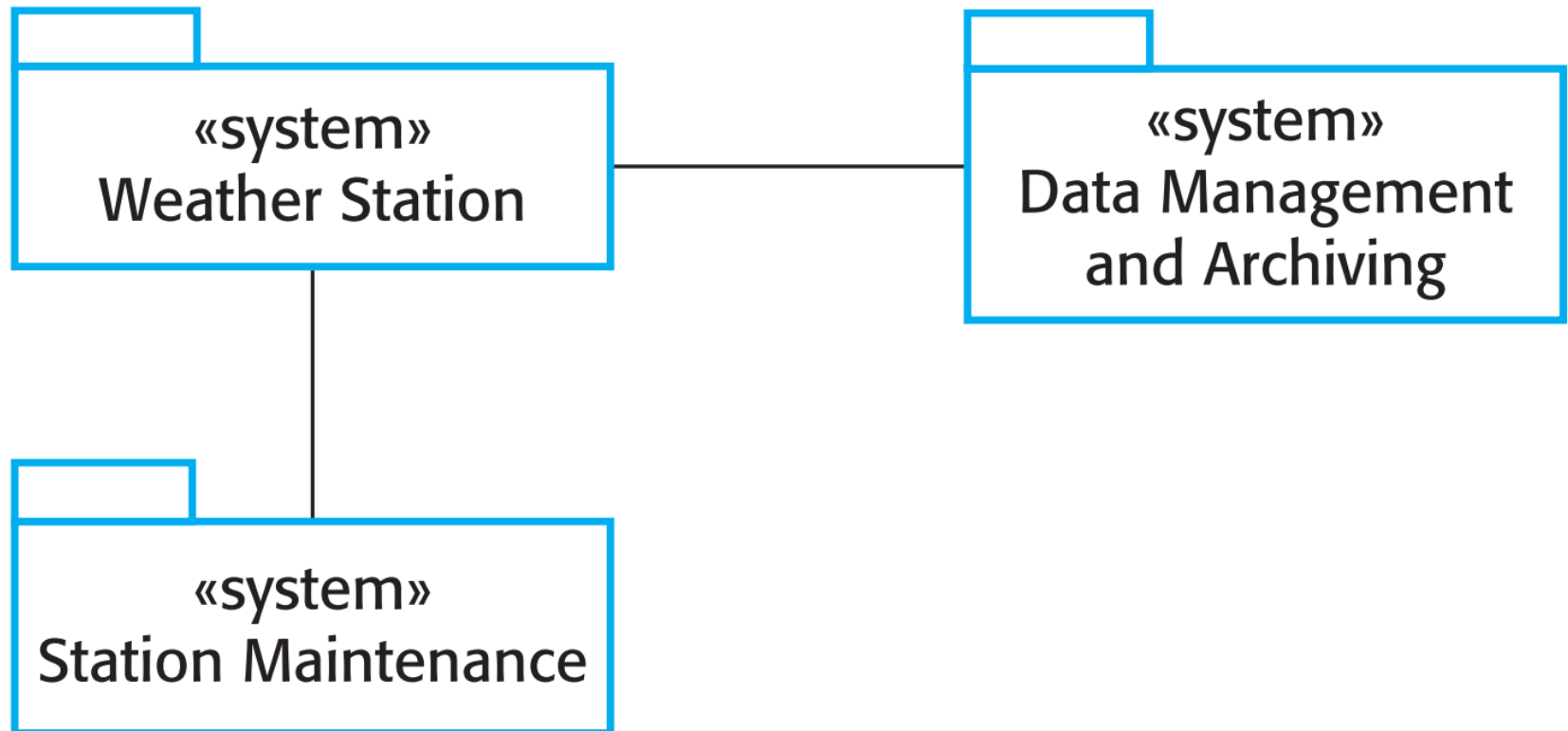
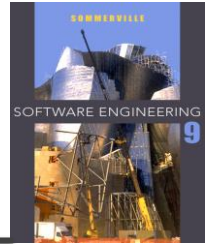
- Bazı akıl hastalıkları, hastaların intihar etmesine veya başkaları için tehlike oluşturmalarına neden olur. Mümkün olan her yerde, sistem tıbbi personeli intihara meyilli veya tehlikeli hastalar konusunda uyarmalıdır.
- Sistem ihtiyaç duyulduğunda hazır olmalıdır, aksi takdirde güvenlik tehlikeye girebilir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

Doğa Hava İstasyonu

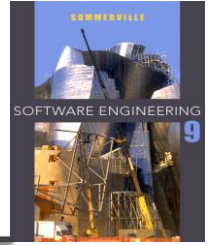


- ✧ Geniş vahşi alanlara sahip bir ülkenin hükümeti uzak bölgelere birkaç yüz meteoroloji istasyonu yerleştirmeye karar verir.
- ✧ Hava durumu istasyonları, sıcaklığı ve basıncı, güneş ışığını, yağmuru, rüzgar hızını ve rüzgar yönünü ölçen bir dizi cihazdan veri toplar.
 - Hava durumu istasyonu, rüzgar 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 İstasyonunun Çevresi



Hava Durumu Bilgi Sistemi



✧ Meteoroloji istasyonu sistemi

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

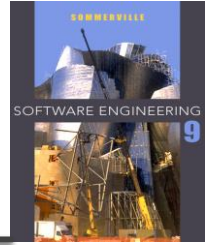
✧ Veri yönetimi ve arşivleme sistemi

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

✧ İstasyon bakım sistemi

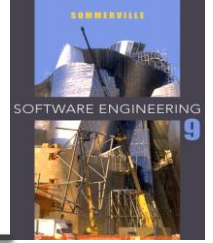
- Bu sistem, doğa hava durumu sistemlerinin sağlığını izlemek ve sorunların raporlarını elde etmek için tüm doğa hava istasyonlarıyla uydu aracılığıyla iletişim kurabilir.

Ek Yazılım İşlevi



- ✧ 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