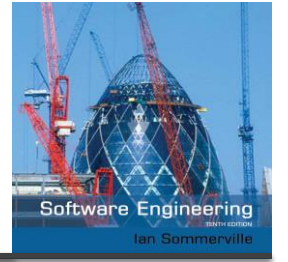


Bölüm 1- Giriş



✧ Profesyonel yazılım geliştirme

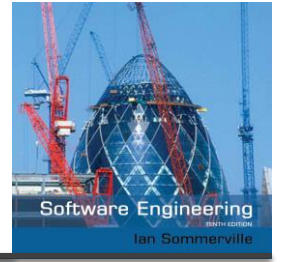
- Yazılım mühendisliği ne demek.

✧ Yazılım mühendisliği etiği

- Yazılım mühendisliğini etkileyen etik ve mesleki konular neler.

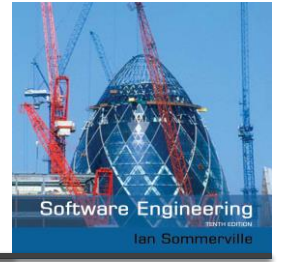
✧ Durum çalışmaları

- Bütün kitapta örnek olarak kullanılan farklı türlerdeki sistemleri tanıyacaksınız.



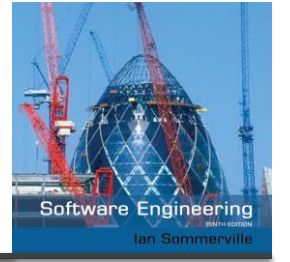
- ✧ TÜM gelişmiş ulusların ekonomileri yazılıma bağımlıdır.
- ✧ Giderek daha fazla sistem yazılım kontrollüdür.
- ✧ Yazılım mühendisliği, profesyonel yazılım geliştirme için teoriler, yöntemler ve araçlarla ilgilenir.
- ✧ Yazılıma yapılan harcamalar, tüm gelişmiş ülkelerde GSMH'nin önemli bir bölümünü temsil etmektedir.

Yazılım maliyetleri



- ✧ Yazılım maliyetleri genellikle bilgisayar sistemi maliyetlerine hakimdir. Bir PC'deki yazılımın maliyeti genellikle donanım maliyetinden daha fazladır.
- ✧ Yazılımın bakımı, geliştirilmesinden daha maliyetlidir. Uzun ömürlü sistemler için bakım maliyetleri geliştirme maliyetlerinin birkaç katı olabilir.
- ✧ Yazılım mühendisliği, uygun maliyetli yazılım geliştirme ile ilgilidir.

Yazılım projesi hataları

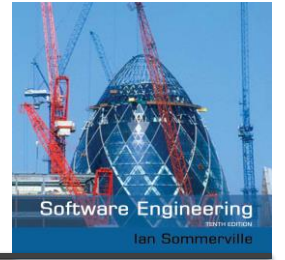


✧ *Artan sistem karmaşıklığı*

- Yeni yazılım mühendisliği teknikleri **daha büyük, daha karmaşık** sistemler oluşturmamıza yardım ederken, **talepler değişmektedir.**
- Sistemler **daha hızlı geliştirilmeli ve teslim edilmeli**; daha büyük, ve hatta daha karmaşık sistemlere gereksinim var ve sistemler daha önceden imkansız olarak düşünülen **yeni yeteneklere sahip olmalı.**
- Daha karmaşık yazılım sunabilmenin zorluklarını karşılayabilmek için yeni yazılım mühendisliği teknikleri geliştirilmelidir.

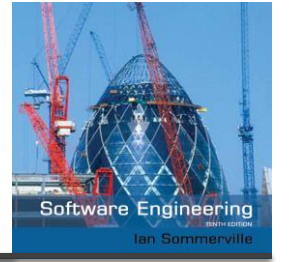
✧ *Yazılım mühendisliği yöntemlerini kullanmadaki başarısızlık.*

- Bilgisayar programlarını yazılım mühendisliği, yöntemlerini ve tekniklerini kullanmadan yazmak oldukça kolaydır.
- Birçok şirket ürünleri ve servisler geliştikçe yazılım geliştirmeye sürüklenmişlerdir.
- Günlük işlerinde yazılım mühendisliği yöntemlerini kullanmazlar.
- Bu nedenle genellikle yazılımları olması gerekenden daha pahalı ve daha az güvenilirdir.
- Bu probleme yönelmek için daha iyi yazılım mühendisliği öğretimine ve eğitime ihtiyacımız var.



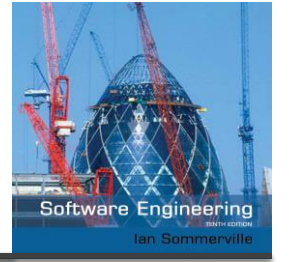
Profesyonel yazılım geliştirme

Yazılım mühendisliği hakkında sık sorulan sorular

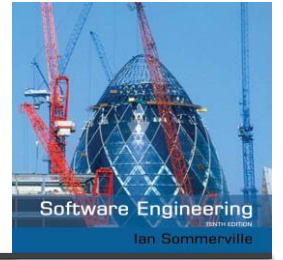


Soru	Cevap
Yazılım nedir?	Bilgisayar programları ve ilişkili belgeleme. Yazılım ürünleri belirli bir müşteri için geliştirilebilir veya genel bir pazar için geliştirilebilir.
İyi yazılımın özellikleri nelerdir?	İyi yazılım kullanıcıya gereken fonksiyonelliği ve performansı sağlamalı ve bakımı yapılabilir, güvenilebilir ve kullanılabilir olmalıdır.
Yazılım mühendisliği nedir?	Yazılım mühendisliği yazılım üretiminin başlangıçtaki ilk fikirden işletim ve bakıma kadar olan tüm yönleriyle ilgilenen bir mühendislik disiplini.
Temel yazılım mühendisliği etkinlikleri nelerdir?	Yazılım spesifikasyonu, yazılım geliştirme, yazılım doğrulama ve yazılımın evrimi.
Yazılım mühendisliği ve bilgisayar bilimleri arasındaki fark nedir?	Bilgisayar bilimleri teori ve temeller üzerine odaklanır; yazılım mühendisliği ise kullanışlı yazılım üretim ve tesliminin uygulamalarıyla ilgilenir.
Yazılım mühendisliği ve sistem mühendisliği arasındaki fark nedir?	Sistem mühendisliği donanım, yazılım ve süreç mühendisliği dâhil olmak üzere bilgisayar tabanlı sistem geliştirmenin tüm yönleriyle ilgilenir. Yazılım mühendisliği bu daha genel sürecin parçasıdır.

Yazılım mühendisliği hakkında sık sorulan sorular



Soru	Cevap
Yazılım mühendisliği ile ilgili esas zorluklar nelerdir?	Artan çeşitlilikle başa çıkma, daha kısa teslim süresi talepleri ve güvenilir yazılım geliştirme.
Yazılım mühendisliğinin maliyetleri nelerdir?	Yazılım maliyetlerinin kabaca %60'ı geliştirme maliyetleridir, %40'ı sınamaya maliyetleridir. Özel ısmarlama yazılım için, evrim maliyetleri genellikle geliştirme maliyetlerini aşar.
En iyi yazılım mühendisliği teknikleri ve yöntemleri nelerdir?	Tüm yazılım projelerinin profesyonel olarak yönetilmesi ve geliştirilmesi gerekir ama farklı sistem türleri için farklı teknikler uygundur. Örneğin oyunlar her zaman bir dizi prototip kullanılarak geliştirilmeliyken, hayati tehlike arz eden kontrol sistemleri tam ve analiz edilebilir bir spesifikasyonun geliştirilmesini gerektirirler. Her şey için iyi olan yöntemler ve teknikler yoktur.
İnternet yazılım mühendisliğinde ne değişiklik yapmıştır?	İnternet sadece çok büyük, çok dağıtık servis-tabanlı sistemlerin geliştirilmesine yol açmamıştır, aynı zamanda yazılımın ekonomisini değiştiren mobil cihazlar için bir uygulama (<i>app</i>) endüstrisi yaratılmasını desteklemiştir.

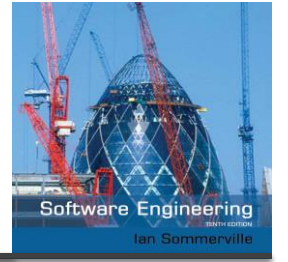


✧ Genel özellikli (jeneric) yazılımlar.

- Serbest piyasada onları satın alabilen herhangi bir müşteriye pazarlanan ve satılan bağımsız sistemlerdir..
- Örnekler– Mobil uygulamalar, grafik programları, proje yönetim araçları vb., bu tür yazılımlar aynı zamanda ; muhasebe sistemleri veya diş tedavileri ile ilgili kayıtları tutan sistemler gibi belirli bir pazar için tasarlanmış uygulamaları içerir..

✧ Ismarlama yazılımlar

- Belirli bir müşteri tarafından kendi ihtiyaçları için ısmarlanmış, belirli bir müşteri için geliştirilen sistemlerdir.
- Örnekler– gömülü kontrol sistemleri, hava trafik kontrol yazılımı, trafik izleme sistemleri.



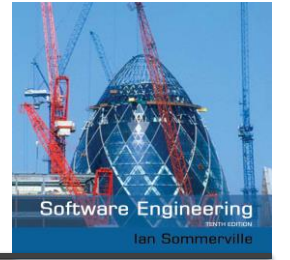
✧ Genel özellikli (jeneric) yazılımlar

- Yazılımın ne yapması gerektiğine ilişkin spesifikasyon yazılım geliştiricisine aittir ve yazılım değişikliği ile ilgili kararlar geliştirici tarafından verilir.

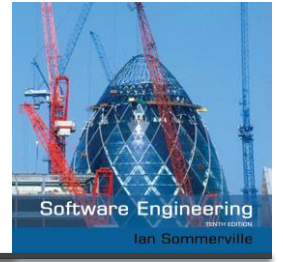
✧ Ismarlama yazılımlar

- Yazılımın ne yapması gerektiğine ilişkin spesifikasyon, yazılımı satın alan kuruluşa aittir ve gerekli yazılım değişikliklerine ilişkin kararlar müşteri tarafından verilir.

İyi yazılımın gerekli özellikleri

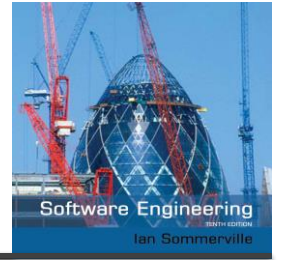


Ürün özelliği	Tanım
Kabul edilebilirlik	Yazılım tasarlandığı kullanıcı tipi tarafından kabul edilebilir olmalıdır. Bunun anlamı, anlaşılabilir, kullanılabilir ve kullandıkları diğer sistemlerle uyumlu olması gerektirir.
Güvenilebilirlik ve güvenlik	Yazılım güvenilebilirliği, güvenilirlik, güvenlik ve emniyet dâhil olmak üzere bir dizi özelliği içerir. Güvenilebilir yazılım bir sistem arızası durumunda fiziksel veya ekonomik bir zarara neden olmamalıdır. Yazılım zararlı kullanıcıların sisteme erişemeyeceği veya zarar veremeyeceği şekilde emniyetli olmalıdır.
Verimlilik	Yazılım bellek ve işlemci döngüleri gibi sistem kaynaklarını gereksiz yere kullanmamalıdır. Bu nedenle verimlilik, çabuk yanıt verme yeteneği, işlem zamanı, kaynak kullanımı vb. içerir.
Bakım kolaylığı	Yazılım müşterilerin değişen gereksinimlerini karşılamak için evrilebileceği şekilde geliştirilmelidir. Bu hayati bir özelliktir çünkü yazılım değişimi, değişen bir iş ortamının kaçınılmaz bir gereksinimidir.



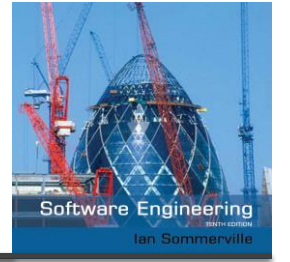
- ✧ Yazılım mühendisliği, sistem spesifikasyonunun ilk aşamalarından sistemin kullanıma verildikten sonraki bakımına kadar yazılım üretiminin tüm yönleriyle ilgilenen bir mühendislik disiplini.
- ✧ Mühendislik disiplini
 - Kurumsal ve finansal kısıtlamaları göz önünde bulundurarak sorunları çözmek için uygun teori ve yöntemleri kullanmak.
- ✧ Yazılım üretiminin tüm yönleri
 - Sadece teknik geliştirme süreci değil. Aynı zamanda yazılım proje yönetimi ve yazılım geliştirmeyi desteklemek için araçların, yöntemlerin ve teorilerin geliştirilmesi.

Yazılım mühendisliğinin önemi



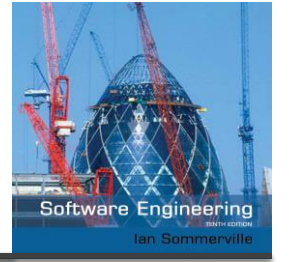
- ✧ Kişiler ve toplum, giderek daha fazla gelişmiş yazılım sistemlerine güvenmektedirler. Emniyetli ve güvenilir sistemleri ekonomik ve hızlı bir şekilde üretebilmemiz gerekiyor.
- ✧ Genellikle, uzun vadede, profesyonel yazılım sistemleri için yazılım mühendisliği yöntemlerini ve tekniklerini kullanmak, kişisel bir programlama projesiymiş gibi sadece programlar yazmaktan daha ucuzdur. Yazılım mühendisliği yöntemini kullanmamak, sinama, kalite güvence ve uzun dönemli bakım için daha yüksek maliyetlere neden olur.

Yazılım süreci faaliyetleri



- ✧ Müşterilerin ve mühendislerin üretilecek yazılımı ve işleyişindeki kısıtları tanımladıkları yazılım spesifikasyonu.
- ✧ Yazılımın tasarlandığı ve programlandığı yazılım geliştirme.
- ✧ Yazılımın, müşterinin ihtiyaçlarına uygun olduğunun sağlanması amacıyla kontrol edildiği yazılım doğrulama.
- ✧ Yazılımın değişen müşteri ve pazar gereksinimlerini yansıtacak şekilde değiştirildiği yazılım evrimi.

Yazılımı etkileyen genel sorunlar



✧ Heterojenlik

- Sistemlerin, gittikçe artarak farklı tür bilgisayar ve taşınabilir cihazlar içeren ağlar arasında dağıtılmış sistemler olarak çalışması gerekmektedir.

✧ İş ve sosyal değişim

- İş dünyası ve toplum, yeni ekonomiler geliştikçe ve yeni teknolojiler elde edildikçe inanılmaz derecede hızla değişir. Mevcut yazılımlarını değiştirebilmeye ve hızlıca yeni yazılım geliştirebilmeye gerek duyarlar.

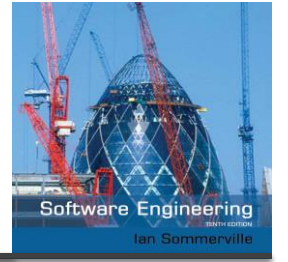
✧ Güvenlik ve güven

- Yazılım hayatımızın her alanına dahil oldukça, o yazılıma güvenebilmemiz şarttır.

✧ Ölçek

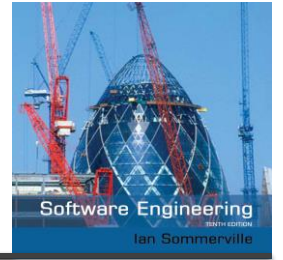
- Yazılım, taşınabilir veya giyilebilir cihazlardaki çok küçük gömülü sistemlerden küresel bir topluluğa hizmet eden İnternet ölçeğinde, bulut tabanlı sistemlere kadar çok geniş bir ölçek yelpazesinde geliştirilmelidir.

Yazılım mühendisliği çeşitliliği



- ✧ Pek çok farklı yazılım sistemi türü vardır ve bunların hepsine uygulanabilecek evrensel bir yazılım mühendisliği seti yoktur.
- ✧ Kullanılan yöntemler, araçlar ve teknikler yazılımı geliştiren kuruluşa, yazılımın türüne ve geliştirme sürecine katılan kişilere bağlıdır.

Uygulama türleri



✧ Bağımsız uygulamalar

- Bunlar, PC gibi kişisel bir bilgisayarda çalışan uygulama sistemleridir. Gerekli tüm fonksiyonelliği içerirler ve bir ağa bağlı olmaları gerekmez.

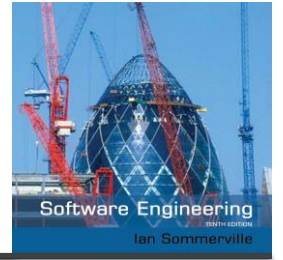
✧ Etkileşimli işlem tabanlı uygulamalar

- Uzak bir bilgisayarda çalışan ve kullanıcılar tarafından kendi bilgisayarlarından veya telefon/tabletlerinden erişilen uygulamalardır. Bunlar, e-ticaret uygulamaları gibi web uygulamalarını içerir.

✧ Gömülü kontrol sistemleri

- Bunlar, donanım cihazlarını kontrol eden ve yöneten yazılım kontrol sistemleridir. Sayısal olarak, belki de diğer bütün sistem türlerinden daha fazla gömülü sistem vardır.

Uygulama türleri



✧ Deste işleme sistemleri

- Bunlar, verileri büyük desteler halinde işlemek için tasarlanmış iş sistemleridir. Çok sayıda girdiye karşılık gelen sonuçları üretirler.

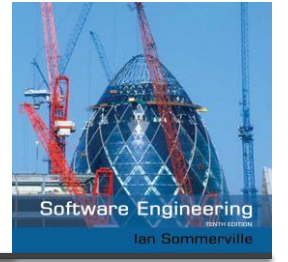
✧ Eğlence sistemleri

- Bunlar, kişisel kullanım için kullanıcıyı eğlendirmeyi amaçlayan sistemlerdir.

✧ Modelleme ve simülasyon sistemleri

- Bunlar, bilim adamları ve mühendisler tarafından fiziksel süreçleri ve durumları modellemek için geliştirilen, birçok ayrı, etkileşen nesneleri içeren sistemlerdir.

Uygulama türleri



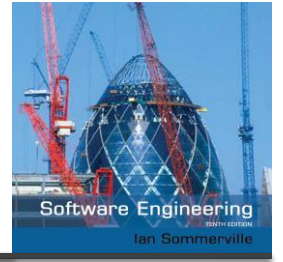
✧ Veri toplama ve analiz sistemleri

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

✧ Sistemlerin sistemleri

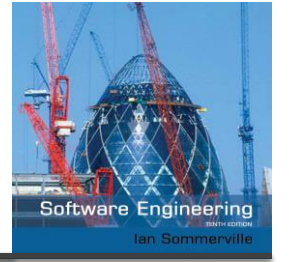
- Bunlar, bir dizi başka yazılım sisteminden oluşan sistemlerdir.

Yazılım mühendisliği temelleri



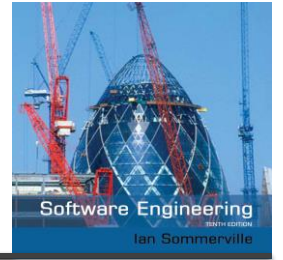
- ✧ Bazı temel ilkeler, kullanılan geliştirme tekniklerinden bağımsız olarak tüm yazılım sistemi türleri için geçerlidir:
- Sistemler, yönetilebilen ve anlaşılan bir yazılım süreci kullanılarak geliştirilmelidirler. Tabii ki, farklı yazılım türleri için farklı süreçler kullanılır.
 - Güvenilebilirlik ve performans tüm sistem türleri için önemlidir.
 - Yazılım spesifikasyonunu ve gereksinimlerini (yazılımın ne yapması gerektiğini) anlamak ve yönetmek önemlidir.
 - Uygun olduğu durumda, yeni yazılım yazmak yerine daha önceden geliştirilmiş olan yazılımı yeniden kullanmalısınız.

İnternet yazılım mühendisliği

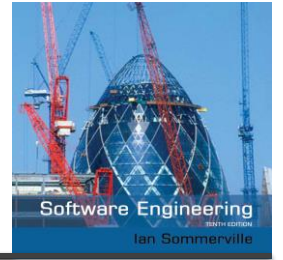


- ✧ Web artık uygulama çalıştırmak için bir platformdur ve kuruluşlar yerel sistemlerden ziyade giderek daha fazla web tabanlı sistemler geliştirmektedir.
- ✧ Web hizmetleri (Bölüm 19'da ele alınmıştır), uygulama işlevlerine web üzerinden erişilmesine izin verir.
- ✧ Bulut bilgi işlem, uygulamaların "bulut" üzerinde uzaktan çalıştığı bilgisayar hizmetlerinin sağlanmasına yönelik bir yaklaşımdır.
 - Kullanıcılar yazılımı satın almazlar, yazılımın kullanım miktarına göre ödeme yaparlar.

Web tabanlı yazılım mühendisliği



- ✧ Web tabanlı sistemler karmaşık dağıtılmış sistemlerdir, ancak daha önce tartışılan yazılım mühendisliğinin temel ilkeleri, diğer herhangi bir sistem türü için olduğu kadar bunlar için de geçerlidir.
- ✧ Yazılım mühendisliğinin temel fikirleri, diğer tür yazılımlara olduğu gibi web tabanlı yazılımlara da uygulanır.

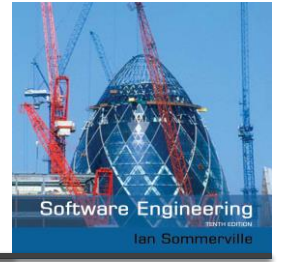


✧ Yazılımın yeniden kullanımı

- Web-tabanlı sistemleri oluşturmak için yazılımın yeniden kullanımı baskın yaklaşım haline gelmiştir. Bu sistemleri inşa ederken, çoğunlukla bir çerçevede bir araya getirilmiş olan mevcut yazılım bileşenlerini ve sistemlerini nasıl birleştireceğinizi düşünürsünüz.

✧ Kademeli ve çevik geliştirme

- Web tabanlı sistemler aşamalı olarak geliştirilmekte ve teslim edilmektedir. Bu tür sistemler için tüm gereksinimleri önceden belirlemenin pratik olmadığı artık genel olarak kabul edilmektedir.

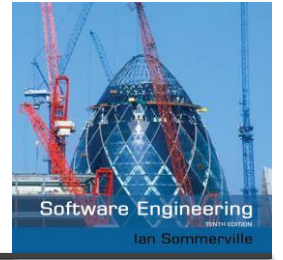


✧ Servis yönelimli sistemler

- Yazılım, yazılım bileşenlerinin tek web servisleri olduğu servis yönelimli yazılım mühendisliği kullanılarak gerçekleştirilebilir.

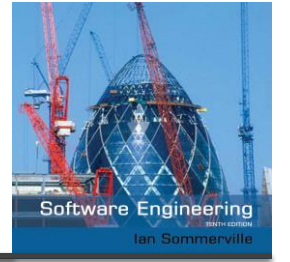
✧ Gelişmiş arayüzler

- Bir web tarayıcısındaki gelişmiş arayüzlerin oluşturulmasını destekleyen AJAX ve HTML5 gibi arayüz geliştirme teknolojileri ortaya çıkmıştır.



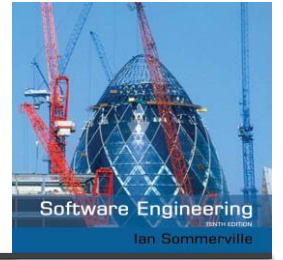
Yazılım mühendisliği etiği

Yazılım mühendisliği etiği



- ✧ Yazılım mühendisliği, sadece teknik becerilerin uygulanmasından daha geniş sorumluluklar içerir.
- ✧ Profesyonel yazılım mühendisleri, saygı görmek için dürüst ve etik açıdan sorumlu bir şekilde davranmalıdır.
- ✧ Etik davranış, yalnızca yasalara bağlı kalmaktan çok daha fazlasıdır, ahlaki açıdan doğru olan bir dizi ilkeyi izlemeyi de gerektirir.

Mesleki sorumluluk konuları

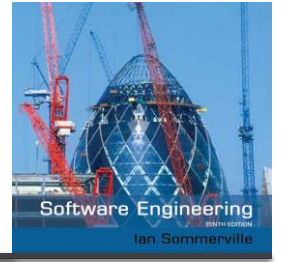


✧ Gizlilik

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

✧ Yeterlilik

- Mühendisler yeterlilik düzeylerini farklı göstermemelidirler. Kendi yeterlilikleri dışında bir işi bilerek kabul etmemelidirler.



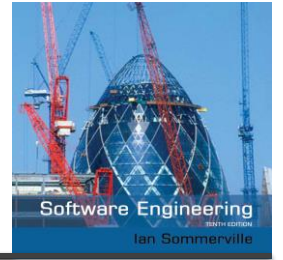
✧ Fikri mülkiyet hakları

- Mühendisler, patentler, telif hakları vb. gibi fikri mülkiyetin kullanımını yöneten yerel yasalardan haberdar olmalıdır. İşverenlerin ve müşterilerin fikri haklarının korunmasını sağlamak için dikkatli olmalıdırlar.

✧ Bilgisayarların kötü kullanımı

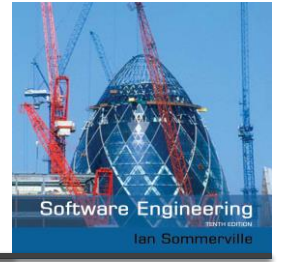
- Yazılım mühendisleri, teknik yeteneklerini başkalarının bilgisayarlarını kötüye kullanmak için kullanmamalıdır. Bilgisayarın kötüye kullanımı, oldukça basitten (örneğin, bir işverenin makinesinde oyun oynamak) çok ciddiye kadar (virüslerin yayılması) değişebilir.

ACM/IEEE Etik Kuralları



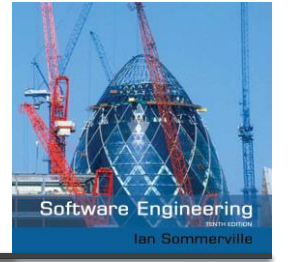
- ✧ ABD'deki profesyonel topluluklar, bir etik uygulama kodu oluşturmak için işbirliği yaptı.
- ✧ Bu kuruluşların üyeleri ve yazılım mühendisleri bu etik kurallar ve mesleki uygulamalara bağlı kalmalıdır.
- ✧ Bu kurallar, uygulayıcılar, eğitimciler, yöneticiler, süpervizörler ve kural koyucular ile mesleğin öğrencileri dahil profesyonel yazılım mühendislerinin davranışlarıyla ve aldıkları kararlarla ilgili sekiz ilke içerir.

Etik kurallarının gerekçesi



- *Bilgisayarların ticarete, endüstride, devlette, tıpta, eğitimde, eğlencede ve toplumun genelinde merkezi ve artan bir rolü vardır. Yazılım mühendisleri, yazılım sistemlerinin analiz, spesifikasyon, tasarım, geliştirilme, sertifikasyon, bakım ve testlerine doğrudan katılımla veya öğretimle katkıda bulunurlar.*
- *Yazılım sistemlerinin geliştirilmesindeki rolleri nedeniyle yazılım mühendislerinin, iyi şeyler yapmak veya zarar vermek, başkalarının iyi şeyler yapmak veya zarar vermesini mümkün kılmak veya başkalarını iyi şeyler yapmak veya zarar vermeleri için etkileme fırsatları vardır. Yazılım mühendisleri, çabalarının mümkün olduğunca iyiye kullanılmasını sağlamak için, yazılım mühendisliğini yararlı ve saygı duyulan bir meslek yapmaya sorumlu kılmalıdırlar.*

ACM/IEEE Etik Kuralları



Yazılım Mühendisliği Etik kurallar ve Profesyonel uygulamalar

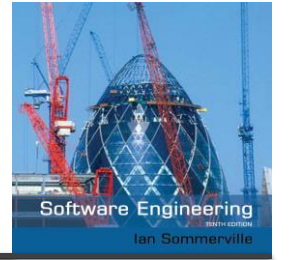
ACM/IEEE-CS Yazılım Mühendisliği Etiği ve Profesyonel Uygulamalar Ortak Çalışması

ÖN SÖZ

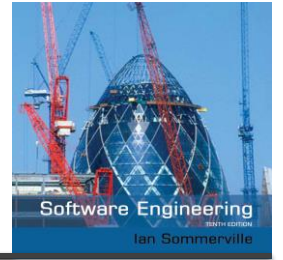
Kuralların kısa sürümü, istekleri yüksek bir soyutlama düzeyinde özetler; tam sürümde eklenen ifadeler örnekler verir ve bu isteklerin profesyonel yazılım mühendisleri olarak davranışımızı nasıl değiştirdiğini ayrıntılandırır. Bu istekler olmadan, ayrıntılar yasaya benzeyebilir ve bıkırtıcı olabilir; ayrıntılar olmadan istekler kulağa iyi ama boş gelebilir; istekler ve ayrıntılar birlikte uyumlu kurallar oluştururlar.

Yazılım mühendisleri yazılımın analiz, spesifikasyon, tasarım, geliştirme, test ve bakımını yararlı ve saygı duyulan bir meslek olarak yapma sorumluluğunu üstlenirler. Bu sorumlulukla uyumlu olarak, toplumun sağlığı, güvenliği ve refahı için yazılım mühendisleri aşağıdaki sekiz ilkeye sadık kalacaklardır:

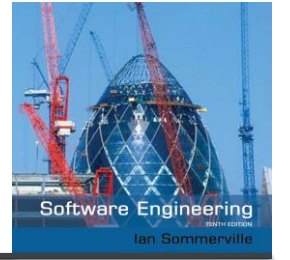
Etik ilkeler



1. TOPLUM – Yazılım mühendisleri toplumun ilgisiyle tutarlı hareket edeceklerdir.
2. MÜŞTERİ ve İŞVEREN – Yazılım mühendisleri toplumun ilgisiyle tutarlı olarak müşterilerin ve işverenin ilgisine en uygun şekilde davranacaklardır.
3. ÜRÜN – Yazılım mühendisleri ürünlerinin ve ürünlerle ilgili değişikliklerin mümkün olan en yüksek mesleki standartları karşılamasını sağlarlar.
4. DEĞERLENDİRME – Yazılım mühendisleri kendi mesleki değerlendirmelerinde bütünlüğü ve bağımsızlığı sağlayacaklardır.
5. YÖNETİM – Yazılım mühendisliği yöneticileri ve liderleri yazılım geliştirme ve bakımının yönetimine etik bir yaklaşıma uyacak ve onu daha da geliştireceklerdir.
6. MESLEK – Yazılım mühendisleri toplum ilgisiyle tutarlı olarak mesleğin bütünlüğünü ve itibarını geliştireceklerdir.
7. MESLEKTAŞLAR – Yazılım mühendisleri meslektaşlarına karşı adil ve destekleyici olacaklardır.
8. KENDİSİ – Yazılım mühendisleri mesleklerinin uygulamalarını göz önüne alarak yaşam boyu öğrenmeye katılacak ve mesleklerinin uygulanmasında etik bir yaklaşım izleyeceklerdir.

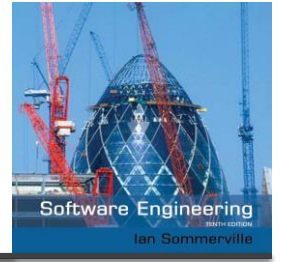


Durum çalışmaları



- ✧ Üst yönetimin politikalarıyla prensipte uyuşmazlık.
- ✧ İşverenin etik olmayan bir şekilde hareket edip, emniyet-kritik tehlike arz eden bir sistemi testler bitmeden teslim etmesi (emniyet doğrulama kayıtlarında sahtecilik yapması).
- ✧ Askeri ve nükleer sistemlerin geliştirilmesinde yer almak.

Durum çalışmaları



✧ İnsülin pompası kontrol sistemi

- Şeker hastaları tarafından kullanılan, kullanıcıya kontrollü olarak insülin dozu veren bir pompayı yöneten gömülü bir sistemdir.

✧ Ruh sağlığı vakası hasta bilgi sistemi (Mentcare)

- (ZihinSaS sistemi) Ruh sağlığı problemleri olan hastalar ve aldıkları tedaviler hakkında bilgi tutan bir tıbbi bilgi sistemidir. .

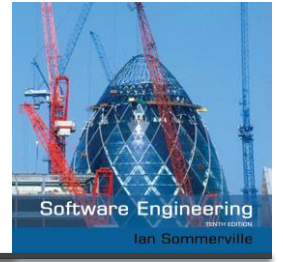
✧ Kır hava istasyonu

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

✧ Dijital bir öğrenme ortamı (iLearn)

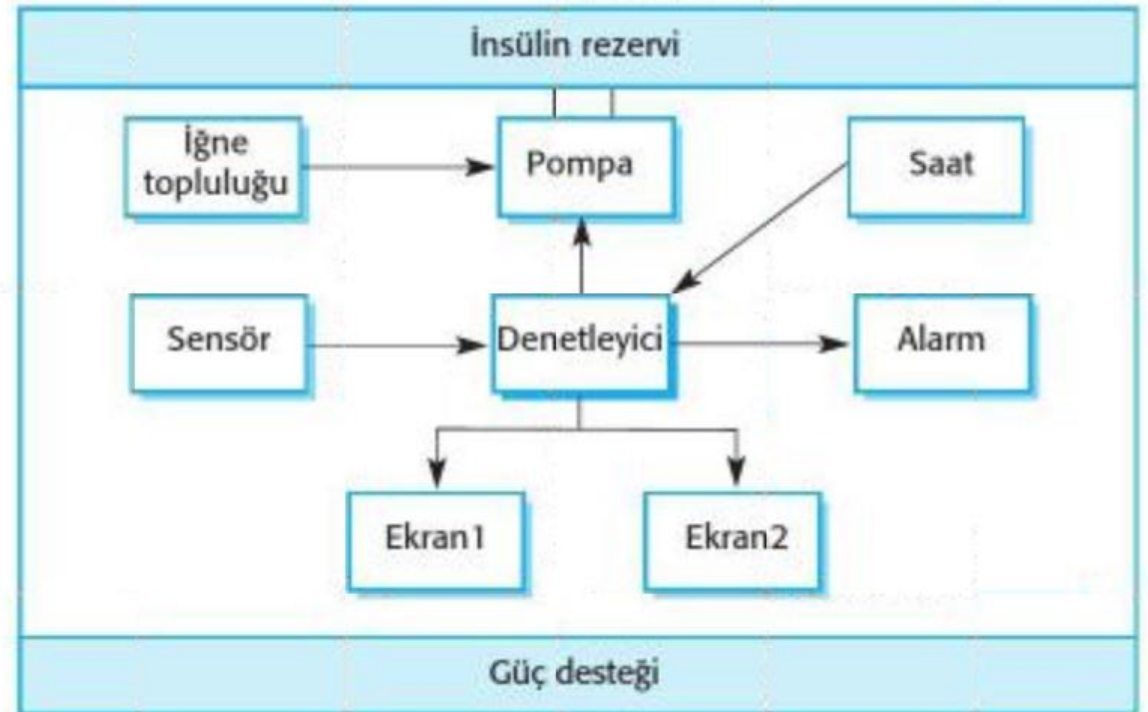
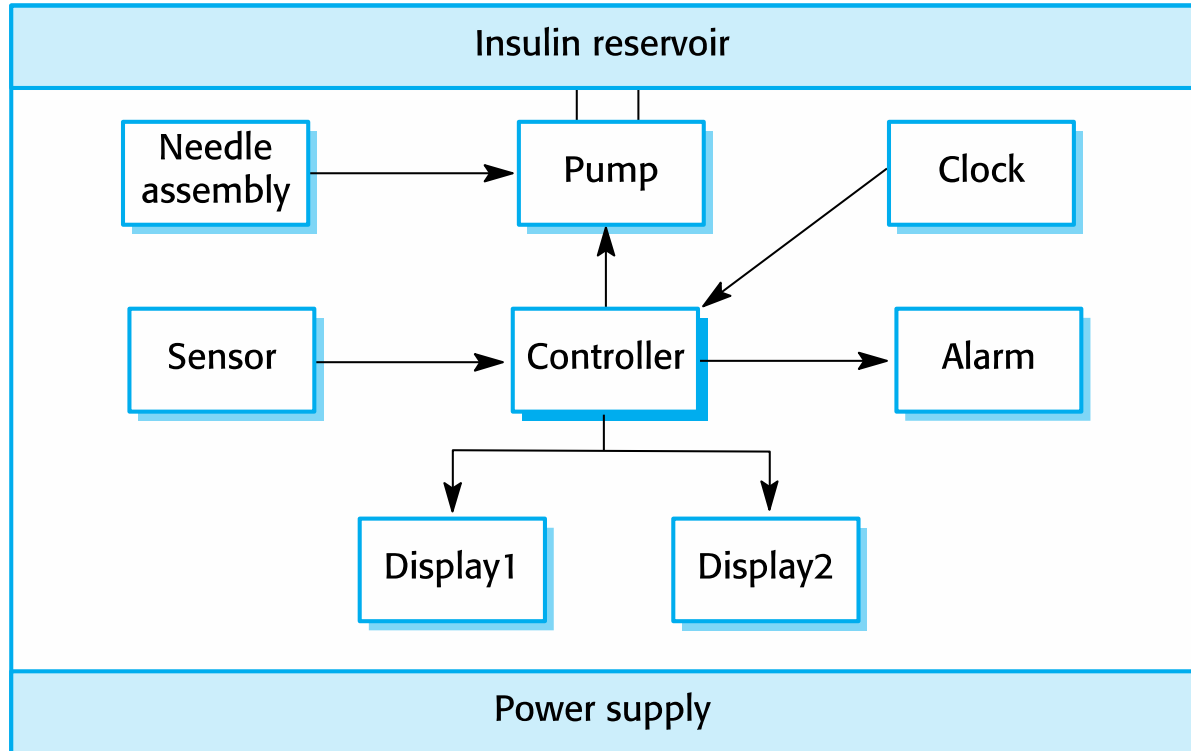
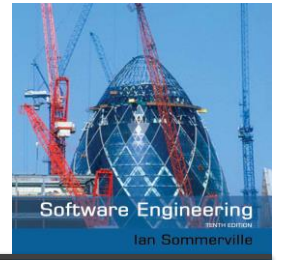
- Okullarda öğrenmeyi destekleyen bir sistem.

İnsülin pompası kontrol sistemi

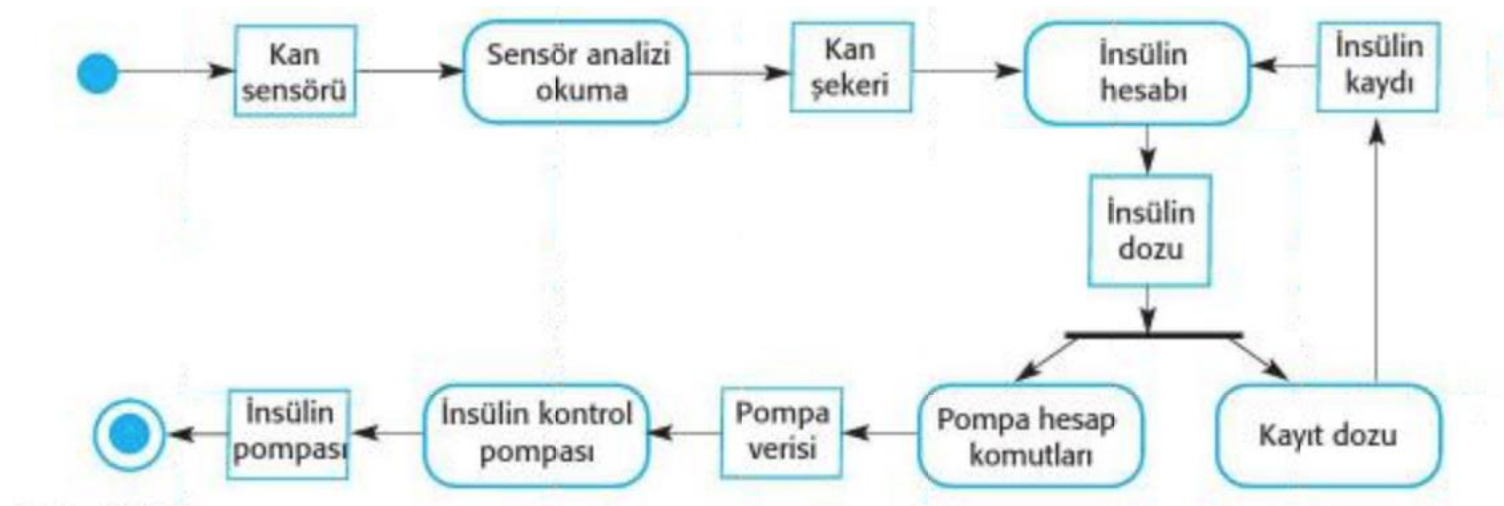
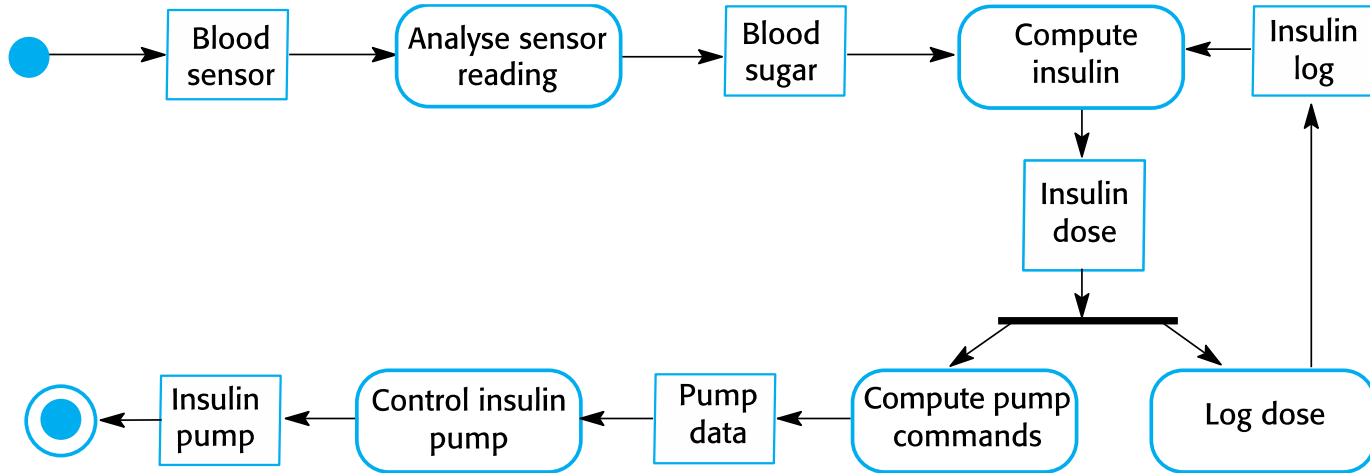
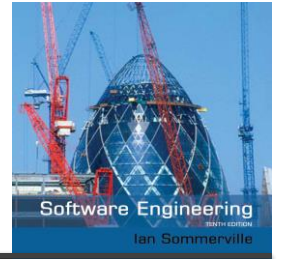


- ✧ Kan şekeri sensöründen veri toplar ve enjekte edilmesi gereken insülin miktarını hesaplar.
- ✧ Kan şekeri düzeylerinin değişim oranına dayalı hesaplama.
- ✧ Doğru insülin dozunu vermek için minyatürleştirilmiş bir pompaya sinyaller gönderir.
- ✧ Düşük kan şekeri kısa dönemde, beyin fonksiyonu bozukluklarıyla başlayarak en sonunda bilinç kaybı ve , ölümle sonuçlanabileceği için daha ciddi bir durumdur. Bununla birlikte uzun dönemde, sürekli yüksek kan şekeri, göz hasarına, böbrek hasarına ve kalp problemlerine neden olabilir.

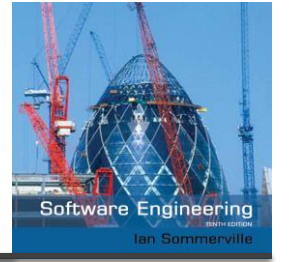
İnsülin pompası donanım mimarisi



İnsülin pompasının aktivite modeli

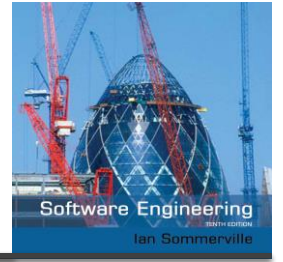


Üst düzey gereksinimler

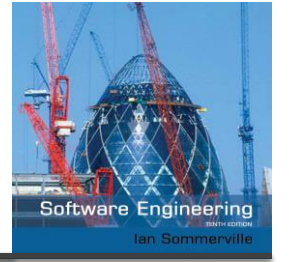


- ✧ Sistem gerektiğinde insülin göndermek için hazır olmalıdır.
- ✧ Sistem güvenilir bir şekilde çalışmalı ve mevcut kan şekeri seviyesini dengelemek için doğru miktarda insülin göndermelidir.
- ✧ Bu nedenle sistem, her zaman gereksinimleri karşılayacak şekilde tasarlanmalı ve uygulanmalıdır.

ZihinSas: Ruh sađlığını desteklemek için bir hasta bilgi sistemi

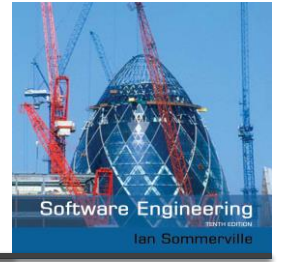


- ✧ Ruh sađlığını desteklemek için bir hasta bilgi sistemi, ruh sađlığı problemleri olan hastalar ve aldıkları tedaviler hakkında bilgi tutan bir tıbbi bilgi sistemidir.
- ✧ Çođu ruh sađlığı hastası özel hastane tedavisine ihtiyaç duymaz, fakat problemleri hakkında ayrıntılı bilgi sahibi olan bir doktorla görüşebilecekleri uzman kliniklere düzenli olarak gitmeye ihtiyaç duyarlar.
- ✧ Hastaların gelmesini kolaylaştırmak için bu klinikler sadece hastanelerde yer almaz. Ayrıca yerel sađlık kuruluşlarında veya halk merkezlerinde bulunabilirler.



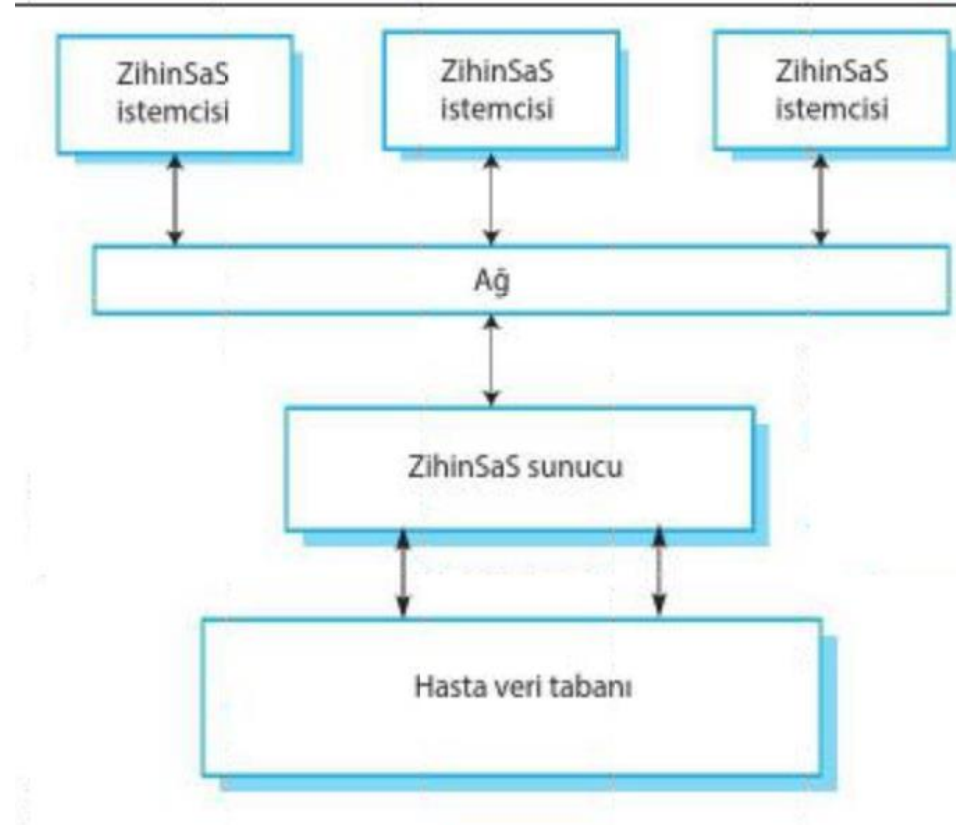
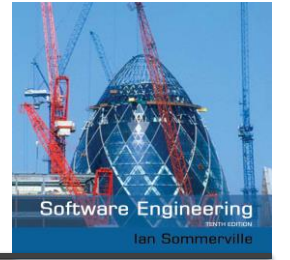
- ✧ ZihinSaS, kliniklerde kullanılmak üzere tasarlanmış bir hasta bilgi sistemidir.
- ✧ Merkezi bir hasta bilgi veri tabanı kullanılır fakat aynı zamanda güvenli ağ bağlantısı olmayan sitelerden erişilip kullanılabilmesi için bir dizüstü bilgisayarda çalışabilecek şekilde tasarlanmıştır.
- ✧ Yerel sistemler güvenli ağ bağlantısı olduğu zaman, veri tabanındaki hasta bilgilerini kullanırlar fakat bağlantı olmadığı zaman hasta kayıtlarının yerel kopyalarını kullanabilirler.

ZihinSaS'ın amaçları

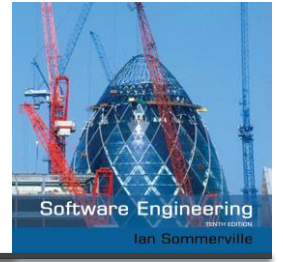


- ✧ Sağlık servis yöneticilerinin yerel ve devlet hedeflerine göre performansının değerlendirebilmesine olanak tanıyan yönetim bilgilesi oluşturmak.
- ✧ Hastaların tedavisini desteklemek için tıbbi personele zamanında veri sağlamak.

ZihinSaS sisteminin organizasyonu



ZihinSaS sisteminin temel özellikleri



✧ Bireysel bakım yönetimi

- Klinik tedavi uzmanları hastalar için kayıtlar oluşturabilir, sistemdeki bilgileri düzenleyebilir, hasta geçmişini görüntüleyebilir, vb.

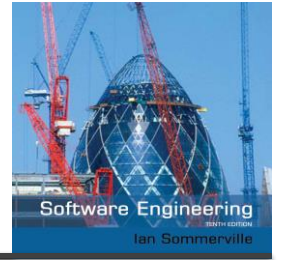
✧ Hasta izleme

- Sistem düzenli olarak tedavi edilen hasta kayıtlarını izler ve olası problemler farkedildiğinde uyarılar verir.

✧ İdari raporlama

- Sistem, her klinikte tedavi gören hasta sayısını, bakım sistemine giren ve çıkan hasta sayısını, ayrı bir bölümde tutulan hasta sayısını, reçete edilen ilaçları ve maliyetlerini vb. gösteren aylık yönetim raporları oluşturur.

ZihinSaS sistem kaygıları



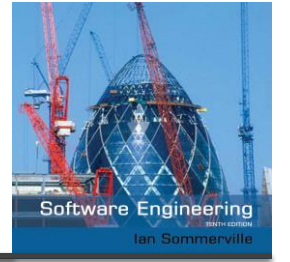
✧ Gizlilik

- Hasta bilgilerinin gizli kalması ve yetkili tıbbi personel ve hastaların dışında hiç kimseye ifşa edilmemesi esastır.

✧ Emniyet

- Bazı akıl hastalıkları, hastaların intihara meyilli olmasına veya diğer insanlar için tehlike oluşturmalarına neden olur. Mümkün olduğu durumda sistem tıbbi personeli, potansiyel olarak intihara meyilli veya tehlikeli hastalar hakkında uyarmalıdır.
- Sistem gerektiğinde kullanılabilir olmalıdır, aksi takdirde gizliliği ihlal edilebilir ve hastalara doğru ilaçları reçete etmek mümkün olmayabilir.

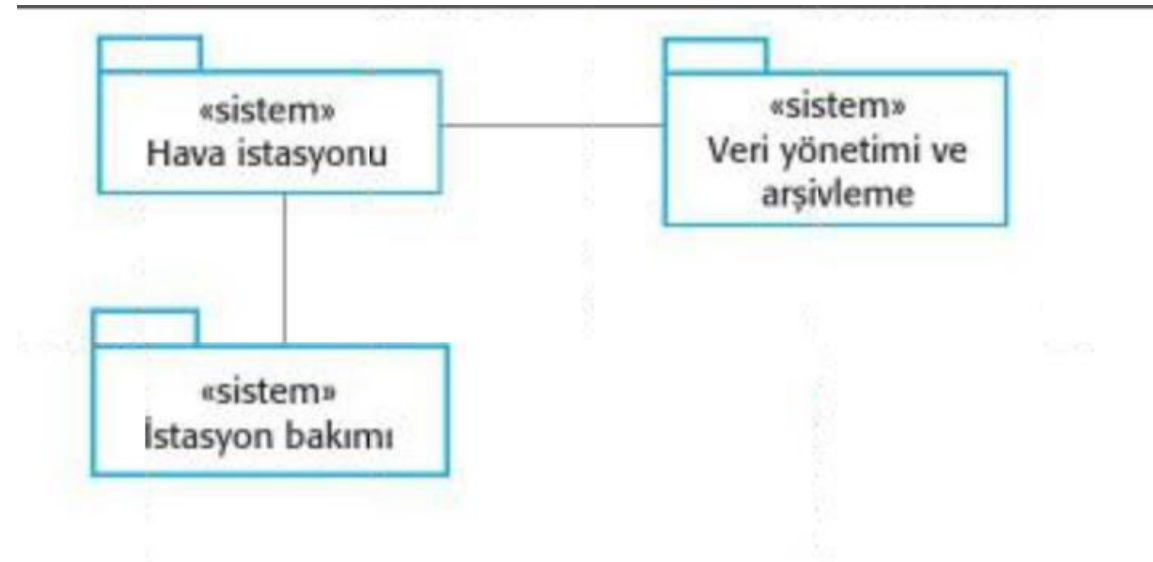
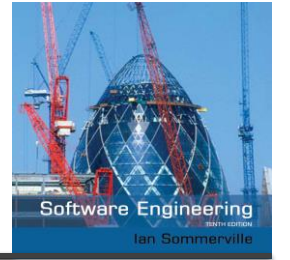
Kır hava istasyonu

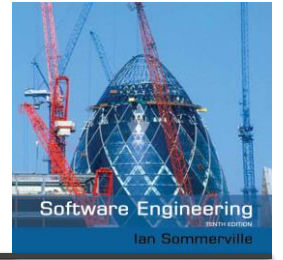


- ✧ İklim değişikliğini izlemeye yardımcı olmak ve uzak alanlardaki hava tahminlerinin doğruluğunu arttırmak için çok büyük kır alanları olan bir ülkenin hükümeti uzak alanlarda yüzlerce hava istasyonu kurmaya karar verir.
- ✧ Meteoroloji istasyonları sıcaklığı, basıncı, güneşin parlaklığını, yağış miktarını, rüzgarın hızını ve rüzgar yönünü ölçen bir dizi araçtan veri toplarlar.
 - Meteoroloji istasyonları, rüzgarın hızı ve yönü, yer ve hava sıcaklıkları, barometrik basınç ve 24 saatlik periyotta yağın yağmur miktarı gibi hava parametrelerini ölçen bir dizi araç içerir. Bu araçların her biri, parametreleri periyodik olarak okuyan ve araçlardan toplanan verileri yöneten bir yazılım sistemi tarafından kontrol edilir.



Meteoroloji istasyonunun çevresi





✧ Meteoroloji istasyonu sistemi

- Bu sistem hava verisi toplamaktan, bazı başlangıç veri işlemeyi gerçekleştirmekten ve veri yönetimi sistemine aktarmaktan sorumludur.

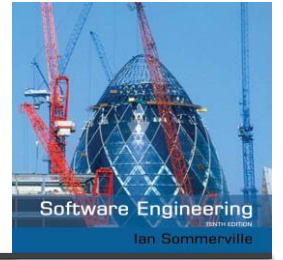
✧ Veri yönetimi ve arşivleme sistemi

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

✧ İstasyon bakım sistemi

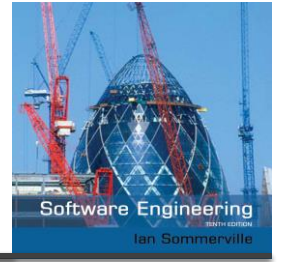
- Bu sistem uydu aracılığıyla, tüm kırsal hava istasyonları ile, bu sistemlerin sağlığını izlemek ve problemler hakkında raporlar sağlamak için haberleşirler.

Ek yazılım işlevselliği



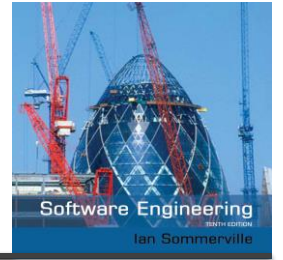
- ✧ Araçları, güç ve haberleşme donanımını izler ve kusurları yönetim sistemine raporlar.
- ✧ Sistemin gücünü yönetir, çevre koşulları izin verdiği zaman pillerin şarj edilmesini ama aynı zamanda şiddetli rüzgar gibi potansiyel olarak olumsuz hava durumlarında jeneratörlerin kapatılmasını sağlar.
- ✧ Yazılımın bölümlerinin yeni sürümleriyle değiştirilmesine ve sistem arızası durumunda yedek araçlarının sisteme alınmasına yani dinamik olarak yeniden yapılandırmayı destekler.

Okullar için Dijital öğrenme ortamı



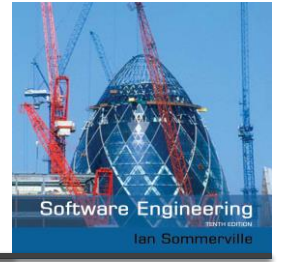
- ✧ Dijital öğrenme ortamı, öğrenme için bir dizi genel amaçlı ve özel olarak tasarlanmış araçların gömülü olabildiği ve sistemi kullanan öğrencilerin gereksinimlerine yönelik bir dizi uygulamanın bulunduğu bir çerçevedir.
- ✧ Ortamın her sürümünde yer alan araçlar, öğretmenler ve öğrenciler tarafından belirli gereksinimlere uygun olarak seçilir.
 - Bunlar hesap çizelgesi, ödev teslimi ve değerlendirme yönetmek için Sanal Öğrenme Ortamı (VLE) gibi öğrenme yönetimi uygulamaları, oyunlar ve simülasyonlar gibi genel uygulamalar olabilir.

Servis tabanlı sistemler



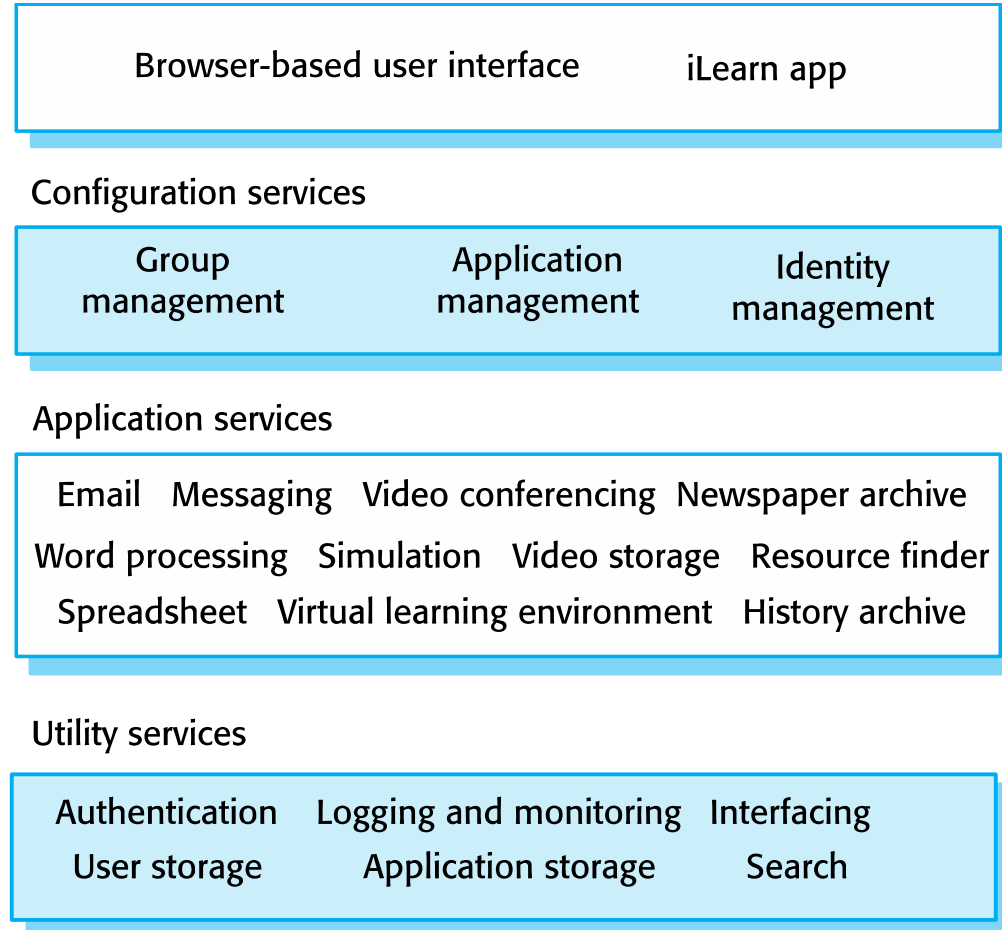
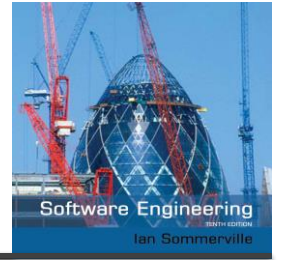
- ✧ Sistem, tüm sistem bileşenlerinin değiştirilebilir servisler olarak düşünüldüğü servis tabanlı bir sistemdir.
- ✧ Ortam, yeni servisler mevcut olduğu zaman sistemin kademeli olarak güncellenmesini sağlar.
- ✧ Ayrıca, sistemin kullanıcılarının yaşlarına uygun olarak farklı sürümlerini sağlayabilecek şekilde tasarlanmıştır.

Dijital öğrenme servisleri

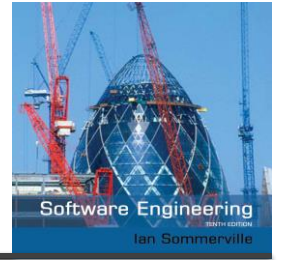


- ✧ *Yardımcı servisler, temel uygulamadan bağımsız fonksiyonellik sağlayan ve sistemdeki diğer servisler tarafından kullanılabilen servislerdir.*
- ✧ *Uygulama servisleri, e-posta, konferans, fotoğraf paylaşımı vb. gibi özel uygulamalar ve bilimsel filmler veya tarihi kaynaklar gibi özel eğitimsel içeriklerine erişim sağlayan servislerdir.*
- ✧ *Konfigürasyon servisleri, bir dizi uygulama servisiyle ortamı uyarlamak ve servislerin öğrenciler, öğretmenler ve ebeveynler arasında nasıl paylaşıldığını tanımlamak için kullanılan servislerdir.*

Dijital öğrenme ortamının mimarisi



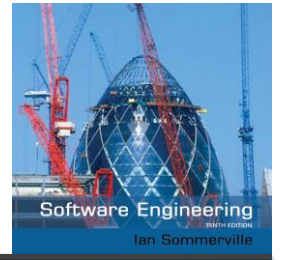
Dijital öğrenme servis entegrasyonu



- ✧ *Entegre servisler, bir API (uygulama programlama arayüzü- UPA) sunan ve diğer servisler tarafından o API aracılığıyla erişilebilen servislerdir. Bu nedenle doğrudan servisten servise iletişim mümkündür.*
- ✧ *Bağımsız servisler, basitçe bir tarayıcı arayüzü aracılığıyla erişilebilen ve diğer servislerden bağımsız olarak çalışan servislerdir. Bilgi diğer servislerle kopyala ve yapıştır gibi sadece açık kullanıcı hareketleriyle paylaşılabilir; her bağımsız servis için kimlik doğrulaması gerekebilir.*

- Yazılım mühendisliği, yazılım üretiminin tüm yönleriyle ilgili bir mühendislik disiplini-
dir.
- Yazılım sadece bir program veya programlar değildir, aynı zamanda sistemin kullanıcı-
larının, kalite güvence personelinin ve geliştiricilerin ihtiyacı olan tüm elektronik
dokümantasyonu içerir. Yazılımlarda, bakım kolaylığı, güvenilebilirlik ve güvenlik,
etkinlik ve kabul edilebilirlik bulunması şart olan özelliklerdir.
- Yazılım süreci yazılım geliştirmede yer alan tüm etkinlikleri içerir. Spesifikasyon, gelişt-
tirme, doğrulama ve evrimin bütün etkinlikleri, tüm yazılım süreçlerinin bir parçasıdır.
- Birçok farklı sistem türü vardır ve her biri geliştirilmesi için uygun yazılım mühendisliği
araçları ve tekniklerine gerek duyar. Eğer varsa çok az sayıda, özel tasarım ve gerçek-
leştirme teknikleri, tüm sistem çeşitlerine uygulanabilir.

Anahtar noktalar



- Yazılım mühendisliğinin temel fikirleri tüm yazılım sistem türlerine uygulanabilir. Bu temeller, yönetilen yazılım süreçlerini, yazılım güvenilebilirliği ve güvenliğini, gereksinim mühendisliğini ve yazılım yeniden kullanımını içerir.
- Yazılım mühendislerinin mühendislik mesleğine ve topluma karşı sorumlulukları vardır. Sadece teknik konularla ilgilenmemeli, kendi işlerini etkileyen etik konuların farkında olmalıdırlar.
- Mesleki topluluklar etik ve mesleki standartları içeren davranış kuralları yayınlarlar. Bunlar kendi üyelerinden beklenen davranış standartlarını belirlerler.