

YMÜ225

Yazılım Gereksinim Analizi

Dr. Feyza Altunbey Özbay

İçerik

- Yazılım Nedir?
- Yazılım Mühendisliği Nedir?
- Yazılım Mühendisliği Tarihçesi
- Yazılım Türleri
- Yazılım Krizi
- İyi Yazılım
- Yazılım Mühendisliği Çeşitliliği
- Yazılım Mühendisliği Etiği

Yazılım Nedir?

Yazılım

- Tanımlanmış bir işlevi yerine getiren,
- Girdi ve Çıktıları olan,
- Herhangi bir donanım üzerinde çalışan,
- Bilgisayar programı veya programlarından ve
- Kullanım ve bakım kılavuzları gibi belgelerden oluşan bir üründür.



Yazılım Mühendisliği Nedir?

- Yazılım mühendisliği, devletin, toplumun ve ulusal ve uluslararası işletmelerin ve kurumların işleyişi için gereklidir.

IEEE Bilgisayar Topluluğunun Yazılım Mühendisliği Tanımı

“Mühendislik eylemlerinin, (Geliştirme, İşletme, ve Bakım), disiplinli, sistematik ve nicelikli bir şekilde yazılıma uygulanması”

Yazılım Mühendisliğinin Tarihçesi

- Yazılım mühendisliği kavramı ilk olarak 1968'de, o zaman yazılım krizi (Naur ve Randell 1969) olarak adlandırılan konuları tartışmak için düzenlenen bir konferansta önerildi. Program geliştirmeye yönelik birbirinden farklı yaklaşımların büyük ve karmaşık yazılım sistemlerinin geliştirilmesinde ölçeklenmediği ortaya çıktı. Yazılım sistemleri güvenilir olmamakta, beklenenden daha fazla maliyetle geç teslim edilmekteydi.
- 1970'ler ve 1980'ler boyunca, yapısal programlama, bilgi saklama ve nesne yönelik geliştirme gibi çeşitli yeni yazılım mühendisliği teknikleri ve yöntemleri geliştirilmiştir.
- Günümüzde yazılım mühendisliğinin temeli olan standart gösterimler geliştirilmiştir.

Yazılım Mühendisliği – Anahtar Kelimeler

- Yazılım mühendisliği, sistem spesifikasyonunun ilk aşamalarından sistemin kullanıma girdikten sonra bakımına kadar yazılım üretiminin **tüm yönleriyle** ilgilenen bir **mühendislik disiplini**dir. Bu tanımda iki anahtar ifade vardır:

-Mühendislik disiplini: Mühendisler çalışan şeyler yaparlar. Uygun olan yerlerde teorileri, yöntemleri ve araçları uygularlar. Ancak, bunları seçici bir şekilde kullanırlar ve uygulanabilir teoriler ve yöntemler olmasa bile her zaman sorunlara çözüm bulmaya çalışırlar. Mühendisler ayrıca kurumsal ve mali kısıtlamalar dahilinde çalışmalarını gerektiğini ve bu kısıtlamalar dahilinde çözümler aramaları gerektiğini bilirler.

Yazılım Mühendisliği

-Yazılım üretiminin tüm yönleri:

Yazılım mühendisliği sadece yazılım geliştirmenin teknik süreçleriyle ilgilenmez. Ayrıca yazılım proje yönetimi ve yazılım geliştirmeyi desteklemek için araçların, yöntemlerin ve teorilerin geliştirilmesi gibi faaliyetleri de içerir.

Yazılım Mühendisliği Neden Önemlidir?

- Kişiler ve toplumlar giderek daha fazla gelişmiş yazılım sistemlerine güvenmektedirler. **Bu nedenle emniyetli ve güvenilir sistemleri ekonomik ve hızlı olarak üretebilmeliyiz.**
- Uzun vadede, profesyonel yazılım sistemleri için yazılım mühendisliği yöntemlerini ve tekniklerini kullanmak, kişisel bir programlama projesiymiş gibi sadece programlar yazmaktan daha ucuzdur. **Bu nedenle, yazılım mühendisliği yöntemlerini kullanmamak sınama, kalite güvence ve uzun dönemli bakım için daha yüksek maliyetlere neden olur.**

Yazılım Mühendisliği

En Çok 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, güvenilir 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 dahil olmak üzere bilgisayar tabanlı sistem geliştirmenin tüm yönleriyle ilgilenir. Yazılım mühendisliği bu daha genel sürecin parçasıdır.
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 (app) endüstrisi yaratılmasını desteklemiştir.

Yazılım Türleri

- Yazılım mühendisleri bir müşteriye satabilecek yazılım ürünlerini geliştirmekle uğraşırlar. İki çeşit yazılım ürünü vardır:
 - Genel yazılımlar: Bunlar, bir geliştirme kuruluşu tarafından üretilen ve bunları satın alabilecek herhangi bir müşteriye açık pazarda satılan bağımsız sistemlerdir. Bu tür ürün örnekleri arasında mobil cihazlar için uygulamalar, kişisel bilgisayarlar için veri tabanları, kelime işlemciler, çizim paketleri ve proje yönetimi araçları bulunur.
 - Özelleştirilmiş (ısmarlama) yazılımlar:

Yazılım Türleri

-Özelleştirilmiş (ısmarlama) yazılımlar: Bunlar, belirli bir müşteri tarafından ısmarlanmış ve onlar için geliştirilen sistemlerdir. Bir yazılım firması, yazılımı özellikle o müşteri için özel olarak tasarlar ve gerçekleştirir.

Bu tür yazılımların örnekleri arasında elektronik cihazlar için kontrol sistemleri, belirli bir iş sürecini desteklemek için yazılan sistemler ve hava trafik kontrol sistemleri bulunur.

Yazılım Türleri

Yazılım türlerinin çeşitliliğinden ötürü evrensel yazılım mühendisliği yöntemleri veya teknikleri yoktur. Fakat birçok yazılım türünü etkileyen dört etken vardır.

- Heterojenlik: Sistemlerin, gittikçe artarak, farklı tür bilgisayarlar ve taşınabilir cihazlar içeren ağlar arasında dağıtık sistemler olarak çalışmalıdır.
- İş ve Sosyal Değişim: Var olan yazılımlar değiştirilebilir ve hızlıca yeni yazılımlar geliştirilebilir olmalıdır.
- Güvenlik ve Güven: Yazılım tüm yönleri ile yaşantımıza dahil olduğu için yazılımlara güvenilmesi şarttır.
- Ölçek: Taşınabilir cihazdan → bulut tabanlı sistemlere kadar geniş bir ölçek.

Yazılım Krizi

- 1960'lardan itibaren yazılım ürünlerine artan talepler karşısında otomasyonu da içeren değişik çözümler uygulandıysa da yeterli üretim kapasitesine erişilemedi.
- Bu yetersizlik, “yazılım krizi” söylemiyle dünya literatüründe yerini aldı.
- Yazılım krizinin temel sebebi, üstel olarak artan talebe karşılık doğrusal hızla artış gösteren üretim kapasitesidir.
- Diğer yandan yazılım projelerindeki başarısızlık oranı da şaşırtıcı derecede yüksek olup bu krizi beslemektedir.

Yazılım Projelerinin Başarısızlık Nedenleri

-Artan sistem karmaşıklığı

Yeni yazılım mühendisliği teknikleri, daha büyük, daha karmaşık sistemler oluşturmamıza yardımcı olurken, talepler de değişir. Sistemler daha hızlı inşa edilmeli ve teslim edilmelidir; daha büyük, daha karmaşık sistemler gereklidir; ve sistemlerin önceden imkansız olduğu düşünülen yeni yeteneklere sahip olması gerekir.

Daha karmaşık yazılımlar sunmanın getirdiği yeni zorlukların üstesinden gelmek için yeni yazılım mühendisliği teknikleri geliştirilmelidir.

Yazılım Projelerinin Başarısızlık Nedenleri

-Yazılım Mühendisliği Yöntemlerini Kullanmadaki Başarısızlık

Yazılım mühendisliği yöntem ve tekniklerini kullanmadan bilgisayar programları yazmak oldukça kolaydır.

Birçok şirket, ürünleri ve hizmetleri geliştikçe yazılım geliştirmeye yöneldi. Günlük işlerinde yazılım mühendisliği yöntemlerini kullanmazlar. Sonuç olarak, yazılımları genellikle olması gerekenden daha pahalıdır ve daha az güvenilirdir.

Bu sorunu çözmek için daha iyi yazılım mühendisliği öğretimine ve eğitime ihtiyacımız var.

İyi Yazılım ?

Ürün özelliği	Tanım
Kabul edilebilirlik	Yazılım tasarlandığı kullanıcı tipi tarafından kabul edilebilir olmalıdır. Bunun anlamı, anlaşılabilir, kullanılabilir ve kullandıkları diğer sistemlerle uyumlu olması gerektiğidir.
Güvenilebilirlik ve güvenlik	Yazılım güvenilebilirliği, güvenilirlik, güvenlik ve emniyet dahil 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 Süreçleri

Yazılım mühendisliğinde kullanılan sistematik yaklaşıma **yazılım süreci** denir. Bir yazılım süreci, bir yazılım ürününün üretilmesine yol açan bir faaliyetler dizisidir. Bunlar;

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

Bilgisayar Bilimleri – Sistem Mühendisliği

Bilgisayar bilimleri, bilgisayarların ve yazılım sistemlerinin altında yatan teori ve yöntemlerle ilgilenirken, yazılım mühendisliği, yazılım üretmenin pratik problemleriyle ilgilenir. Elektrik mühendisleri için bir miktar fizik bilgisi gerekli olduğu gibi, yazılım mühendisleri için de bir miktar bilgisayar bilimi bilgisi gereklidir.

Sistem mühendisliği, yazılımın önemli bir rol oynadığı karmaşık sistemlerin gelişimi ve evriminin tüm yönleriyle ilgilenir. Sistem mühendisleri, sistemin belirlenmesine, genel mimarisinin tanımlanmasına ve ardından bitmiş sistemi oluşturmak için farklı parçaların entegre edilmesine dahil olur.

Yazılım Mühendisliği Çeşitliliği

- **Bağımsız uygulamalar:** Kişisel bilgisayarlardaki ofis uygulamaları, bilgisayar destekli tasarım programları, seyahat uygulamaları, fotoğraf düzenleme yazılımı vb.
- **Etkileşimli işlem-tabanlı uygulamalar:** Uzaktaki bir bilgisayarda çalışan ve kullanıcıların kendi bilgisayarlarından, tabletlerinden veya telefonlarından erişilebilen uygulamalardır. E-ticaret gibi.
- **Gömülü kontrol sistemleri:** Donanım cihazlarını kontrol eden ve yöneten yazılım kontrol sistemleridir. Cep telefonlarındaki yazılımlar, arabada savrulmayı engelleyen fren kontrol sistemini kontrol eden yazılım vb.
- **Deste işleme sistemleri:** Verdileri desteler halinde işlemek için tasarlanmış iş sistemleridir. Telefon faturalama sistemleri, maaş ödeme sistemleri vb.

Yazılım Mühendisliği Çeşitliliği

- **Eğlence sistemleri:** Kişisel kullanıma yönelik ve kullanıcıyı eğlendirmeye yönelik sistemlerdir.
- **Modelleme ve simülasyon için sistemler:** Bilim adamları ve mühendisler tarafından fiziksel süreç ve durumları takip etmek için geliştirilmiştir.
- **Veri toplama ve analiz sistemleri:** Kendi çevrelerinden veri toplayan ve o veriyi işlenmesi için başka sistemlere gönderen sistemlerdir.
- **Sistemlerin sistemi:** Ticari kurumlarda ve büyük organizasyonlarda kullanılan ve çok sayıda başka sistemden oluşan sistemlerdir.

Yazılım Mühendisliği Temelleri

- Yazılımların farklı özellikleri olduğu için her sistem özelleşmiş yazılım mühendisliği teknikleri gerektirir. Yine de tüm yazılım sistemlerine uygulanacak bir takım **Yazılım Mühendisliği Temelleri** vardır.
 - Yönetilebilen ve anlaşılmış bir yazılım süreci kullanarak geliştirilmelidirler.
 - Güvenilebilirlik ve performans tüm sistemler için geçerlidir.
 - Yazılım spesifikasyonunu ve gereksinimlerini anlamak ve yönetmek önemlidir.
 - Var olan kaynaklar etkin bir biçimde kullanılmalıdır.

Yazılım Mühendisliği Etiği

Diğer mühendislik disiplinleri gibi, yazılım mühendisliği, bu alanda çalışanların özgürlüğünü sınırlayan sosyal ve yasal bir çerçevede yürütülür.

- **Gizlilik:** İşveren ve müşterilerin gizliliğine saygı duyulmalıdır.
- **Yeterlilik:** Yetkin olmadığınız konuların dışındaki işleri kabul etmemelisiniz.
- **Fikri mülkiyet hakları:** Telif hakkı ve patent gibi yerel yasalardan haberdar olmalısınız.
- **Bilgisayarın kötü kullanımı:** Teknik yeteneklerinizi başkalarının bilgisayarlarını suistimal etmek için kullanmamalısınız.

Yazılım Mühendisliği Etiği

- Profesyonel topluluklar veya kuruluşlar etik standartların oluşturulmasında önemli rol oynarlar.
 - ACM, IEEE (Institute of Electrical and Electronic Engineers – Elektrik ve Elektronik Mühendisleri Enstitüsü) ve İngiliz Bilgisayar Topluluğu gibi organizasyonlar mesleki davranış ve etik kuralları yayınlarlar.

Yazılım Mühendisliği Etik kurallar ve Profesyonel uygulamalar

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

ÖNSÖZ

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 bıktırıcı 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:

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.