

KAYNAK KİTAPLAR

- Software Engineering: A Practitioner's Approach / Roger S. **Pressman**.
McGraw/Hill, 2014, 8th ed.
- Applying UML and Patterns: Intro. OOAD & Iterative Development / Craig
Larman. Prentice-Hall, 2004, 3rd ed.
- Software Engineering / Ian Sommerville.
Pearson, 2015, 10th ed.

DERS İÇERİĞİ

- • Yazılım Mühendisliğine Giriş (Pressman)
- • Yazılım Geliştirmede Süreç Modelleri (Pressman)
- • Gereksinim Mühendisliği (Pressman)
- • Nesneye Yönelik Çözümleme (Larman)
- • Nesneye Yönelik Tasarım (Larman)
- • Yazılım Sınama Teknikleri (Pressman)
- • Yazılım Proje Yönetimine Giriş (Pressman)
- • Yazılım Ölçütleri (Pressman)

Yazılım :

- Herhangi bir boyuttaki herhangi bir tür donanımda çalışan bilgisayar programını VE,
- Basılı veya elektronik ortamdaki her tür dokümanı içeren ürün.
- Dokümanlar yazılım geliştirme ve son kullanıcıya yönelik olabilir.
- Yazılım bir üründür, ancak başka ürünler geliştirmeye veya elde etmeye yarayan bir araç da olabilir.
- Yazılım fiziksel bir ürün olmadığı için aşınmaz, ancak zamanla yetersizleşebilir.
- Değişim kaçınılmazdır: Yazılım, yaşam döngüsü süresince değişikliklere uğrar.
- Değişiklikler, yazılımda yeni hatalar oluşturabilir.
- Yeni hatalar tam olarak düzeltilmeden yeni değişiklikler gerekebilir.
- Yaşam döngüsü (software Life Cycle): Yazılımın bir fikir olarak doğmasından, kullanım dışı bırakılmasına kadar geçen süreç.

YAZILIM TÜRLERİ

- Sistem Yazılımı :
 - Diğer programlara hizmet sunmak üzere hazırlanmış programlar.
 - Derleyiciler, işletim sistemleri, vb.
- Mühendislik Yazılımı / Bilimsel Yazılım :
 - Mühendislik ve bilimsel hesaplamalarda kullanılmak üzere hazırlanmış programlar.
 - Büyük hacimli verilerle uğraşır.
 - “Numara öğütmek / Number crunching”.
- Gömülü (Embedded) Yazılım :
 - Donanım ile çok sıkı ilişkidir.
 - Denetim amaçlıdır.
 - Gerçek zamanlı uygulamalardır.

Yazılım Türleri

Uygulama Yazılımı :

- • **Product-line**, shrink-wrapped, (commercial) off-the-shelf (COTs), vb.
- • Bkz. TS/BS ISO/IEC 25051 COTS Yazılım Ürünleri standardı
- • Bir çok mühendislik alanında olduğu gibi Yazılım Mühendisliği alanında da tanımlanmış standartlar vardır.
- • Erişim için kütüphaneye başvurunuz.
- • Ciddi bilgilere erişim için kütüphaneler kullanılmalıdır.
- • Farklı müşteriler tarafından kullanılacak genel amaçlı yazılımlar
- • Cari hesap uygulamaları, çeşitli otomasyon programları, kelime işlem uygulamaları, vb.
- Kurumsal Yazılım:
 - • Belirli ticari iş gereksinimlerine yönelik programlar.
 - • İş süreçleri ile ilgili bilgiye sahip olmalıdır.
 - • Genellikle müşteriye özel tasarlanır.
 - • Veri dönüştürme ve değerlendirme uygulamaları, iş süreçlerinin kimi zaman gerçek zamanlı izlenilmesi, vb.
 - • Zamanla "eski yazılım" haline dönüşür!

ESKİ YAZILIM (Legacy Software):

- İş sürecinin önemli bir parçası olan ve çok uzun süredir kullanılan yazılımlar.
 - Eski yazılımda bulunabilecek olumsuzluklar:
 - Eksik veya hatalı dokümantasyon
 - Zamanla karmaşıklaşmış kod
 - Esnek olmayan yapı
 - Eski donanımla çok sıkı ilişki
 - Yazılım mühendisliğindeki gelişmelerden yoksunluk nedeniyle düşük kalite.
- Eski yazılımın değiştirilmesini gerektiren nedenler :
 - İş alanındaki yeni gereksinimler
 - Güncel sistemlerle birlikte çalışabilmesi için uyumluluk kazandırılması
 - Donanımın ömrünün dolması nedeniyle daha güncel ortama taşınma gerekliliği

YAZILIM HAKKINDAKİ YANILGILAR: MÜŞTERİ AÇISINDAN

- • Programın yazılmasına başlanması için amaçları genel olarak belirlemek yeter, ayrıntılar sonra kararlaştırılabilir. Nasıl olsa yazılım esnektir. • Belirsiz gereksinimler, çürük atılmış temele benzer.
- • Yazılım esnektir. Değişen gereksinimler kolayca sisteme uyarlanabilir.
- • Yazılım yaşam döngüsünde ilerledikçe, değişen gereksinimleri yazılıma uyarlamanın bedeli üstel olarak artar.

YAZILIM HAKKINDAKİ YANILGILAR: İDARİ

- • İşler yetişmiyorsa takıma yeni programcılar ekleriz.
- • Yazılım hayat döngüsü içerisinde ilerledikçe, yeni elemanların yazılıma hakim olması üstel olarak zorlaşır. İşler daha da gecikir.
- • Geliştirmesini üstlendiğim yazılımı tamamen veya kısmen fason yaptırırım.
- • Proje ilerlemesini kendi içinde denetleyemeyen bir firma, dışarıya verdiği işi izlemekte de zorlanacaktır.
- • Açık kaynak yazılım üretirsem kar edemem.
- • Danışmanlık hizmetleri ile kar edilebilir.
- • Başka iş modelleri de vardır.

YAZILIM HAKKINDAKİ YANILGILAR: PROGRAMCI AÇISINDAN

- • Yazılımı tamamlayıp müşteriye teslim edince işimiz biter. Yazılım üstünde harcanan çabanın yarısından fazlası, yazılımın müşteriye ilk teslimatından sonra harcanmaktadır.
- • Yazılımı tamamlamadan kalitesini ölçemem. Kalite güvence yöntemleri yazılım hayat döngüsünün her aşamasında uygulanabilir. Çözümleme sürecinde dahi kullanılabilecek kalite ölçütleri bulunmaktadır.
- • Yazılım eşittir program. Gereksinim analizi başlı başına bir emektir.
- • Dokümantasyon ve sınaama çalışmalarını da unutmayın!
- • Bazı durumlarda entegrasyon çalışmaları da gerekmektedir.
- • Yazılım mühendisliğinin gereklerini uygulayarak boşuna çaba harcıyoruz.
- • Haritası olmayan yolunu kaybeder.
- • Kalite için harcanan çaba, karşılığını yazılım hayat döngüsünün ilerleyen aşamalarında fazlasıyla ödeyecektir.
- • Küresel ölçekte yazılım projelerinin %50'si başarısızlığa uğramaktadır.

YAZILIM MÜHENDİSLİĞİ

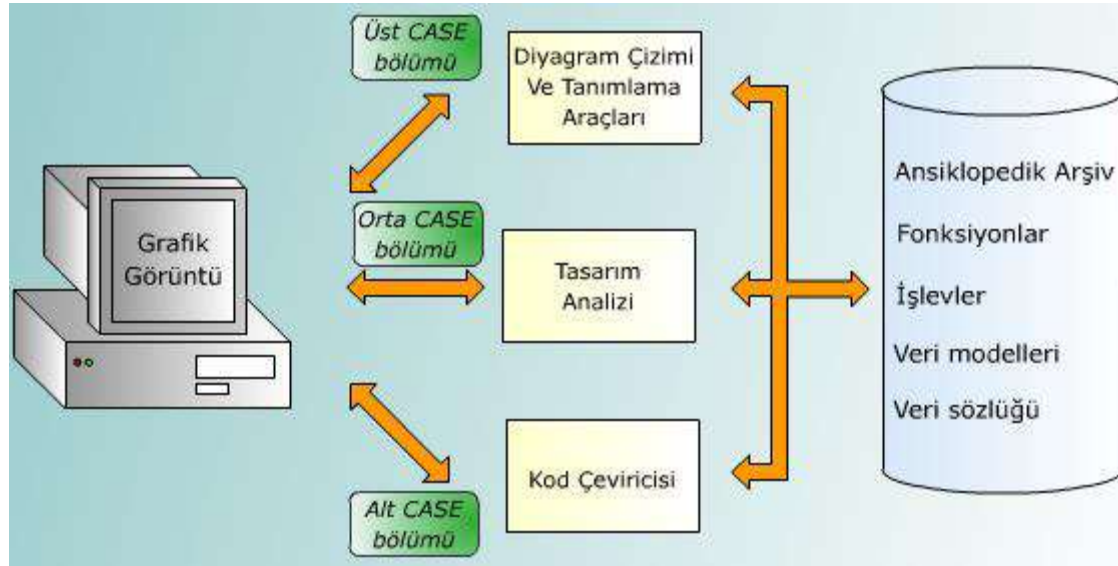
- MÜHENDİS ? Hendese ?
- ÖLÇME
- TASARIM
- Yazılım Mühendisliği
- Kaliteli Yazılım
- Ekonomik yolla
- Mühendislik Prensipleri

YAZILIM GELİŞTİRME SÜRECİ TEMEL ADIMLARI

- GEREKSİNİM ANALİZ
- Tasarım
- Gerçekleme
- Test

YAZILIM GELİŞTİRME SÜRECİ DESTEK AKTİVİTELERİ

- Proje Planlama
- Kalite Yönetimi
- Yazılım Ölçümü
- Risk Analizi
- Yazılım Konfigürasyon Yönetimi



CASE SİSTEMİ

- CASE (Computer Aided Software Engineering); yazılım mühendisliği görevlerinin, bilgisayar yardımı ile otomatik olarak (!) gerçekleştirilmesidir.
- CASE ortamı, belirli bir sorunun çözümü ya da bir kurumun yönetim bilişim sistemi için gerekli yazılımı otomatik olarak geliştiren bir bilgisayar sistemidir.
- Bu sistem; otomatik yazılım araçlarının ve yapısal yöntemlerin bir kombinasyonu olup, yazılım geliştirme sürecini adım adım izleyerek gerçekleştirmekte ve yazılımı üretmektedir.
- Avrupa ülkelerinde "bütünleşik programlama desteği ortamı" anlamında IPSE (Integrated Programming Support Environments) adı da verilir

CASE sisteminin 3 ögesi:

- İşlemler: Yazılımın geliştirme sürecini düzenlemekte kullanılan yöntem bilimi
- Yöntemler: Projenin gerektirdiği standart tasarım teknikleri ve yöntemleri
- Bütünleşik otomatik yazılım geliştirme araçları
- İşlemler ve yöntemlere ait bilgiler bir ansiklopedik arşivde saklanmaktadır.
- Bu bilgileri kullanarak, otomatik araçlar, diyagram çizimi ve betimleme, tasarım analizi, kodlama işlemlerini gerçekleştirmektedirler.
- İşlemlerin yürütülmesinde etkileşim ve ekrandan yararlanılmaktadır.
- Her basamakta elde edilen sonuçlar, sırası geldiğinde kullanılmak üzere ansiklopedik arşivde saklanmaktadır.

4. Kuşak diller (4th Generation Language: 4GL) olarak tanımlanan diller kullanılarak, grafik arayüzler üzerinden yazılımın kaynak kodunun değiştirilebilmesini sağlayan gereçlerin (4GT) kullanıldığı tekniklerdir.

- 4GL diller doğal dile yakın dillerdir. 4GT gereçler komutları yerine getirebilecek

bir gösterim üzerinden (şemalar, akış diyagramları gibi) kaynak kod üretebilmektedir.

- Şema ile kod arasında çift yönlü dönüşüm söz konusudur.
- Programlama bilmesi gerekmeyen konu uzmanlarının ve üst yöneticilerin kolaylıkla yazılıma müdahale edebilmesi avantajını vaat etmektedir.
- Yine de algoritmik düşünme yeteneği olmadan kullanılamazlar.
- Bu araçlar günümüzde sadece bazı özel alanlarda ve sınırlı olarak

SÜREÇ SERTİFİKASYONU

- Olgunlaşmış bir yazılım geliştirme sürecine sahip olmayan bir yazılım firması, projelerini başarı ile sonuçlandıramaz.
- Bir yazılım firması, süreçlerinin yeterliliğini bağımsız kurumlara onaylatmayı seçebilir.
- Gerekli olduğu durumlar:
- Bazı büyük müşteriler sertifikalı yazılım firmaları ile çalışmayı şart koşarlar.
- Gereksiz olduğu durumlar:
- Çok küçük şirketler ve/veya projeler için ek yük olarak görülebilir.
- Güncel model ve standartlar:
- CMMI: Capability Maturity Model Integration
- SEI tarafından önerilmiştir (Software Engineering Institute of Carnegie-Mellon University) (kurumsal)
- PMI: Genel amaçlı bir proje yönetimi yaklaşımı (bireysel ve kurumsal)
- ISO 9001:2015 standartları (Genel) (kurumsal)
- ISO/IEC 90003:2018 (Yazılım geliştirmeye özel) (kurumsal)

CMMI DÜZEYLERİ

- CMMI düzeyleri:
- 0. Düzey: Yürütülmeyen (Level 0: **Incomplete**).
- 1. Düzey: Giriş düzeyi (Level 1: **Initial/Performed**). İş şansa ve anahtar kişilere kalmış. Etkinliklerin tümü yürütülüyor ama bunların tanımlanması için hiçbir sistematik girişim yoktur, her seferinde farklılaşabilirler.
- 2. Düzey: Yönetilen/Yinelenebilir (**Managed/Repeatable**). Temel planlama ve izleme yöntemleri kullanılarak, önceki projelerdeki başarılar yeni projelerde tekrarlanabilir.
- 3. Düzey: Tanımlanmış (**Defined**). Kişi ve risk yönetimi ile projenin yönetimi iyileştirilir. Tüm organizasyonda standart süreçler tanımlanmıştır
- Büyük müşteriler en az bu düzey yazılım evleri ile çalışmak ister.
- 4. Düzey: Nicel Yönetilen (Quantitatively **Managed**). Süreç ve yazılım ölçütleri kullanılarak kalite yönetimine geçilir. İlerleme sürekli izlenir, bütçe ve zaman hedeflerinden sapmalar erkenden belirlenerek gerekli önlemler alınır.
- 5. Düzey: İyileştirilmiş (**Optimized**). Süreç yönetimi geçmiş deneyimlerin ışığında sürekli iyileştirilir.

YAZILIM GELİŞTİRME SÜREÇLERİ

CMMI DÜZEYLERİ

- CMMI, her düzeyde belli süreç alanlarının kapsanıyor olmasını ister.
- Süreç alanları belli hedeflere ulaşmak için beklenen uygulamalardır.
- Her firma gerekli süreç alanlarını kendine özgü süreçlerle kapsar.
- CMMI ilgi alanları:
 - CMMI-DEV (Development): Yazılım geliştirme
 - CMMI-SVC (Service): Hizmet sunumu ve yönetimi
 - CMMI-ACQ (Acquisition): Ürün ve hizmet alımı
- Bu üç alan, CMMI v2.0 ile (2018) tek bir olgunluk modeli altında birleştirilmiştir.
- Tarihçe:
 - 1987-1997: CMM
 - 1998-2009: CMMI
 - 2010: CMMI v1.3, çevik süreçler desteklenmeye başladı
 - 2018: CMMI v2.0
- Continuous (kısmi, sadece bazı süreç etkinliklerini kapsayan) ve Staged (bütüncül, daha değerli) ayırımı kaldırılıp tüm süreç etkinliklerini kapsayacak içerikte birleştirildi.
- Kaynakları ücretli/abonelik tabanlı hale geldi.

CMMI DÜZEYLERİ

• CMMI Level 3+ sertifikası almış kamu ve özel kurumlarımıza örnekler:

- MilSoft (Level 5)
- TÜBİTAK BİLGEM Yazılım Teknolojileri Araştırma Enstitüsü (Level 5)
- ASELSAN (Level 3)
- Cybersoft (Level 3)
- Havelsan (Level 3)
- Koç Sistem (Level 3)
- Etiya (Level 3) (YTÜ Teknopark)
- ICTerra (Level 3) (İstanbul Teknopark)
- OBSS (Level 3) (İstanbul Teknopark)