

YAZILIM MÜHENDİSLİĞİ – Şubat 2012
Yrd.Doç.Dr. Yunus Emre SELÇUK
GENEL BİLGİLER

BAŞARIM DEĞERLENDİRME

- Sınav tarihleri: Daha sonra duyurulacak
 - 1. Ara sınav yazılı, 2. Ara sınav: test, Final sınavı: yazılı
- Proje ödevi: Son ders haftasında sunum yapılacaktır.
- Puanlama (değişebilir):
 - 1. Vize * %20, 2. Vize * %20, Final * %40, Proje * %20

KAYNAK KİTAPLAR

- Software Engineering: A Practitioner's Approach / Roger S. Pressman. McGraw/Hill, 2005, 6th ed.
- UML Distilled / Martin Fowler. Addison Wesley, 2003, 3rd ed.
- ... ve diğerleri

1

GENEL BİLGİLER

DERS İÇERİĞİ

- Yazılım Mühendisliğine Giriş
- Yazılım Geliştirmede Süreç Modelleri
- Gereksinim Mühendisliği
- Nesneye Yönelik Çözümleme
- Nesneye Yönelik Tasarım
- Yazılım Ölçütleri
- Yazılım Sınama Teknikleri
- Yazılım Proje Yönetimine Giriş

2

YAZILIM MÜHENDİSLİĞİNE GİRİŞ

YAZILIM

- Yazılım :
 - Herhangi bir boyuttaki herhangi bir tür donanımda çalışan bilgisayar programını VE,
 - Basılı veya elektronik ortamdaki her tür dokümanı içeren ürün.
 - Dokümanlar yazılım mühendislerine ve son kullanıcıya yönelik olabilir.
- Yazılım hem bir üründür, hem de başka ürünler elde etmeye yarayan bir araçtır.
- Yaşam döngüsü: Yazılımın bir fikir olarak doğmasından, kullanım dışı bırakılmasına kadar geçen süreç.
- 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.
- Çözüm: Yazılım mühendisliği ilkelerine uyularak daha iyi tasarlanmış yazılım.

3

YAZILIM MÜHENDİSLİĞİNE GİRİŞ

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 öğretmek / Number crunching”.
- Gömülü (Embedded) Yazılım :
 - Donanım ile çok sıkı ilişkidir.
 - Denetim amaçlıdır.
 - Gerçek zamanlı uygulamalardır.

4

YAZILIM MÜHENDİSLİĞİNE GİRİŞ

YAZILIM TÜRLERİ

- Uygulama Yazılımı :
 - Product-line, shrink-wrapped, off-the-shelf, vb.
 - 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!

5

YAZILIM MÜHENDİSLİĞİNE GİRİŞ

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

6

YAZILIM MÜHENDİSLİĞİNE GİRİŞ

YAZILIMI ETKİLEYEN EĞİMLER

- Yaygınlaşan Bilgi-İşlem :
 - Hesaplama gücünün giderek küçülen alanlara sıkıştırılabilmesi, bilişimin günlük yaşantımızla daha kolay bütünleşmesine olanak sağlıyor.
- Yaygınlaşan Haberleşme Ağı :
 - Kablosuz ağların yaygınlaşması, bilişimin günlük yaşantımızla daha kolay bütünleşmesine olanak sağlıyor.
- Özgür / Açık Kaynak Yazılım :
 - Gevşek bir ekip tarafından geliştirilen yazılım, daha anlaşılır ve geliştirilebilir olmalıdır.
- Ayrıca:
 - Takım çalışması zorunluluğu
 - Küreselleşme
 - Ekonomik krizler

7

YAZILIM MÜHENDİSLİĞİNE GİRİŞ

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.
- Sonuç: Yazılım esnek bir oyun hamurundan çok kil veya cam gibidir.
 - Çevik süreçlerle esnekliğin artırılması hedeflenmektedir.

8

YAZILIM MÜHENDİSLİĞİNE GİRİŞ

YAZILIM HAKKINDAKİ YANILGILAR: PROGRAMCI AÇISINDAN

- Yazılımı tamamlayınca 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 kullanılabilir kalite ölçütleri bulunmaktadır.
- Yazılım eşittir program.
 - Gereksinim analizi başlı başına bir emektir.
 - Dokümantasyon ve sınama 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.

9

YAZILIM MÜHENDİSLİĞİNE GİRİŞ

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 yaptırıyorum.
 - 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.

10

YAZILIM MÜHENDİSLİĞİNE GİRİŞ

YAZILIM SÜREÇLERİNİN GENEL ADIMLARI

- Çözümleme (Analysis)
- Tasarım (Design)
- Gerçekleme (Implementation)
- Sınama (Testing)
- Bakım (Maintenance)

ÇÖZÜMLEME

- Çözümleme: Bir şeyi anlayabilmek için parçalarına ayırmak.
- Sistemi anlamaya yönelik çalışmalardan ve üst düzey planlama eylemlerinden oluşur.
 - Uygulama alanı
 - Kullanıcı gereksinimleri
 - Program parçaları arasındaki üst düzey etkileşimler (NYP'deki parçalar: sınıflar ve nesnelər)
- “Bir sorunu anlamadan çözemezsiniz.”

11

YAZILIM MÜHENDİSLİĞİNE GİRİŞ

YAZILIM SÜREÇLERİNİN GENEL ADIMLARI

- Çözümleme
- Tasarım
- Gerçekleme
- Sınama
- Bakım

TASARIM

- Tasarım: Bir araştırma sürecinin çeşitli dönemlerinde izlenecek yol ve işlemleri tasarlayan çerçeve.
- Çözümleme ile anlaşılan sorun tasarım aşamasında kağıt üzerinde (!) çözülür.
- Yazılım ↔ Tasarıma yönelik şemalar (NYP'de UML şemaları), elektronik ↔ devre şemaları, mimari ↔ kat planları

GERÇEKLEME

- Eldeki tasarım, bir programlama dili ile kodlanır.

12

YAZILIM MÜHENDİSLİĞİNE GİRİŞ

YAZILIM SÜREÇLERİNİN GENEL ADIMLARI

- Çözümleme
- Tasarım
- Gerçekleme
- Sınama
- Bakım

SINAMA

- Sınama neden önemlidir?
 - Yazılım sürecinde ilerledikçe, ortaya çıkabilecek hataların giderilme maliyeti üstel olarak artar.
 - Aksi gibi, hataların büyük çoğunluğu isteklerin belirlenmesi ve tasarım aşamalarında ortaya çıkar.
 - Bu yüzden: Erkenden, sık sık ve kolay sınama yapın.

13

YAZILIM MÜHENDİSLİĞİNE GİRİŞ

YAZILIM SÜREÇLERİNİN GENEL ADIMLARI

- Çözümleme
- Tasarım
- Gerçekleme
- Sınama
- Bakım

BAKIM

- Yazılımın faaliyete geçirilmesinden sonra sistemde yapılan değişikliklerdir.
 - Yazılım hatalarının düzeltilmesi:
 - Kodlama hataları
 - Tasarım hataları (!)
 - Gereksinim ve analiz hataları (!!)
 - Sistemin işlevlerini değiştirme veya işlevlere eklemeler/çıkarmalar,
 - Yazılımın farklı bir ortama taşınması (yazılım, donanım, iklim),

14

YAZILIM MÜHENDİSLİĞİNE GİRİŞ

YAZILIM SÜREÇLERİNİN GENEL ADIMLARI

- Çözümleme
- Tasarım
- Gerçekleme
- Sınama
- Bakım

BAKIM

- Yeniden mühendislik (Refactoring / Software re-engineering)
 - Teknik bakış açısı: Yazılımın işlevini değiştirmeden iç yapısını değiştirmek.
 - Olası eylemler:
 - Yazılımın belgelendirilmesi
 - Yazılımın güncel bir programlama diline geçirilmesi
 - Tasarımın iyileştirilmesi/değiştirilmesi
 - Yazılımın farklı bir ortama taşınması