

YAZILIM MÜHENDİSLİĞİ

Prof.Dr. Oya Kalıpsız

GİRİŞ

1

YAZILIM YETERLİLİK OLGUNLUK MODELİ

CMM: Yeterlilik Olgunluk Modeli (Capability Maturity Model)

- Olgunluk Seviyeleri:
 - Düzey 1. Başlangıç düzeyi: Yazılım gelişimi ile ilişkili süreçlerin tanımlanması için hiçbir sistematik girişim bulunmamaktadır.
 - Düzey 2. Tekrarlanabilir düzey: Projeler içerisinde etkili yönetim uygulamalarının geliştirilmesi üzerinde önemle durulmaktadır.
 - Düzey 3. Tanımlanmış düzey: Tüm organizasyonlar içerisinde standart süreçler tanımlanmıştır ve bu süreçler özel projelerde kullanılmak üzere yeniden düzenlenmiştir.
 - Düzey 4. Yönetilebilir düzey: Süreçler ve ürünler için nicel kaliteyi amaçlar, ve bu amacı gerçekleştirmek için yazılım ölçütlerini kullanır.
 - Düzey 5. Optimize edilmiş düzey: Teknolojik yeniliklerin sürekli süreç gelişimi ve değişik süreçlerin nicel maliyet-kar analizleri üzerinde odaklanmıştır

2

YAZILIM YETERLİLİK OLGUNLUK MODELİ

CMM: Yeterlilik Olgunluk Modeli (Capability Maturity Model)

- CMM'de Bilgisayar destekli yazılım mühendisliği gereçlerinin (CASE: Computer Aided Software Engineering tools) kullanımı:
 - Düzey 1. Başlangıç düzeyi:
 - Yazılım araçlarının kullanımına dair hiçbir standart yoktur.
 - Yazılım ölçütleri kullanılmaz.
 - Yeniden kullanılabilirlik yoktur veya minimal düzeydedir.
 - Düzey 2. Tekrarlanabilir düzey:
 - Yazılım aracı kullanımına yönelik proje standartları vardır.
 - Çalışanlara yazılım araçlarının kullanımına yönelik eğitim verilmiştir.
 - Bazı takım tabanlı yazılım araçları, diğer yazılım araçları ile tümleştirilmiştir (entegrasyon)
 - Tasarım ve kodun bir kısmı sınırlı olarak yeniden kullanılabilir.

3

YAZILIM YETERLİLİK OLGUNLUK MODELİ

CMM: Yeterlilik Olgunluk Modeli (Capability Maturity Model)

- CMM'de Bilgisayar destekli yazılım mühendisliği gereçlerinin (CASE: Computer Aided Software Engineering tools) kullanımı (devam):
 - Düzey 3. Tanımlanmış düzey:
 - Yazılım araçlarının erişebildiği, organizasyona ait veri deposu oluşturulmuştur.
 - Ortak tekrar kullanım kütüphanesi oluşturulmuştur.
 - Organizasyon standartlarına bağlı olarak yazılım ölçütleri (metrikler, ileride değinilecek) toplanmıştır.
 - Ölçütler eğilimler ve profillerin belirlenmesi gibi işlemlerde kullanılmak üzere organizasyona ait veri tabanında saklanmıştır.
 - Düzey 4. Yönetilebilir düzey:
 - Yazılım ölçütlerinin niteliksel analizi ile süreç gelişimi oluşturulmuştur.
 - Düzey 5. Optimize edilmiş düzey:
 - Yeni yazılım araçları dahil olmak üzere tüm gereçler, teknikler ve proje yönetim süreçleri değerlendirilip sürekli iyileştirmeye tabi tutulmaktadır.

4

YAZILIM GELİŞTİRME YÖNTEMLERİ

DÖRDÜNCÜ KUŞAK TEKNİKLERİ

- 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 kullanılabilmektedir

5

CASE SİSTEMİ

GENEL BİLGİLER

- 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

6

CASE SİSTEMİ

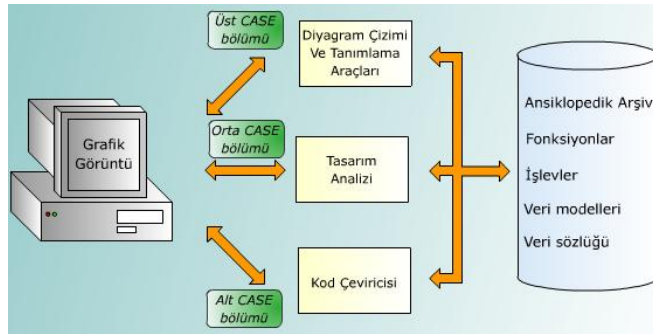
CASE SİSTEMİ ÖĞELERİ

- 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.

7

CASE SİSTEMİ

CASE TEKNOLOJİSİ



- Bölümler:
 - Üst CASE bölümü: Bilgisayara dayalı diyagram çizim araçları şeklinde arayüz üzerinden kullanılan, iş mantığını betimleyen gereçler.
 - Orta CASE bölümü: Sistem analizi ve tasarımı gereçleri.
 - Alt CASE bölümü: Kod üretim gereçleri
- Orta ve alt bölümlerde ölçüm ve belgelendirme ile hata ayıklama gereçleri de yer almaktadır (otomatik).

8

YAZILIM MÜHENDİSLİĞİ
Prof.Dr. Oya Kalıpsız

YAZILIM GEREKSİNİM ANALİZİ MODELLERİ

9

YAZILIM GEREKSİNİMLERİ

YAZILIM GEREKSİNİMLERİ SPESİFİKASYONU

- Yazılım gereksinimlerinin tanımlanmasında kullanılan araçlara değinilmiştir.
 - SRS: Software Requirements Specification
- SRS raporuna yönelik çeşitli standartlar mevcuttur.
 - IEEE 830-1998: SRS odaklı
 - ISO/IEC/IEEE 29148:2011 Systems and Software Engineering – Life Cycle Processes – Requirements Engineering
- SRS raporunda yer alması önerilen konu başlıkları şunlardır:

10