

1) Bilgisayar Mühendisliği

Algoritmalar

Donanım

Yazılım geliştirme

İşletim sistemleri

Derleyiciler

Hesaplama kuranları

Yazılım Mühendisliği

Yazılım mimarisi

Yazılım kalitesi ve güvenliği

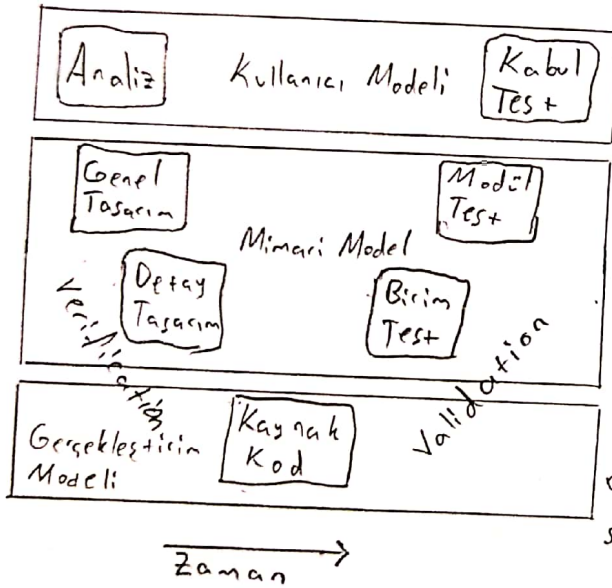
Proje yönetimi

Teknik planlama

Risk yönetimi

Bilgisayar mühendisliği hem yazılım hem donanım işleriyle ilgilidir. Yazılım mühendisliği ise nispeten daha çok yazılım ile ilgilidir. Ayrıca yazılım mühendisliği, bilgisayar mühendisliğine göre donanıma çok daha az odaklanmıştır, programların olması gerektiği gibi ve güvenli gelişim sürecine daha fazla vurgu yapar.

2)



Sol tarafta üretim, sağ tarafta sınamaya başlanmaktadır.

Kullanıcı modeli: Geliştirme sürecinin kullanıcı ile olan ilişkilerini belirtmekte ve sistemin nasıl kabul edileceğine ilişkin sınamaya planları ortaya çıkarılmaktadır.

Mimari model: Sistem tasarımı ve oluşacak alt sistem ile tüm sistemin sınamaya işlemleri.

Gerçekleştirim modeli: Yazılım modüllerinin kodlanması ve test edilmesi.

- Model, kullanıcının projeye katkısını artırır.
- Belirsizliklerin az, iş tanımlarının belirlenmiş olduğu BT projeleri için idealdir.
- BT projesinin iki aşamalı olarak ihale edilmesine uygundur:
 - İlk ihalede kullanıcı modeli hedeflenerek, iş analizi ve kabul sınamalarının tanımları yapılmaktadır,

• İkinci ihalede ise ilkinde elde edilmiş olan kullanıcı modeli tasarlanıp gerçekleştirilmektedir.

3)

Planlama: Temel ihtiyaçlar belirlenir, proje için fizibilite çalışmaları yapılır ve proje planlaması gerçekleştirilir.

Analiz: Sistemin işlevlerini ve gereksinimlerini açıklığa kavuşturarak dokümanla etmektir. Gereksinimlerin net olmadığı durumlarda mühendis ve müşteri arasında iletişim ve beraber çalışmanın daha fazla olması gerekir. Geşitli yazılım geliştirme metodolojilerinde bu aşamada kullanım dokümanları ve test plan dokümanları da oluşturulabilir.

Tasarım: Müşterinin gereksinim ve isteklerini karşılamak üzere yazılım ürününün özelliklerinin belirlenmesidir. İki tür tasarımdan bahsedilebilir: yüksek düzeyde tasarım, mimari tasarım ve detay tasarım.

Gerçekleştirme: Tasarım aşamasının belirli bir olgunluğa ulaşmasıyla birlikte kodlama aşaması başlar. Müşteriye teslim edilecek ürünü programlama aşamasıdır.

Bakım: Teslim ile birlikte bakım aşaması da başlar. Hata giderici, altyapıyı iyileştirici, önleyici, ürüne yeni özellikler kazandırıcı gibi farklı bakım faaliyetleri mevcuttur.

4)

İşlev noktaları geliştirmenin erken aşamalarında sapmanın değere denir. Sistemin oluşturulduğu ortamdan bağımsız elde edilir. Problem tanımlı girdi olarak üç adım izlenir: problemin bilgi ortamının incelenmesi, problemin teknik karmaşıklığının incelenmesi, işlev noktası hesaplama.

Öncelikle problem bilgi ortamının incelenmesinde kullanıcı ifadeleri, kullanıcı çıktıları, kullanıcı sorguları, dosyalar, dışsal arayüzler gibi etmenlerin tahmin edilmesi gerekir. Bu etmenler sayısal olarak karmaşıklığına göre ağırlık faktörü ile aarpılır ve hepsinin toplamı alınır. Böylece AIN bulunur. Problemin teknik karmaşıklığının incelenmesinde ise uygulamanın güvenliği, performansı, iletişimi,

İşsel karmaşıklık gibi konular üzerine bir takım sorular sorulur. Daha sonra bu sorular 5 üzerinden puanlanarak TKF elde edilir.

$$IN = AIN (0,65, 0,01, TKF)$$

Çıkan sonuç ile üretkenlik, kalite ve maliyet gibi konular da fikir edinilebilir.

5)

Rely: Yazılım güvenilirliği.

Data: Veri tabanının büyüklüğü.

Cplx: Karmaşıklık

Time: İşletim zamanı kısıtı

Stoc: Ana bellek kısıtı

Virt: Bilgisayar platform değişim olasılığı.

Turn: Bilgisayar iş geri dönüş zamanı.

Acap: Analist yeteneği

Aexp: Uygulama tecrübesi

Pcap: Programcı becerisi

Ve xp: Bilgisayar platformu tecrübesi

Lexp: Programlama dili tecrübesi

Modp: Modern programlama teknikleri

Tool: Yazılım geliştirme araçlarının kullanımı

Sced: Zaman kısıtı