



SAKARYA
ÜNİVERSİTESİ

BSM 101

BİLGİSAYAR MÜHENDİSLİĞİNE GİRİŞ

İSMAİL ÖZTEL

~ Yazılım Mühendisliği ~

İÇERİK

- Sistem ve yazılım
- Yazılım mühendisliği yöntembilimleri
- Yazılım geliştirme modelleri
- Yazılım geliştirme süreci
- Yardımcı süreçler



Giriş

- Bilgisayar sistemleri geliştikçe insanların bu sistemlerden beklentileri de artmaktadır.
- Bu beklentilere cevap vermek için basit programlar değil, iyi tasarlanmış ve geliştirilmiş yazılımlara ihtiyaç vardır.
- Giderek büyüyen ve karmaşıklaşan yazılımlar, yazılımların tasarlanmasını, gerçekleştirilmesini, sınanmasını da güçleştirmektedir.
- Bu güçlükleri giderebilmek için disiplinli çalışma yöntemlerine ihtiyaç duyulur.

Sistem ve yazılım

- Günümüzde bir yazılım yalnızca bir bilgisayar programı değil; belirli bir yöntem dahilinde geliştirilen bir program, veri ve belgeler topluluğudur.
- Genellikle, ticari olarak en maliyeti sistem bileşeni yazılımlardır.
- Bir yazılımla birlikte bir firma, diğerlerinin önüne geçebilir; askeri bir sistem diğerlerinden daha üstün hale gelebilir.
- Aynı işleve sahip yazılımlar arasında bir tercih yaparken kullanım kolaylığı, etkin ve hatasız çalışma gibi özellikler belirleyici olmaktadır.
- Bu unsurların yazılımlarda meydana gelebilmesi için de özel bir yaklaşım gereklidir.

Sistem ve yazılım

- Bilgisayar sistemi mühendisliği:
 - Yazılım, donanım ve belgelendirme ile birlikte bir görevi / görevleri yerine getirecek elemanları tanımlayıp bir araya getiren çalışmaları içerir.
 - Bilgisayar sistemi mühendisliği süreçlerinde nasıl bir sistem geliştirileceğine, hangi yazılım ve donanımın hazır alınacağına, hangisinin geliştirileceğine karar verilir.
 - Sistemler proje olarak geliştirilir ve bir proje yaşam döngüsü kullanmak çok önemlidir.

Sistem ve yazılım

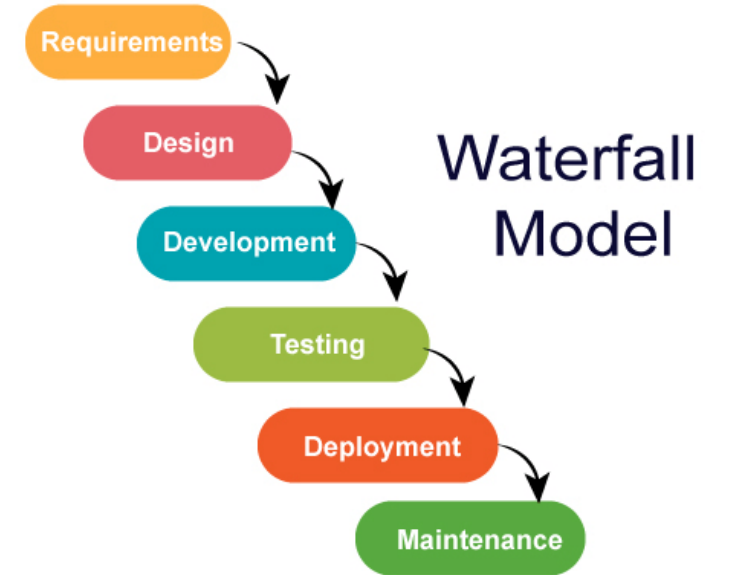
- Sistem geliştirme süreci:
 - Her projeden önce sistemi geliştirmek için bir “ yapılabirlik ” araştırması yapılır.
 - Sistem düzeyinde ilk tasarım yapılır.
 - Sistemin genel mimarisi ortaya konur.
 - Sistem gerçekleştirme yazılım ve donanım olmak üzere iki koldan icra edilir.
 - Yazılım ve donanım tümleştirme işlemi yapılır.
 - Sistem test edilir.
 - Sistem kullanıcıya teslim edilir
 - Sistem devrede kaldığı sürece sistem bakım aşaması devam eder.

Yazılım Mühendisliği Yöntembilimleri

- Yazılım geliştirme yaşam döngüsü boyunca kullanılacak süreç, belirtim ve belgelendirme gibi yöntemleri içeren disipline yöntembilim denir.
- Yazılım Geliştirme Standartları:
 - Uluslararası kuruluşlar tarafından çeşitli yazılım geliştirme standartları geliştirilmektedir.
 - Buradaki amaç ortak bir noktaya doğru ilerlemektir.
 - DOD-STD-2167A standardı 1988 yılında yürürlüğe girdi, ağırlıklı olarak askeri sistem yazılımlarını geliştirmek için kullanıldı.
 - Sonrasında MIL-STD-498 kullanılmaya başlamıştır.
 - MIL-STD-498 basitleştirilerek ISO/IEC 12207 oluşturulmuştur.
 - Sonrasında IEEE/EIA 12207 software life cycle processes standart olarak benimsenmiştir.

Yazılım Mühendisliği Yöntembilimleri

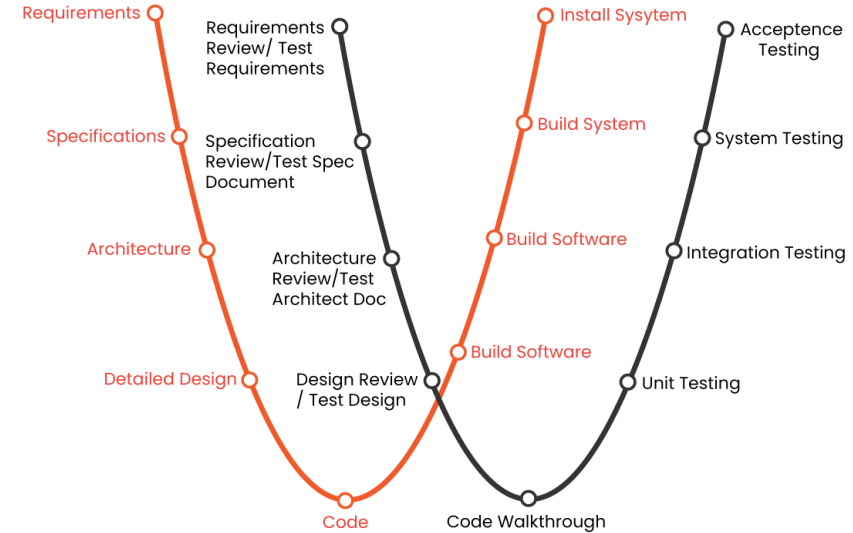
- Yazılım Geliştirme Modelleri:
 - Klasik Çevirim / çağlayan / şelale / büyük tasarım / geleneksel model:
 - Arka arkaya, sistematik olarak ilerleyen bir yazılım geliştirmeyi sağlar.
 - Öncelikle tüm ihtiyaçlar tanımlanır.
 - İhtiyaçlara göre tasarım yapılır.
 - Gerçekleme yapılır.
 - Geliştirilen bölümler tümleştirilir.
 - Sistem testleri yapılır.
 - Kullanıcıya teslim edilir ve bakım aşaması başlar.
 - Aşağıdakilerin geçerli olduğu projelerde uygun bir yöntemdir:
 - İyi tanımlı, ihtiyaçları kesinleşmiş
 - Fazla zaman almayacak olan



<https://www.javatpoint.com/jira-waterfall-model>

Yazılım Mühendisliği Yöntembilimleri

- Yazılım Geliştirme Modelleri:
 - V modeli:
 - Klasik modeldeki test işlemlerinin ne zaman yapılacağı belirlenir.
 - Modeldeki sol kol üretim, sağ kol test ile ilgilidir.
 - Aşağıdakilerin geçerli olduğu projelerde uygun bir yöntemdir:
 - İsterlerin iyi tanımlandığı
 - Belirsiz durumların az olduğu
 - Aşamalar halinde ilerlenmesi gereken



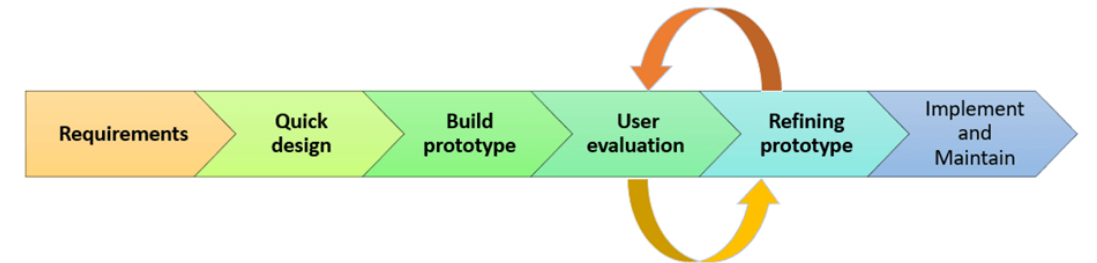
<https://www.testbytes.net/blog/v-model-and-w-model-software-testing/>

Yazılım Mühendisliği Yöntemleri

- Yazılım Geliştirme Modelleri:

- Prototipleme:

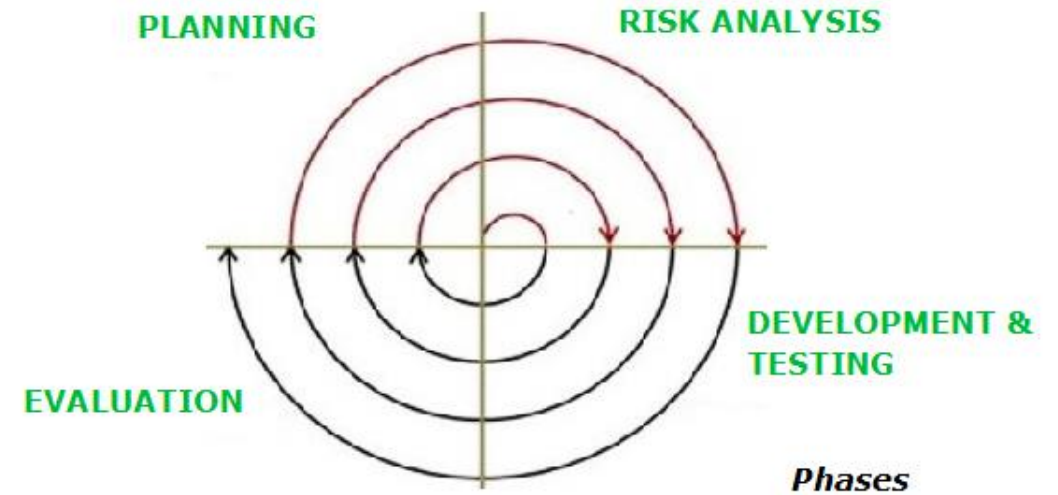
- Bazen müşteri yazılımdan ne istediğini belirtir ancak detaylandıramaz.
- Bazen da yazılım geliştirici kullanılacak donanım ve işletim sisteminden, kullanılacak algoritmanın veriminden emin olmayabilir.
- Bu gibi durumlarda prototipleme yaklaşımı kullanılabilir.



<https://www.guru99.com/software-engineering-prototyping-model.html>

Yazılım Mühendisliği Yöntembilimleri

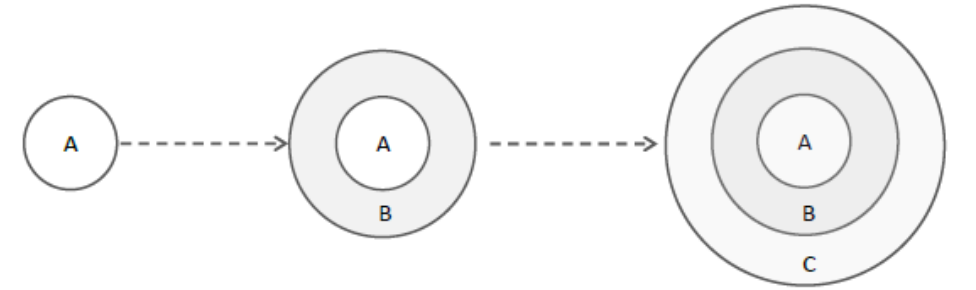
- Yazılım Geliştirme Modelleri:
 - Spiral model:
 - Hem klasik çevrim hem de prototipleme yöntemlerinin iyi yönlerini alarak geliştirilmiş bir yöntemdir.



<https://www.professionalqa.com/spiral-model>

Yazılım Mühendisliği Yöntembilimleri

- Yazılım Geliştirme Modelleri:
 - Evrimsel geliştirme modeli:
 - Evrimler halinde ürün oluşturmayı hedefler.
 - Her çevrimde geliştirilen ürünler uygulamada tam işlevselliğe sahiptir. Ortaya çıkan her ürün kullanıma sunulur.
 - Kullanım sırasında elde edilen tecrübeler ile bir sonraki evrime geçilir.
 - Her evrim kapsamı ve işlevleri biraz daha arttırır.



<https://ehindistudy.com/2020/07/02/evolutionary-model-in-hindi-software-engineering/>

Yazılım Geliştirme Süreci

- Yazılım İsterlerinin Çözümlemesi
 - Geliştirilen yazılım kalitesi doğrudan müşteri isteklerinin ne kadar yerine getirildiği ile alakalıdır.
 - Yazılımın “ ne “ yapacağı bu aşamada belirlenir.
 - İsterlerin çözümümlenmesi sırasında üretilen belgelerde çeşitli teknikler kullanılabilir:
 - Kullanım senaryoları
 - Veri akış diyagramları
 - Varlık ilişki diyagramları, vb.
 - İsterlerin sınıflandırılması daha iyi anlaşılmasını kolaylaştıracaktır:
 - Bu aşama sonrasında isterler belirlenmiş, belgelendirilmiş olacaktır.

Yazılım Geliştirme Süreci

- Yazılım İsterlerinin Çözömlenmesi
 - Bu belge/belgelerin içerisinde aşağıdakiler bulunabilir:
 - Sistemin genel anlatımı
 - Çalışma sırasında sistemin içinde bulunabileceği durumlar
 - Sistemin her bir işlevinin tanımları
 - Arayüz isterleri
 - Emniyet ve güvenlik isterleri
 - Donanım kaynakları
 - Kısıtlar (tasarıma ve/ veya gerçekleştirime ait)

Yazılım Geliştirme Süreci

- Yazılım Tasarımı
 - İyi bir bilgisayar yazılımı tasarım ile başlar.
 - Hiçbir yazılıma tasarım olmadan başlanmamalıdır.
 - Tasarım sırasında tekrarlanarak yapılan işlemler:
 - Sistemin bütün olarak tasarlanması
 - Var olan öğelerin geliştirilecek olan yazılım için belirlenmesi ve hazırlanması
 - Yeni öğelerin belirlenmesi, vb.

Yazılım Geliştirme Süreci

- Yazılım Tasarımı
 - Yazılım tasarımı ön tasarım ve ayrıntılı tasarım olarak iki aşamada gerçekleşir ve nihayetinde kodlama için kullanılacak veri tasarımı, mimari tasarım, yordamsal tasarım ve arayüz tasarımı elde edilir.
 - Veri tasarımı: veriye erişim yöntemi, hız, etkinlik gibi kriterler için en uygun veri türleri ve yapıları belirlenir.
 - Mimari tasarım: Yazılım modülleri belirlenir.
 - Yordamsal tasarım: Modüllerin iç yapılarının algoritmik olarak (metinsel ya da grafiksel) ifade edilmesi
 - Arayüz tasarımı: grafiksel kullanıcı arayüzleri ve donanım tümleştirme yöntemleri belirlenir.

Yazılım Geliştirme Süreci

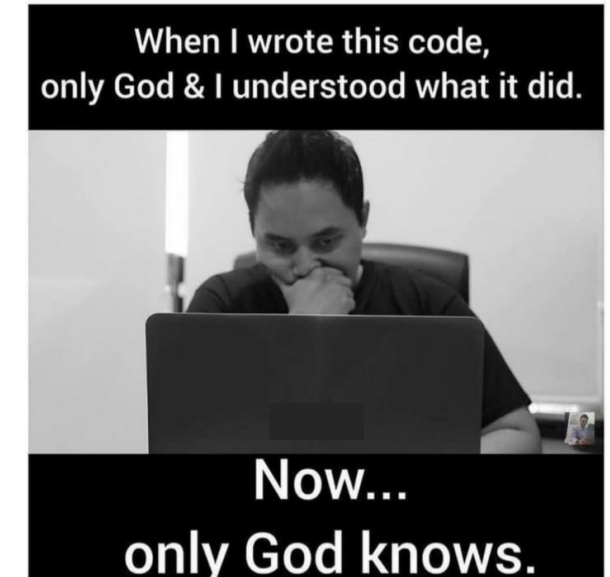
- Yazılım Tasarımı
 - Tasarımlar el yordamı ile ya da yardımcı araçlar ile yapılabilirler.
 - Tasarım aşaması sonunda tasarımlar mutlaka belgelendirilmelidir. Bu belgede genel olarak aşağıdaki konu başlıkları yer alabilir.
 - Arayüzler
 - Kullanılan veritabanları
 - Kısıtlamalar
 - Veri tanımlaması
 - Ayrıntılı modül tasarımları

Yazılım Geliştirme Süreci

- Yazılım Gerçekleştirimi
 - Bu aşamada tasarım ile ortaya çıkan öğeler kodlama dilleri ile makinelerin anlayabileceği hale getirilir.
 - Yazılan kodlar “kaynak kod” olarak isimlendirilirken, bu kodların üzerinde çalışacağı donanımın anlayacağı şekilde makine koduna dönüştürülürler.
 - Kodlamaların belirli kısımları ya da tamamı belirli bir şablonla yapılabilir.
 - Daha önce kullanılmış kod modülleri de tekrar kullanılabilir.

Yazılım Geliştirme Süreci

- Yazılım Gerçekleştirimi
 - Kod yazarken aşağıda yer alan hususlara dikkat edilmelidir:
 - Tasarruflu bellek kullanımı
 - Atık toplama
 - Hataya dayanıklılık
 - Hızlı çalışma
 - Kullanıcı dostu olma
 - Modülerlik
 - Tekrar kullanım
 - Kodu anlatan açıklama satırları
 - Kodu anlatan açıklama satırları ve kodun kendisi bu aşamada belgelendirme olarak kabul edilebilir. Gerekirse basit metin dosyaları da ek açıklamalar için kullanılabilir.



Yazılım Geliştirme Süreci

- Yazılım Testi
 - Çözümleme, tasarım ve gerçekleştirme aşamalarından sonra son bir kontrol için yapılır.
 - Yazılımlarda yürütme anında sorun çıkmaması için, yazılımın önceden iyi bir testten geçirilmesi gerekir.
 - İyice sınanmadan teslim edilen bir yazılım telafisi olmayan hatalara sebep olabilir.
 - Yazılım testleri, yeterlilik testi ve sistem testi olmak üzere iki aşamada yapılabilir:
 - Yeterlilik testi; yazılımın, istelere tam olarak cevap verip veremediğinin anlaşılması için yapılır.
 - Sistem testi; yazılım üzerinde koşacağı donanım ile birlikte test edilmesidir.

Yazılım Geliştirme Süreci

- Yazılım Testi
 - Bu aşamanın sonunda ortaya çıkan dökümanlarda genel olarak aşağıdaki hususlar belgelendirilir:
 - Sistemin nasıl test edileceği ve gerekli personel ile ekipmanlar
 - Test takvimi
 - Senaryolar ve beklenen sonuçlar
 - Her senaryonun sonucu, karşılaşılan sorunlar ve ilave istekler

Yazılım Geliştirme Süreci

- Yazılım Bakımı
 - Yazılımın kullanıcıya tesliminden sonra bakım aşaması başlar.
 - Donanımsal ve yazılımsal olarak ikiye ayrılır:
 - Donanımsal bakım: donanım temizliği, parça değişimi, komple donanım değişimi
 - Yazılımsal bakım:
 - Teslimden sonra ortaya çıkan hatalı hesaplama, hatalı davranış gibi yazılımsal problemleri gidermek için yapılan yazılımsal müdahale
 - Donanım ya da işletim sistemi gibi değişikliklerde yazılımın ortama uyum sağlaması için yapılan bakım
 - Belirlenen isterlerin dışında sistem performansını arttıracak son kullanıcı tavsiyelerine göre yapılan yazılımsal iyileştirmeler
 - Geliştirici tarafından sonradan fark edilen, olabilecek hataları önlemek için yazılımda değişiklik