

Yazılım Mühendisliği

Balık Kılıcı Tekniği

Yazılım Metodolojileri

Yazılım Yaşam Döngüsü Temel Adımları (Çekirdek Süreçler)

Gelişigüzel Model

Barok Modeli

Çağlayan (Şelale) Modeli

V Süreç Modeli

Helezonik (Spiral) Model

Evrimsel Geliştirme Süreç Modeli

Artırımsal Geliştirme Süreç Modeli

Araştırma Tabanlı Süreç Modeli

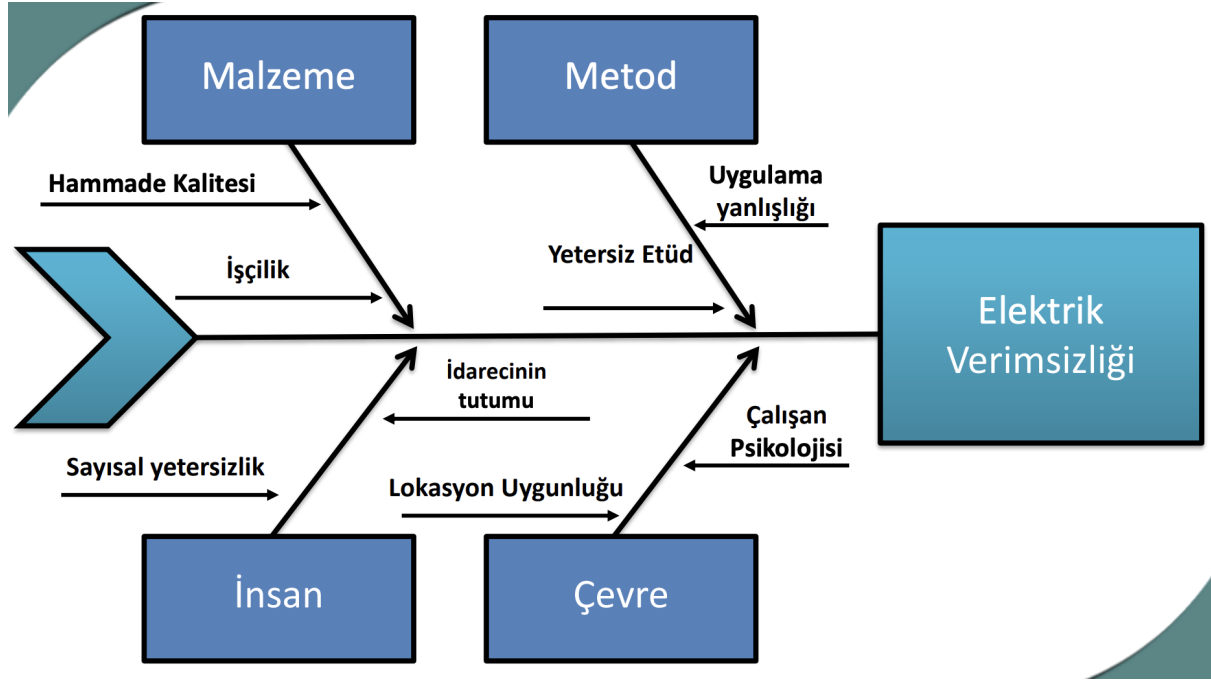
Yazılım Testi

Balık Kılıcı Tekniği

- Ishikawa diagramı olarak da bilinir.
- Bu teknik ile bir problemin nedenlerini bulmamızı, nedenleri ve nedenler arasındaki ilişkileri incelememizi sağlar. Fakat problem için çözümler oluşturmaz.
- Bu tekniğin işe yaraması için nedenlerin tartışıldığı ortamda fikir özgürlüğü ve tarafsızlık olmazsa olmazdır.

Uygulanış

- Problemin Belirlenmesi
- Nedenleri tespit edilecek problemin başlığı diyagramın baş kısmına yazılır.
- Daha sonra balığın omurgasına 45 derecelik açıyla oklar çizilir ve okların üzerine ana nedenler yazılır.
- Son olarak ana nedene de oklar çizilerek bu nedenlerin alt nedenleri yazılır.



Yazılım Metodolojileri

Yazılım Yaşam Döngüsü Temel Adımları (Çekirdek Süreçler)

- Analiz
- Planlama
- Tasarım
- Gerçekleştirim
- Bakım

Gelişigüzel Model

- Herhangi bir kural yok. Tamamen kişiye bağımlı.
- Genelde tek kişinin geliştirdiği yazılımlarda kullanılır.
- Basit yazılımlar için kullanılabilir. Karmaşık yazılımlar için çok yetersizdir.

Barok Modeli

Günümüzde belgeleme yazılım geliştirmenin doğal bir ürünü olmasına karşın burada ayrı bir adım olarak ele alınır. Ayrıca aşamalar arası geri dönüşlerin nasıl yapılacağı belirsizdir. Yazılımın gerçekleştirim aşamasına ağırlık verir.

Adımlar:

- İnceleme
- Analiz
- Tasarım
- Kodlama
- Modül Testleri
- Alt Sistem Testleri
- Sistem Testi
- Belgeleme
- Kurulum

Çağlayan (Şelale) Modeli

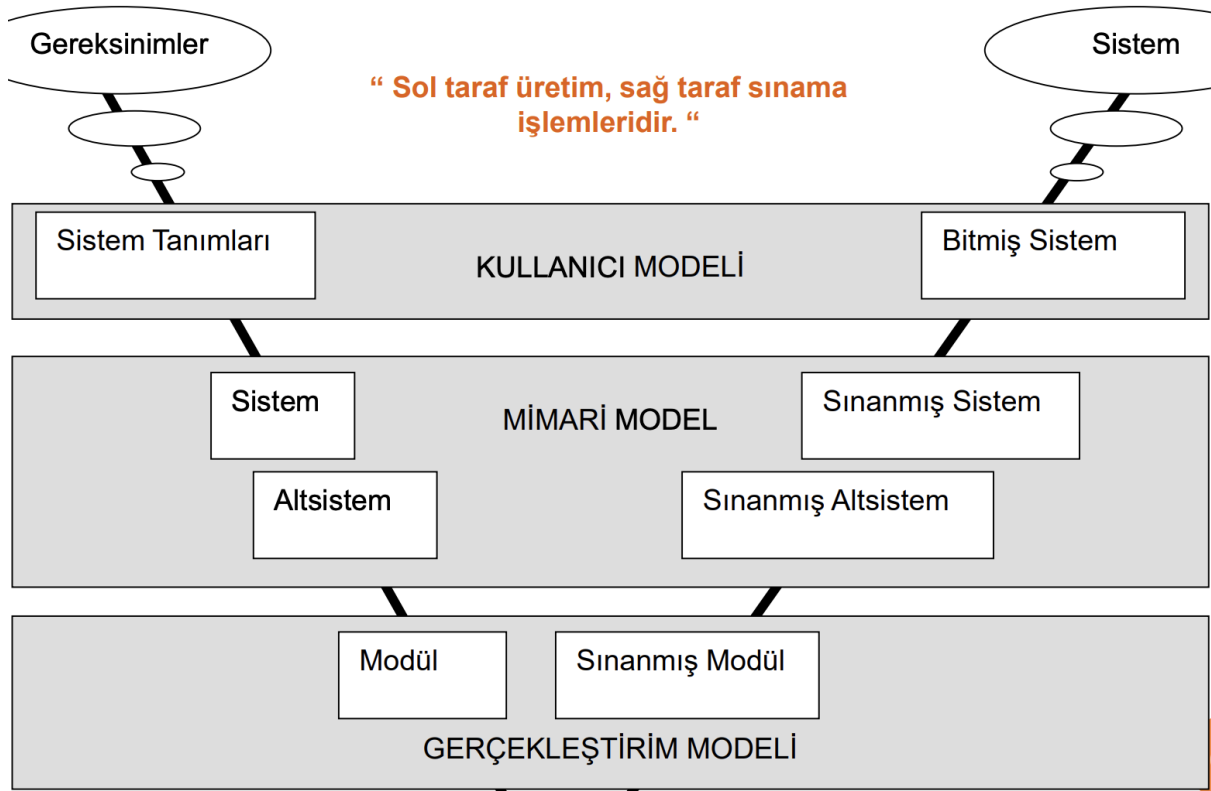
Geleneksel model olarak da bilinir. Bu model iyi tanımlı ve az zaman gerektiren projelere uygundur. Eğer tanımlamalarda problemler olursa geri dönüşlerin maliye etkisi vardır. Belgelemeyi işin doğal bir parçası olarak görür. Ayrıca adımlar arası geri dönüşler tanımlıdır. Bu modelde gerçekleştirim adımına gelmek vakit aldığı için projedeki çalışanların motivasyonu bozulabilir ve patronların gözünde proje kusurlu gözükebilir.

Adımlar:

- Gereksinimlerin Tanımlanması
- Sistem ve Yazılım Tasarımı
- Kodlama ve Modül Testi
- Birleştirme ve Sistem Testi
- Sistem Bakımı

V Süreç Modeli

Kullanıcının da projeye katkısı olmaktadır. Geleneksel model gibi belirsizliklerin az olduğu projelere uygundur.



Helezonik (Spiral) Model

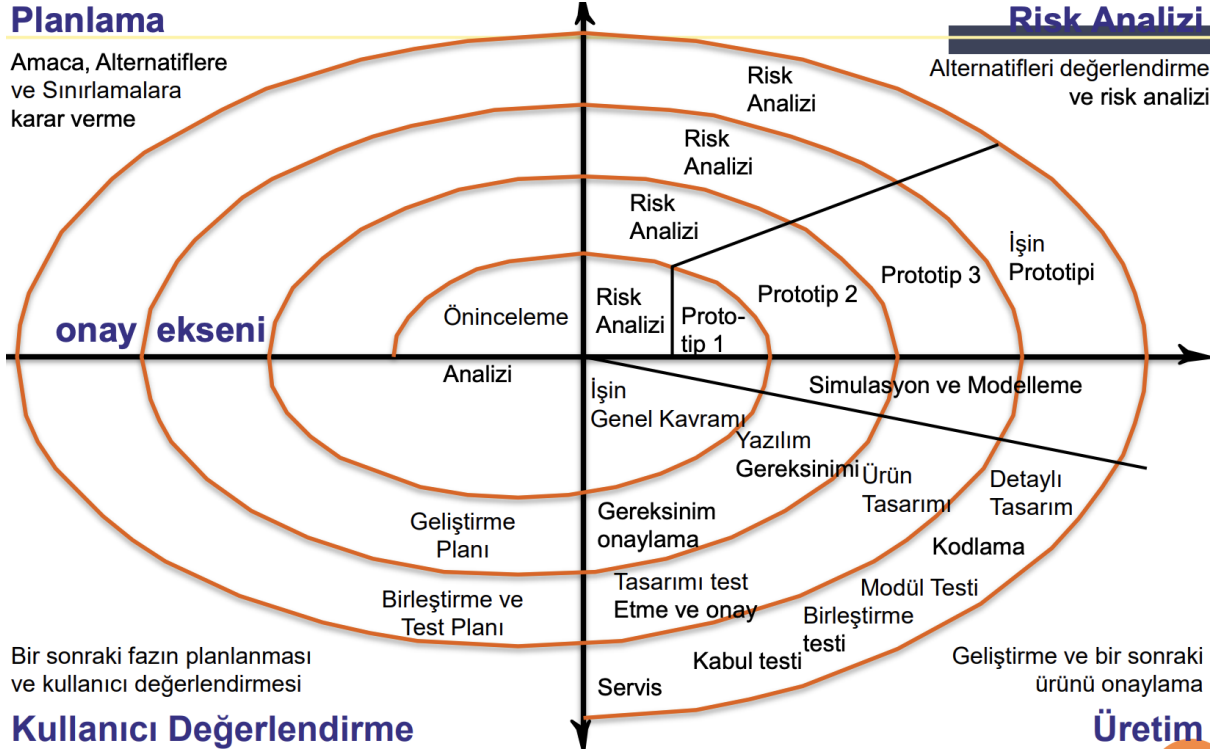
Kullanıcıyı projeye daha çok dahil eder. Birden bütün bir yazılım yapmaktansa küçük prototipler oluşturulur. Bu oluşturulan prototipler kullanıcılara onayına sunulur. Böylece proje geliştirilirken kullanıcı ile sürekli iletişim halinde kalınır. Ayrıca elde çalışan bir yazılım olduğu için işveren projeye daha olumlu bakar. Risk analiz olgusu ön plana çıkmıştır ve yinelemeli artımsal bir yaklaşım vardır.

Planlama

Amaca, Alternatiflere ve Sınırlamalara karar verme

Risk Analizi

Alternatifleri değerlendirme ve risk analizi

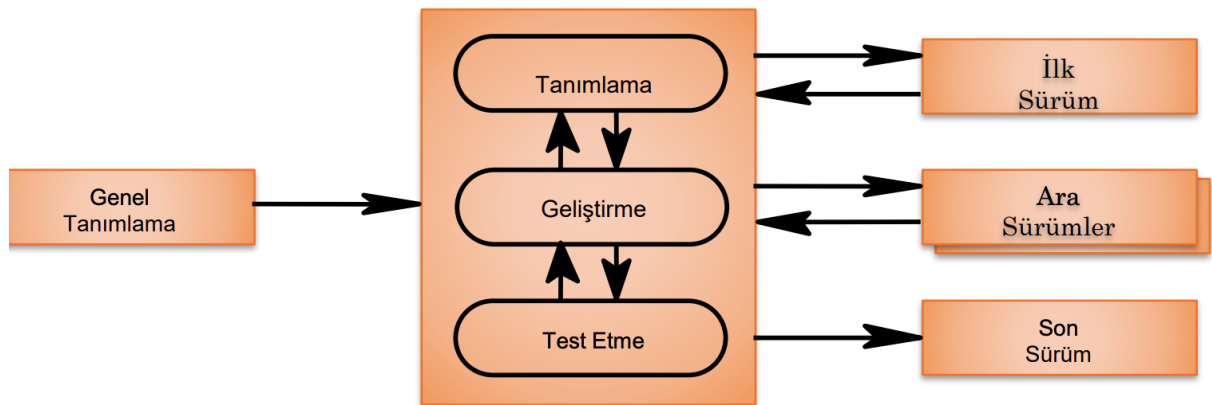


Evrimsel Geliştirme Süreç Modeli

İlk tam ölçekli modeldir ve farklı yerlerde bir çok çalışma birimleri olan şirketler için uygundur. Önce sistemler geliştirilir ondan sonra pilot alanlara yollanır. Geri bildirimlere göre düzeltmeler yapılır. Zamanla geliştirilen yazılım bütün birimlere iletilir.

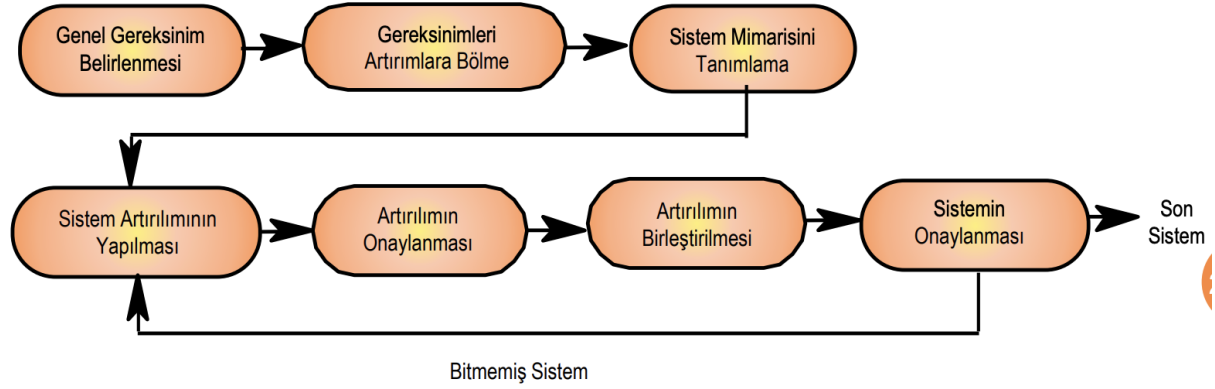
Pilot uygulama yapılmasının faydaları:

- Çalışanlar yazılıma alışır ve tepkileri en aza indirger.
- Daha önce denenmemiş şeylerin geçerliliğinin test edilmesine yardımcı olur.
- Sistemi kabul öncesi denenmesini sağlar.



Artırımsal Geliştirme Süreç Modeli

Yazılım sürümler şeklinde geliştirilir. Her sürüm bir önceki sürümü kapsar. Buna örnek olarak öğrenciler bir ödevini yapması ve tez yazmasını örnek verebiliriz. Geliştirilen yazılım eksiklikleri kullanılabilecek durumda ise bu model kullanılabilir.



Araştırma Tabanlı Süreç Modeli

Yap-at prototipi olarak da bilinir. Geliştirilecek yazılımın sonuçları çok belirsiz, sınırlı kullanıma sahip veya kısa bir zaman sonrasında kullanılmayacaksa bu model kullanılabilir. Bu modelde zaman ve fiyat kestiriminden söz edilemez. Bu model daha çok araştırma projelerinde kullanılır.

Not: Günümüzdeki metodolojiler genelde Çağlayan ya da Helezonik modeli temel almaktadır

Yazılım Testi

Bir yazılım ürününün testi ; ürünü son kullanıcıya teslim edilmeden önce yazılımın tüm yönleriyle kontrol edilmesidir.

- Saydam Kutu Testi
- Kara Kutu Testi
- Birim Testi
- Tümlleştirme Testi
- Yeterlilik Testi
- Sistem Testi

- Kabul Testi
- Entegrasyon Testi
- Kullanıcı Arayüzü Testi
- Performans Testi
- Yük Testi
- Lokalizasyon Testi
- Kullanışlılık Testi
- Güvenlik Testi
- Uyumluluk Testi
- Doyum Testi
- Kurma/Kaldırma Testi
- Ağ Testi
- Artımsal Tümlleştirme Testi
- Tümlleşim Testi
- İşlevsellik Testi
- Regresyon Testi
- Alfa Testi
- Beta Testi