YILDIZ TEKNIK ÜNİVERSİTESİ

YAZILIM MÜHENDİSLİĞİ

Mustafa Yoğurtçu
Özgür Kuru

İçerik

- NEDİR?
- TARİHÇE
- YAZILIM TÜRLERİ
- YAZILIM GELİŞTİRME ADIMLARI
- MODELLER
- TESTLER

NEDİR?

YAZILIM NEDİR?

Herhangi bir boyuttaki herhangi bir tür donanımda çalışan bilgisayar programını ve basılı veya elektronik ortamdaki her tür dokümanı içeren ürün.

NEDİR?

YAZILIM MÜH. NEDİR?

Yazılım Mühendisliği; sistemli, düzenli, ölçülebilir bir yaklaşımın yazılım geliştirmede, yazılımın işlenilmesinde ve bakımında uygulanmasıdır.

YAZILIM MÜH. AMACI NEDİR?

- Kullanıcı ihtiyaçlarını analiz ederek, uygun çözümler üretebilmek,
- Kullanıcının belirlediği fakat genelde sürtüşmelere yol açan zaman, maliyet, kullanılabilirlik gibi noktalarda uzlaşma sağlayabilmek,
- Mühendislik yaklaşımlarını kullanırken etik, sosyal, yasal ve ekonomik ilgileri bütünleştirecek uygun çözümleri tasarlayabilmek,

İLK ORTAYA ÇIKIŞ

Yazılım Mühendisliği terimi ilk kez 1950'lerin sonunda 1960'ların başında görülmeye başlandı. Programcılar zaten inşaat, elektrik ve bilgisayar mühendisliklerini biliyorlardı ve yazılım için mühendisliğin ne olduğunu tartışmaya başladılar.

İLK ORTAYA ÇIKIŞ

NATO Bilim Komitesi, 1968'deki ve 1969'daki Yazılım Mühendisliği üzerine bu alana destek olan iki konferansa (Garmisch, Almanya) destek oldu. Çoğu kişi bu konferansların yazılım mühendisliğindeki resmi başlangıç olduğuna inanır.

YAZILIM TÜRLERİ

- **Sistem Yazılımı** :
  Diğer programlara hizmet sunmak üzere hazırlanmış programlar.
- **Mühendislik Yazılımı / Bilimsel Yazılım** :
  Mühendislik ve bilimsel hesaplamalarda kullanılmak üzere hazırlanmış programlar.

YAZILIM TÜRLERİ

- **Gömülü yazılım**:
  Denetim makineleri ve bilgisayar sayılmayan aygıtlar için yazılmış yazılımlardır.
- **Uygulama Yazılımı:**
  Bilgisayar uygulaması, bilgisayarların çeşitli işlerde kullanılmasını sağlayan, belirli bir bilgisayar mimarisi (i386, PowerPC, Motorola 680x0 vs.) için uygulama geliştirme dilleri (C/C++, Perl, Python, Java vs.) aracılığı ile hazırlanan yazılımdır.

YAZILIM GELİŞTİRME ADIMLARI

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

Çözümleme

Bir şeyi anlayabilmek için parçalarına ayırmak gereklidir. Gerçeklenecek sistemi anlamaya yönelik çalışmalardan ve üst düzey planlama eylemlerinden oluşur.

Tasarım

- Bir araştırma ve/veya geliştirme sürecinin çeşitli dönemlerinde izlenecek yol ve işlemleri tasarlayan çerçeve.
- Çözümleme ile anlaşılan sorun tasarım aşamasında kağıt üzerinde çözülür.

Gerçekleme

- Eldeki tasarım, bir programlama dili ile kodlanır.

Sınama

- Test altında hizmetlerin veya ürünlerin kalitesi hakkında paydaşlara bilgi sağlamak için yürütülen bir araştırmadır.
- Yazılım uygulamalarının risklerini anlamak için yazılımı bağımsız ve nesnel olarak incelemektir.

Bakım

- Yazılımın faaliyete geçirilmesinden sonra sistemde yapılan değişikliklerdir.

YAZILIM GELİŞTİRME MODELLERİ

WATERFALL

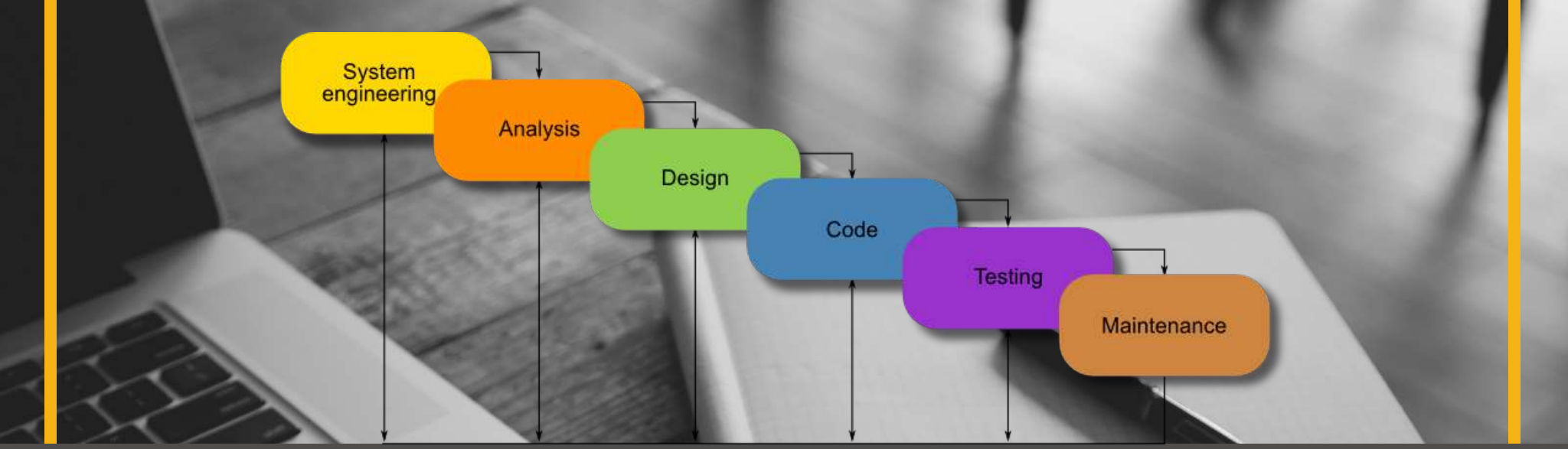
V-SHAPE

AGILE

SPIRAL

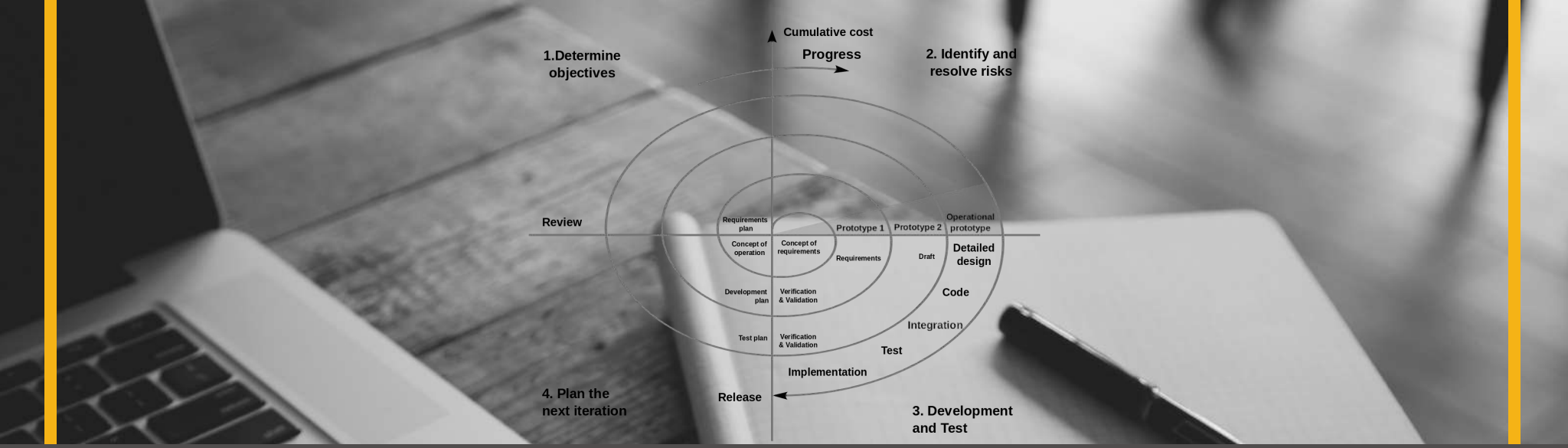
ITERATIVE

YILDIZ TEKNİK ÜNİVERSİTESİ



WATERFALL

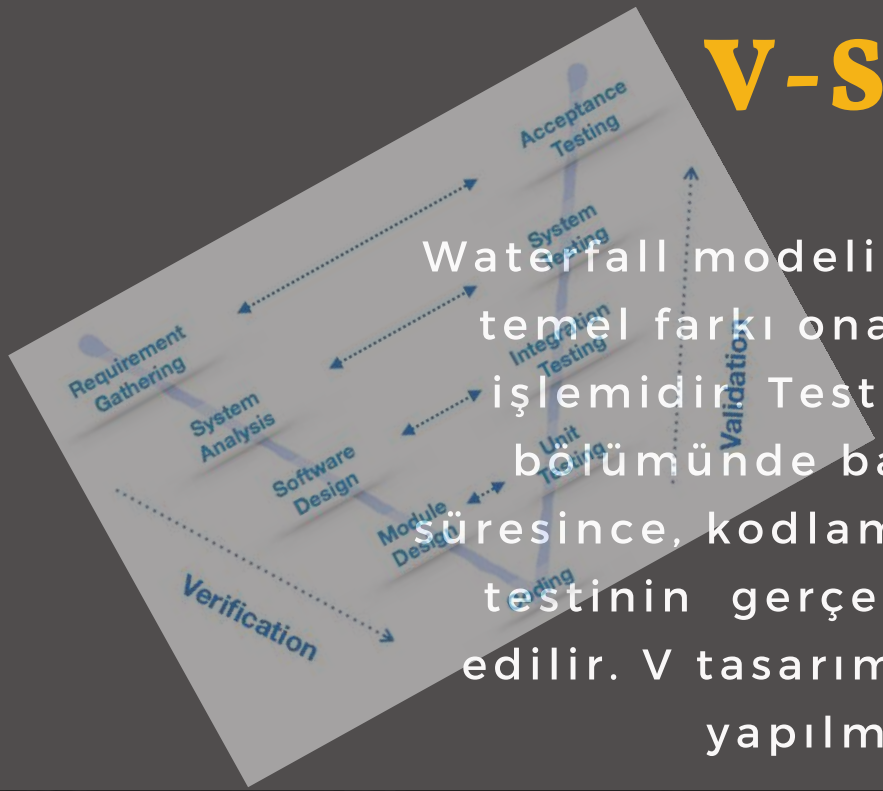
Bu model kullanılarak geliştirilen uygulamalarda, her bir bölüm ardışık olarak yapılır, her bölümden sonra gerçekleştirilen bölümün sonuçları gösterilir. Fakat waterfall model uzun süreli projeler için uygun olmamakla beraber,esneklik sağlamamaktadır.



SPIRAL

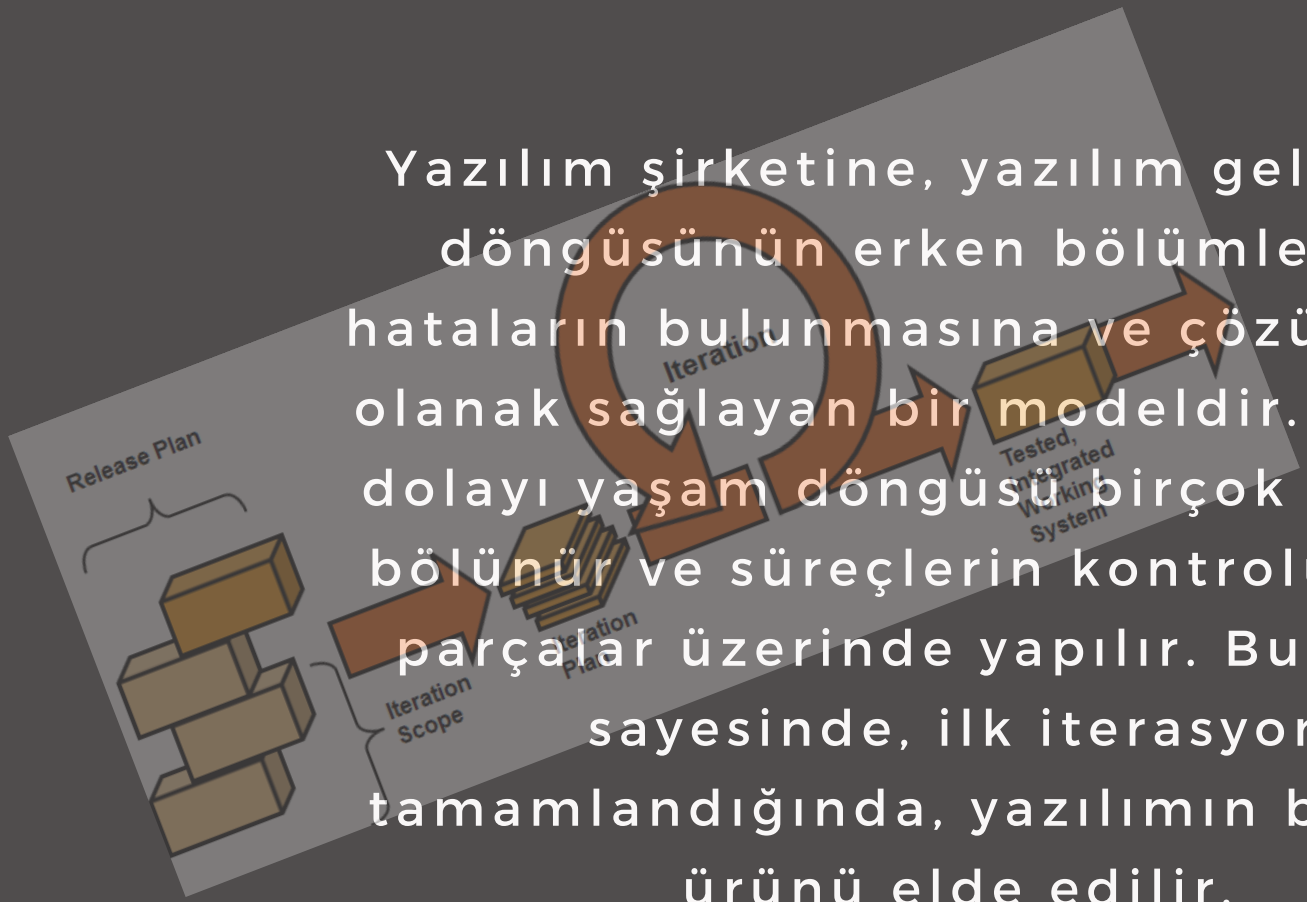
Bu modelin temelinde, yazılım geliştirme süreci boyunca risk analizi önemlidir. Bu modelde,her bir klasik waterfall modeli, çok sayıda iterasyona bölünür ve her iterasyonda planlamayı ve risk analizini inceler.

V-SHAPE



Waterfall modeline çok benzemektedir, temel farkı onaylama sahası ve test işlemidir. Testlere dökümantasyon bölümünde başlanır, integrasyon süresince, kodlamada ve yazılım ürünün testinin gerçekleştiriminde devam edilir. V tasarımı, ileriye yönelik test yapılmasını sağlar.

ITERATIVE



Yazılım şirketine, yazılım geliştirme döngüsünün erken bölümlerinde hataların bulunmasına ve çözülmesine olanak sağlayan bir modeldir. Bundan dolayı yaşam döngüsü birçok parçaya bölünür ve süreçlerin kontrolü küçük parçalar üzerinde yapılır. Bu model sayesinde, ilk iterasyon tamamlandığında, yazılımın basit bir ürünü elde edilir.

AGILE

Temel olarak iterative modele benzer, insan faktörünü kullanarak geliştirme sağlar. Geliştirme süreci boyunca, yazılım takımının geri dönüşlerinden yararlanılır.

YILDIZ TEKNİK ÜNİVERSİTESİ

TESTLER

- **YAKLAŞIM TARZLARINA GÖRE:**

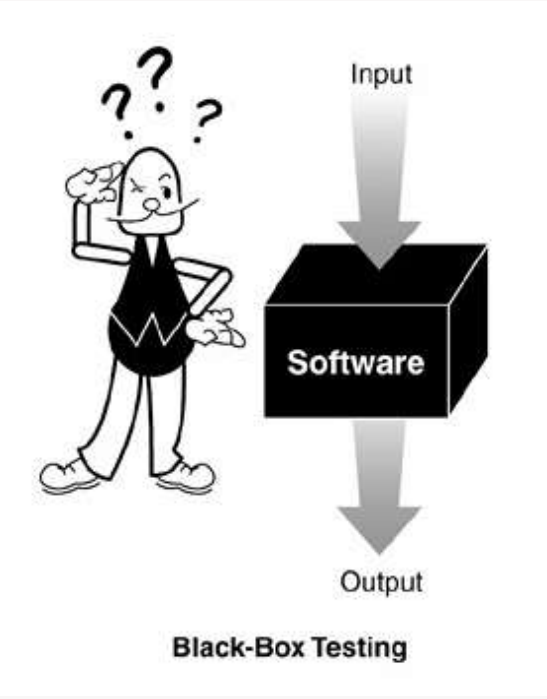
  BLACK-BOX TEST
  WHITE-BOX TEST

- **YÜRÜTÜLME SIRALARINA GÖRE:**

  VERIFICATION TESTS
  VALIDATION TESTS
  - ALFA TEST
  - BETA TEST

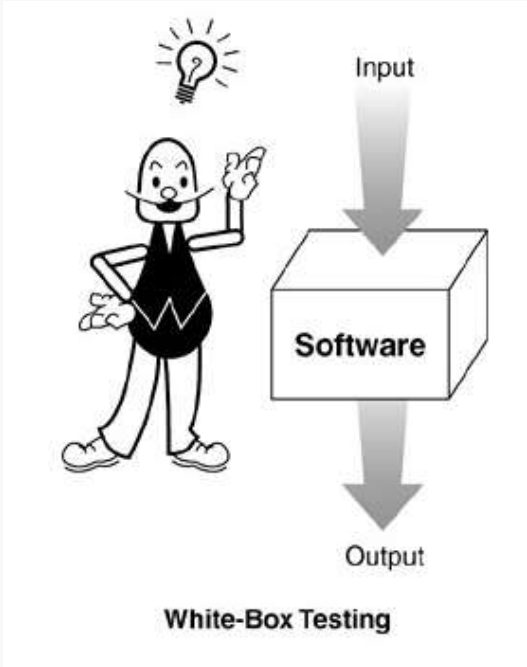
BLACK-BOX TEST

Sınanacak birimin iç işleyişi bilinmez, sadece birimin beklenen girdilere karşı beklenen çıktıları üretip üretilmediğine bakılır.



WHITE-BOX TEST

Sınanacak birimin iç işleyişi bilinmez, sadece birimin beklenen girdilere karşı beklenen çıktıları üretip üretilmediğine bakılır.



White-Box Testing

VERIFICATION TESTS

Yazılım ekibi tarafından yapılan testlerdir.Ürünü kullanacak kişilerin isteklerinin karşılanıp karşılanmadığını test eden etkinliklerdendir.

- **Unit Test**: Yazılımın en küçük bileşeninin sınanmasıdır.
- **Integration Test**: Unit testi geçmiş işlevlerin (birim) birlikte doğru bir şekilde çalışmasını test eder.

VALIDATION TESTS

Son kullanıcı tarafından yapılan testlerdir. Geçerleme ise ürünün içsel niteliğine ilişkin izleme ve denetim etkinliklerinden oluşur.

- **Alpha Test**: Yazılım ekibinin denetiminde kullanıcılar tarafından yapılan testlerdir.
- **Beta Test**: Kullanıcıların bağımsız bir şekilde yaptığı testlerdir. Hatalar düzenli aralıklarla yazılım ekibine bildirilir.


DİNLEDİĞİNİZ İÇİN
TEŞEKKÜRLER!