

YAZILIM TEST MÜHENDİSLİĞİ

Dr. Öğr. Üyesi Hasan YETİŞ

Yazılım Test Teknikleri

- Yazılım test tekniklerinin belirlenmesindeki en ayırt edici unsur, yazılım modülünün iç detayının bilinip bilinmeyeceğidir.



Kara Kutu Testi

- Kara kutu test tekniğinde yazılım kodlarına ve içeriğine bakılmadan yazılımın işlevselliği değerlendirilir.
- Dikkat edilen en önemli husus, çalıştırılan programın doğru sonuç vermesi ve arayüzler olarak da kullanıcı beklentisini karşılamasıdır.
- Kara kutu testinde ana odak noktası, bir bütün olarak sistemin işlevselliğinin değerlendirilmesidir.
- Test edilen uygulama kullanıcının bakış açısıyla kontrol edilip onaylanır. Bu da Kara kutu test yönteminin kullanıcı gereksinimlerine dayandığının kanıtıdır.

Kara Kutu Testi

- Kara kutu test yöntemi kullanılarak aşağıdaki hususlar kontrol edilir:
 - Kullanıcılar tarafından gerçekleştirilen doğru eylemler
 - Sistemin girdilerle etkileşimi
 - Sistemin etkileşimlere yanıt verme süresi
 - Kullanıcı fiziksel arayüzünde olabilecek sorunlar
 - Yazılımın performans durumları
 - Yazılımda olabilecek beklenmedik uygulama hataları

Kara Kutu Testi

- Kara kutu testi ile gerçekleştirilen test çeşitlerinde, uygulamanın gereksinim ve özelliklerine odaklanıldığı için, *Şartname Tabanlı Test*, *Girdi-Çıktı Testi*, *Kapalı kutu Testi* gibi isimlerle de bilinir.
- Kara kutu testleri yazılımın kod kısmıyla değil, doğru çalışması ve doğru sonuç vermesi ile ilgilenmektedir. Dolayısı ile yazılımın çalışmasının test edilmesi için örnek giriş değerlerinin belirlenmesi gerekmektedir.
- Örnek giriş değerleri belirlenirken farklı sonuçlara sebep olacak giriş değerlerinin seçilmesi testin daha verimli bir şekilde gerçekleştirilmesine sebep olacaktır.

Kara Kutu Testi

- Kara kutu testlerinde, test verilerinin seçilmesi için farklı yaklaşımlar bulunmaktadır.
 - Eşit Bölmelere Ayırma (Equivalence Partitioning)
 - Sınır Değerleri Analizi (Boundary Value Analysis)
 - Karar Tablosu (Decision Table)
 - Durum Geçiş Tablosu (State Transition Table)
- Kullanım Durumları Testi (Use Case Testing)

Örnek Senaryo

- **Kullanılacak olan örnek senaryo kullanıcı girişi olacaktır. Senaryoya göre eşleşen kullanıcı adı ve şifre ile giriş yapan kullanıcıların sisteme başarılı bir şekilde girebilmesi için e-Posta onayı alması gerekmektedir.**
- Başarılı bir şekilde giriş yapan kullanıcıların, daha önceden yapmış oldukları giriş sayısına göre yeni üye, sadık üye ve profesyonel üye gibi farklı sayfalara yönlendirilmesi sağlanacaktır.
- 0-1.000 kez giriş yapan kullanıcılar standart üye, 1.000-10.000 arası giriş yapanlar sadık üye ve 10.000-üzeri giriş yapanların kadim üye sayfalarına yönlendirilmesi gereksin.
- Art arda 3 kez hatalı şifre giren kullanıcıların üyelikleri bloke olacaktır.

Kara Kutu Testi

- **Eşit Bölümlere Ayırma**
- Bu tekniğin esas amaçlarından bir diğeri de benzer çıktıyı üretebilecek durumları yok ederek, gereksiz senaryoların azaltılmasıdır.

Tablo 4.1. Farklı denklik sınıflarına ait test verileri

Kullanıcı Adı	Şifre	Girilen Şifre	e-Posta Onay
Kullanıcı1	123	123	1
Kullanıcı2	123	111	1
Kullanıcı3	123	123	0
Kullanıcı4	123	112	0

Örnek Senaryo

- Kullanılacak olan örnek senaryo kullanıcı girişi olacaktır. Senaryoya göre eşleşen kullanıcı adı ve şifre ile giriş yapan kullanıcıların sisteme başarılı bir şekilde girebilmesi için e-Posta onayı alması gerekmektedir.
- Başarılı bir şekilde giriş yapan kullanıcıların, daha önceden yapmış oldukları giriş sayısına göre yeni üye, sadık üye ve profesyonel üye gibi farklı sayfalara yönlendirilmesi sağlanacaktır.
- **0-1.000 kez giriş yapan kullanıcılar standart üye, 1.000-10.000 arası giriş yapanlar sadık üye ve 10.000-üzeri giriş yapanların kadim üye sayfalarına yönlendirilmesi gereksin.**
- Art arda 3 kez hatalı şifre giren kullanıcıların üyelikleri bloke olacaktır.

Kara Kutu Testi

- **Sınır Değerleri Analizi**
- Giriş verilerinin sınırlarından kaynaklanan kusurları veya hataları tanımlamak için kullanılan yöntemdir.

Tablo 4.2. Farklı sınır değerlerine ait test verileri

Kullanıcı Adı	Şifre	Girilen Şifre	Toplam Giriş Sayısı	e-Posta Onay
Kullanıcı1	123	123	999	1
Kullanıcı2	123	123	1.000	1
Kullanıcı3	123	123	1.001	1
Kullanıcı4	123	123	9.999	1
Kullanıcı5	123	123	10.000	1
Kullanıcı6	123	123	10.001	1

Kara Kutu Testi

- **Karar Tablosu**
- Bir tablo ile bu test senaryoları ve çıktıları arasındaki sonuçlar belirtilir. Her kurala özgü bir test senaryosu tasarlanır.

Tablo 4.3. Karar tablosu tabanlı değerlendirmede oluşacak ihtimaller

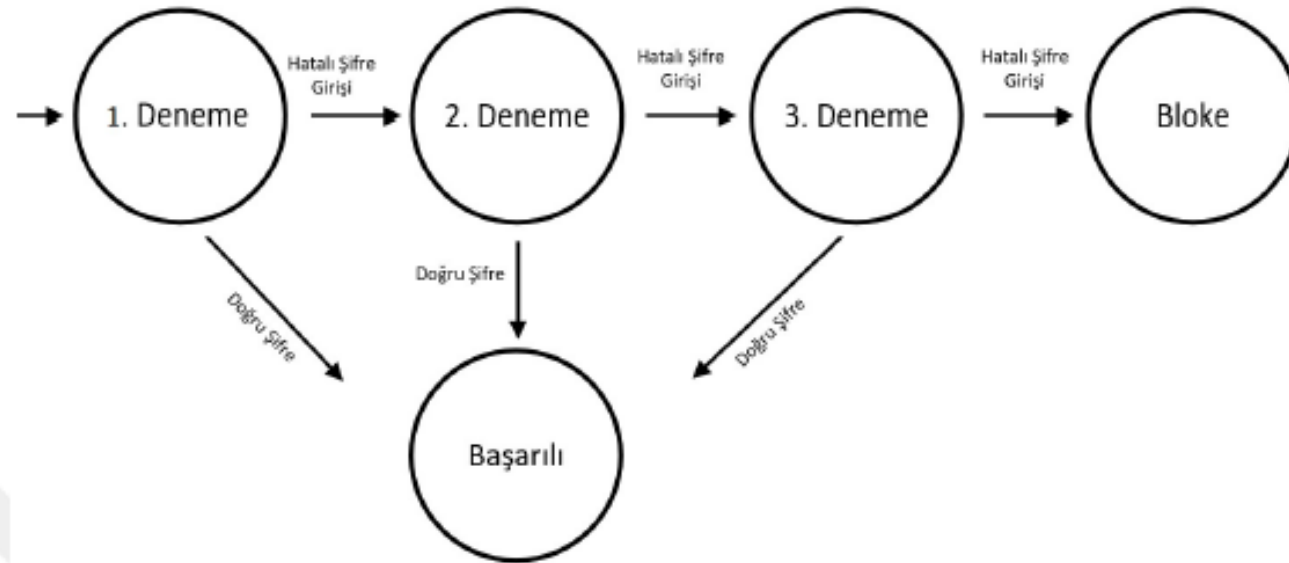
Eşleşen Kullanıcı Adı-Şifre	Onaylı e-Posta Adresi	Toplam Giriş Sayısı	Sonuç
E	E	0-1.000	Standart
E	E	1.000-10.000	Sadık
E	E	10.000+	Kadim
E	H	0-1.000	Onay
E	H	1.000-10.000	Onay
E	H	10.000+	Onay
H	E	0-1.000	Hata
H	E	1.000-10.000	Hata
H	E	10.000+	Hata
H	H	0-1.000	Hata
H	H	1.000-10.000	Hata
H	H	10.000+	Hata

Örnek Senaryo

- Kullanılacak olan örnek senaryo kullanıcı girişi olacaktır. Senaryoya göre eşleşen kullanıcı adı ve şifre ile giriş yapan kullanıcıların sisteme başarılı bir şekilde girebilmesi için e-Posta onayı alması gerekmektedir.
- Başarılı bir şekilde giriş yapan kullanıcıların, daha önceden yapmış oldukları giriş sayısına göre yeni üye, sadık üye ve profesyonel üye gibi farklı sayfalara yönlendirilmesi sağlanacaktır.
- 0-1.000 kez giriş yapan kullanıcılar standart üye, 1.000-10.000 arası giriş yapanlar sadık üye ve 10.000-üzeri giriş yapanların kadim üye sayfalarına yönlendirilmesi gereksin.
- **Art arda 3 kez hatalı şifre giren kullanıcıların üyelikleri bloke olacaktır.**

Kara Kutu Testi

- **Durum Geçiş Tablosu**
- Sistemin durumu koşullara veya olaylara bağlı olarak değiştiğinde uygulanması gerekir



Şekil 4.2. Kullanıcı girişi için durum geçiş diyagramı

Kara Kutu Testi

- Durum Geçiş Tablosu

Tablo 4.4. Farklı durum geçişlerinin testi için kullanılacak örnek test verileri

Kullanıcı Adı	Şifre	Girilen Şifre	Hatalı Giriş Sayısı	e-Posta Onay
Kullanıcı1	123	123	0	1
Kullanıcı2	123	111	0	1
Kullanıcı3	123	123	1	1
Kullanıcı4	123	111	1	1
Kullanıcı5	123	123	2	1
Kullanıcı6	123	111	2	1

Kara Kutu Testi

- **Kullanım Durumları Testi (Use-Case Testing)**
- Temel olarak test yöntemi bir kullanıcı ve bu kullanıcının sistemden beklediği durumların oluşup oluşmadığını ölçümler.
- Kullanım durumları tekniğini verilen senaryo üzerinde örneklendirecek olursak, kullanıcının giriş yapması için kullanıcı adı ve şifre alanlarını doldurmuş olması gerekir.
 - kullanıcı adının girilmemesi,
 - şifrenin girilmemesi,
 - hem kullanıcı adı hem şifrenin girilmemesi

Beyaz Kutu Testi

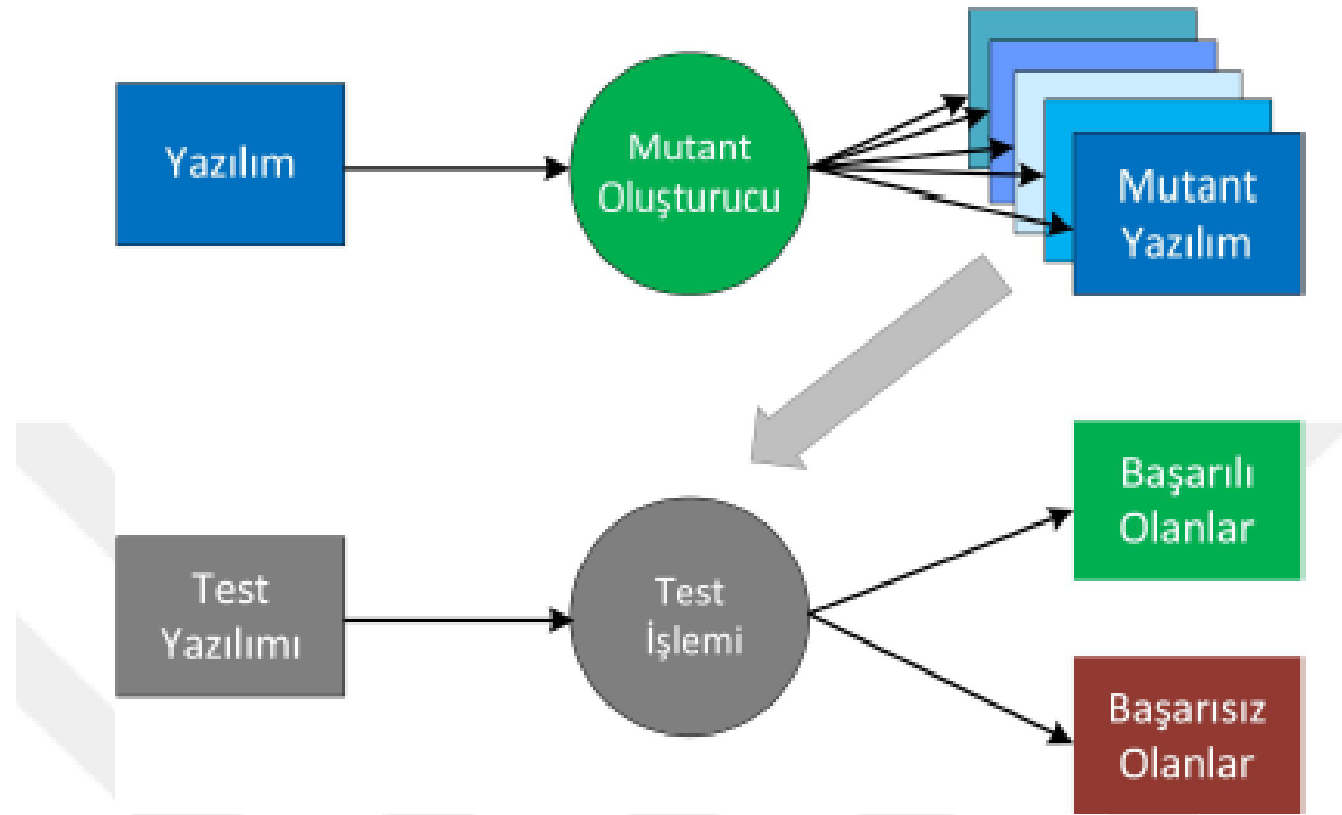
- Genellikle birim ve entegrasyon test seviyesinde etkin kullanılan bu test, literatürde ***Cam kutu, Saydam kutu, Şeffaf kutu, Filigran Test*** veya ***Yapısal Test*** adlarıyla da bilinir.,
- Bu tekniğin, temel amacı kodun iç yapısı incelemek ve doğrulamaktır
- Bu test tekniği ile bir kod bloğundaki
 - güvenlik açıkları,
 - kodlamadaki hatalı ya da eksik yapılandırılmış döngüler,
 - gereksiz veya kısır döngüler,
 - koşullu döngülerin işlevselliğigibi kriterler kontrol edilir.

Beyaz Kutu Testi

- Beyaz kutu test teknikleri, kara kutu testinde olduğu gibi sadece işlevsellikten ziyade **iç yapıları, kullanılan veri yapılarını, iç tasarımı, kod yapısını ve yazılımın çalışmasını** analiz eder.
- Kod optimize edilir
- Fazladan kod satırlarının kaldırılmasına yardımcı olur.
- Daha erken bir aşamada başlayabilir.
- Test edenlerin, kara kutu testinin aksine kod ve programlama dili hakkında derinlemesine bilgi sahibi olmaları gerekir.
- Mevcut kod test edildiğinden eksik işlevler tespit edilemez.

Beyaz Kutu Testi

- **Mutasyon Testi**
- Kodda bilinçli hatalar oluşturur. (*Arıza Temelli Test Stratejisi*)
- Test uzmanları tarafından gerçekleştirilen bu yöntemde, mutant adı verilen sürümlerle küçük hatalı kodlar oluşturulur.
- Her varyasyon bir çeşit hatayı içermelidir.
- Geliştirilen test senaryoları orijinal ve mutant programa ayrı ayrı uygulanarak hatalı kodun test senaryosuna karşı davranışı analiz edilir.
- Asıl amaç, geliştirilen test senaryoları ile hataların bulunup bulunamayacağını doğrulamaktır
- Mutasyon testi, yazılım test senaryolarının doğruluğunu kanıtlamayı amaçlar. Farklı bir deyişle testin test edilmesidir.



Şekil 4.3. Mutasyon testi

Beyaz Kutu Testi

- **Mutasyon Testi**
- *Değer Mutasyonları:* Programdaki hataların tespiti için değerlerin değiştirildiği mutasyon testi yöntemidir. Temelde sabit olarak kullanılan değerler daha küçük veya daha büyük olarak değiştirilir.
- *Karar Mutasyonları:* Aritmetiksel ve mantıksal operatörlerin yanında ilişkisel operatörlerle ilgili var olan hataların tespitinde kullanılır.
- *İfade Mutasyonları:* Sistemde bulunan kod bloklarının silinmesi veya farklı bir ifadeyle değiştirilmesi durumunda oluşan hataların tespitinde kullanılır.