

BİM539,  
Ders 4: Test Planı ve  
Traceability Matrix  
(İzlenebilirlik Matrisi)

Gereksinimlere sahipsiniz ve  
hataları arıyorsunuz...

Nasıl?

Test planı  
geliştirerek!

- Resmi veya resmi olmayan gereksinim olabilir.
- Neyi test edeceksiniz? Sorumluluğunuz nedir?  
Takip etmek gerekli midir?

# Neyi test edeceksiniz?

- Throw-away script?
- Geliştirme aracını?
- Dahili bir web sayfasını?
- Girişimci bir uygulamayı?
- Ticari bir uygulamayı?
- İşletim sistemini?
- Aviyonik uygulamayı?

# Test konsept bağımlı bir eylemdir

- Nasıl test edeceksin
- Ne kadar test edeceksin
- Hangi araçları kullanacaksın
- Hangi belgeleri sağlayacaksınız
- ...Bütün bunların tamamı yazılımın konseptine göre değişkenlik gösterebilir.

# Test Planları ve Test Senaryoları

- Test, bir test planının işletilmesi ile gerçekleştirilir.
- *Test planı*: Birlikte çalıştırılacak ilişkili test senaryolarından oluşan bir listedir.
- *Test senaryosu*: Bireysel davranışa sahip en küçük test plan parçasıdır.
  - Neyin, nasıl test edileceği tanımlanır
  - Beklenen davranış tanımlanır

# Bir test senaryosu temel olarak şu bileşenlerden oluşur:

- *Öncül Koşul*: Test başlamadan önce sistemin durumu
  - Environment / global değişken değerleri, ...
  - Ekran durumu, veritabanı durumu, ...
- *İşlem Adımları*: Son koşulları elde etmek için uygulanacak adımlar
- *Son Koşul*: Sistemi test ettikten sonra **BEKLENEN** durum
  - Environment / global değişkenler atanır, ...
  - Çıktı ekrana yazdırılır, paket network ile gönderilir, ...

IEEE 829, "Standard for Software Test Documentation", kaynaklar/IEEE829.pdf'a bakılabilir



# Örnek

Alışveriş sepeti boşken, "Satın Al" butonuna tıkladığımda sepetteki nesne sayısı 1 olmalıdır.

Ön koşul: Boş alışveriş sepeti

İşlem Adımları: "Satın Al" butonuna tıklamak

Son Koşul: Alışveriş sepeti 1 nesne gösterir

# Ayrıca şöyle bir şey eklenebilir:

- *Tanımlayıcı*: Test senaryosu tanımlama yolu
  - Sayı olabilir
  - Genellikle bir etiketi bulunur, ör: INVALID-PASSWORD-THREE-TIMES-TEST
- *Test Senaryosu*: Test senaryosu tanımı

# Birim test yapmak istersek, aşağıdakileri de eklemeliyiz:

- *Girdi değerleri*: Metot parametresi olarak geçilecek değerler
- *Çıktı değerleri*: Metottan dönecek **beklenen** değer/değerler
- Girdi değerleri ve ön koşul arasındaki farklılıklar
  - Metodu etkileyen parametreler dışındaki her şey ön koşuldur
    - Metot tarafından okunan global değişkenler
    - Metot tarafından okunan dosyanın içeriği
- Çıktı değerleri ile son koşul arasındaki farklılıklar
  - Metodun etkilediği çıktı değerleri hariç her şey son koşuldur
    - Metot tarafından değiştirilen global değişken
    - Metot tarafından değiştirilmiş dosya içeriği

# Örnek

`SORT_ASCENDING` işaretlendiğinde, `[9,3,4,2]` parametresi ile sıralama fonksiyonu çalışır ve küçükten büyüğe sıralanmış yeni dizi döndürür: `[2,3,4,9]`.

**Ön koşul:** `SORT_ASCENDING` işaretleme

**Girdi değerleri:** Array `[9,3,4,2]`

**İşlem adımları:** sort metodunu girdi değerleri ile çağır

**Çıktı değeri:** Array `[2,3,4,9]`

**Son koşul:** None

# Bir başka örnek

```
int print_hello_world() {  
    System.out.print("Hello World");  
    return 1;  
}
```

- Yukarıdaki metot için test senaryosu yazacağımızı varsayalım:
  - Çıktı değerleri nedir?
  - Son koşul ne olmalıdır??
  - Çıktı değeri: 1
  - Son koşul: Hello World ekrana basılır

# Tam olarak bir test senaryosu aşağıdaki bileşenleri içerir

- Tanımlayıcı
- Test senaryosu
- Ön koşul
- Girdi değerleri
- İşlem adımları
- Çıktı değerleri
- Son koşul

IEEE 829, "Standard for Software Test Documentation", [kaynaklar/IEEE829.pdf](#)

# Test Planı

- Alt sistem veya bir özelliğe ait test senaryoları listesi
- Örnekler:
  - Veritabanı bağlantı test planı
  - Pop-up uyarıları test planı
  - Basınç güvenlik kilidi test planı
  - Sistem kapatma test planı

# Basınç güvenlik kiliti test planı

LOW-PRESSURE-TEST

HIGH-PRESSURE-TEST

SAFETY-LIGHT-TEST

SAFETY-LIGHT-OFF-TEST

RESET-SWITCH-TEST

RESET-SWITCH2-TEST

FAST-MOVEMENT-TEST

RAPID-CHANGE-TEST

GRADUAL-CHANGE-TEST

MEDIAN-PRESSURE-TEST

LIGHT-FAILURE-TEST

SENSOR-FAILURE-TEST

SENSOR-INVALID-TEST



# Bir grup test planı ise *test suite (test paketi)* oluşturur...

- Regresyon Test Paketi
  - Basınç Güvenlik Regresyon Test Planı
  - Güç Dengeleme Regresyon Test Planı
  - Su Akış Regresyon Test Planı
  - Kontrol Akış Regresyon Test Planı
  - Güvenlik Regresyon Test Planı

# Test Paketi Oluşturma...

- top-down olarak başlayın
  - Sistemi özelliklere göre bölmenin en iyi yolu nedir?
- Belirli bir özellik için (test planı), hangi açılardan test etmek istiyorum belirle
- Her bir bakış açısı için, denklik sınıflarını/edge veya/corner sınıfların vs. hangi test senaryoları ile kapsanacağını belirle
- Her bir test senaryosu birbirinden bağımsız ve tekrarlanabilir olmalıdır.
  - Bağımsız: Bir test senaryosu bir başka senaryonun çalışmasını etkilememelidir
  - Tekrarlanabilir: Ön koşul + çalıştırma adımları her zaman son koşulu oluşturmalıdır

# Test Run - Testleri Yürütme

- *Test run*: Test plan veya test paketinin yürütülmesidir.
- Bir test senaryosu bağımsız olmaz ise ne olur?
  - Test senaryosunun davranışı daha öncesinde bir başka test senaryosunu çalıştırıp çalıştırmamanıza bağlı olarak değişebilir
- Test senaryosu tekrarlanabilir olmaz ise ne olur?
  - Aynı girdi ve işlem adımları ile farklı davranışlar gözlemlenebilir
- Büyük ihtimalle her ikisi de ön koşulların oluşturulmaması sebebiyle gerçekleşir
  - OS durumu, DB durumu, dosya sistemi durumu, hafıza durumu, ...

# Ön Koşul Kurulum Scripti

- Bir test planı belli bir ön koşula bağımlı olabilir
  - Bu durumda test planı yürütülmeden ön koşulun ayarlanması gerekir
  - Bir test senaryosu ön koşulu değiştirirse, testin ortasında tekrar aynı koşulun oluşturulması gerekir
  - Ön koşulu oluşturmak çoğu zaman gereksiz görülen ve yorucu bir iştir
- *Ön Koşul Kurulum Scripti*
  - Makinaya bir uygulamanın kurulumu/tekrar kurulumu olabilir
  - Test için uygun veritabanı girdilerinin oluşturulması olabilir
  - Hafıza veri ile başlatılabilir
  - Donanım cihazının tekrar başlatılması olabilir



# Test Koşmasından Sonraki Durum

- Test koşması sırasında test eden kişi manuel olarak veya otomatik olarak her bir test senaryosunu çalıştırır ve her biri için bir durum ataması yapar
- Olası Durumlar
  - PASSED – Beklenen değer ile sonuçlandı
  - FAILED – Sonuçlandı ancak beklenen değer değil
  - PAUSED – Çalıştırma sırasında test durduruldu
  - RUNNING – Test çalıştırılıyor
  - BLOCKED – Ön koşul sağlanmadığı için test tamamlanamadı
  - ERROR – Test çalıştırılmasında problem

# Hatalar

- Bir test senaryosu hata ile karşılaşılsa bu hata dosyalanmalıdır
  - Aynı hataları tekrar tekrar dosyalamaya gerek yoktur
- Bir sonraki derste bu hataların dosyalanması üzerine konuşacağız

# Traceability Matrix (İzlenebilirlik Matrisi)

- Ele alalım:
  - Bir test senaryosu birden fazla gereksinimi test ediyor olabilir
  - Bir gereksinim birden fazla test senaryosu ile test ediliyor olabilir
  - Bu many-to-many türünde karmaşık bir ilişkidir!
  - Neyin ne kadar test edildiği nasıl takip edilir?
- **İzlenebilirlik Matrisi:** gereksinimler ve test senaryoları arasındaki ilişkiyi tanımlayan tablodur
  - Yazılım geliştirme süresince gereksinimler nasıl uygulanır?
  - Test kapsamının eksikliği veya gereksiz testleri belirlemede yardımcı olur

# İyi İzlenebilirlik Matrisi Örneği

GER1: TEST\_CASE\_1, TEST\_CASE\_2

GER2: TEST\_CASE\_3

GER3: TEST\_CASE\_4, TEST\_CASE\_7

GER4: TEST\_CASE\_5, TEST\_CASE\_9

GER5: TEST\_CASE\_6, TEST\_CASE\_10

- *Tüm gereksinimler kendileri ile ilişkili en az bir test senaryosuna sahiptir ve tüm test senaryoları bir gereksinim ile eşlenir*



# Problemli İzlenebilirlik Matrisi 1

GER1: TEST\_CASE\_1, TEST\_CASE\_2

GER2:

GER3: TEST\_CASE\_4, TEST\_CASE\_7

GER4: TEST\_CASE\_5, TEST\_CASE\_9

GER5: TEST\_CASE\_6, TEST\_CASE\_10

- *Gereksinim 2 için test senaryosu yok!*

# Problemlı İzlenebilirlik Matrisi 2

GER1: TEST\_CASE\_1, TEST\_CASE\_2

GER2: TEST\_CASE\_3

GER3: TEST\_CASE\_4, TEST\_CASE\_7

GER4: TEST\_CASE\_5, TEST\_CASE\_9

GER5: TEST\_CASE\_6, TEST\_CASE\_10

?????: TEST\_CASE\_11

- *Test senaryosu 11 neyi test ediyor?*

# Uygulamalı İzlenebilirlik Matrisi



# Textbook Chapters 6 ve 8'i okuyunuz

- Daha ileri bir şeyler okumak isterseniz:

**IEEE Standard for Software Test Documentation (IEEE 829-2008)**

- kaynaklar/IEEE829.pdf içerisinde bulunmaktadır.