

Mühendislik Projesi Engineering Project

Hazırlayan: M. Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

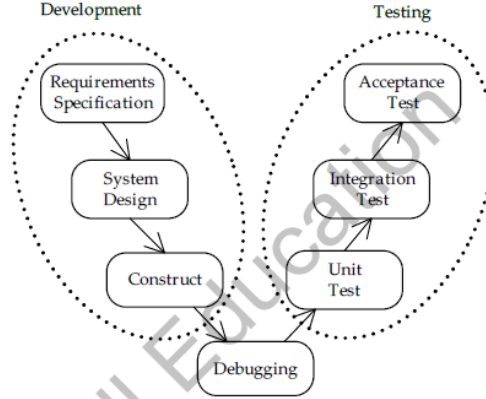
Bu dersin sunumları, "Ralph M. Ford, Chris S. Coulston, Design for Electrical and Computer Engineers, McGraw Hill, 2008." kitabı kullanılarak hazırlanmıştır.

İçerik

- ▶ Test prensipleri
- ▶ Test oluşturma
 - ▶ Debugging
 - ▶ Unit testing
 - ▶ Matrix tests
 - ▶ Step-by-step tests
 - ▶ Automated test scripts
 - ▶ Integration testing
 - ▶ Acceptance testing

Test prensipleri

- ▶ Geliştirme sürecinde ve müşteriye sunulmadan önce **sistemler test edilir.**
- ▶ Test ile **hataları erken görmek** ve **düzeltilmek amaçlanır.**
- ▶ Yaygın kullanılan test modelinde (**V model**) **tasarım süreçleri sol kısımda, test süreçleri sağ kısımda** yer almaktadır.



3

Test prensipleri

- ▶ Test senaryolarının aşağıdakileri sağlaması gereklidir:
 - ▶ **Hangi modülün ne iş yapacağı kesin olarak tanımlanmalıdır.**
 - ▶ Testler geliştiriciler için **erken geri bildirim sağlamalıdır.**
 - ▶ Testler tasarımcıları **uç çalışma durumları için düşünmeye zorlamalıdır.**
 - ▶ **Test senaryoları dokümante edilmelidir.**
- ▶ Sistemlerin **gereksinimleri karşılayıp karşılamadığının test edilmesi gereklidir.**
- ▶ Test dokümanları sistemin ne iş yapacağı ve bileşenleri hakkında bilgileri içermelidir.

4

Test prensipleri

Test etme, gözlemlenebilirlik ve kontrol edilebilirlik

- ▶ Testler iki gruba ayrılır:
 - ▶ Kara kutu (black box)
 - ▶ Beyaz kutu (white box)
- ▶ **Black box testler**, sistem iç organizasyonu hakkında bilgi olmadan yapılır.
- ▶ Black box testte, **sistemin girişleri değiştirilir ve elde edilen çıkış ile beklenen çıkış karşılaştırılır.**
- ▶ **White box test**, sistemin iç çalışma bilgisi kullanılarak yapılır.
- ▶ White box test için **sistemin iç düğümleri ve bileşenleri için testler oluşturulur.**
- ▶ White box test ile **tüm düğümler, sınır ve geçersiz değerler için test edilir.**

5

Test prensipleri

Test, gözlemlenebilirlik ve kontrol edilebilirlik

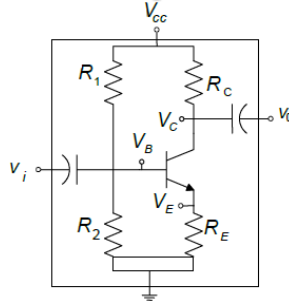
- ▶ **Test edilebilirlik**, bir sistemde oluşan **hatanın kısa sürede belirlenmesini** ve yerinin bulunmasını ifade eder.
- ▶ **Kontrol edilebilirlik**, sistemdeki herhangi **bir düğümü veya bileşeni önceden tanımlı değere set edebilmeyi** ifade eder.
- ▶ **Gözlemlenebilirlik**, sistemdeki herhangi **bir düğümün veya bileşenin gözlemlenebilir olmasını** ifade eder.
- ▶ **Black box** testte, **kontrol edilebilirlik ve gözlemlenebilirlik düşüktür.**

6

Test prensipleri

Örnek

- ▶ Aşağıdaki basit yükselteç devresi girişteki v_i değerini yükselterek $v_0 = A.v_i$ değerini üretir.
- ▶ **Dikdörtgen** sistemin **black box** görünümünü ifade eder.
- ▶ **Giriş değiştirilerek çıkış gözlemlenir.**
- ▶ White box test ile sistemin iç bileşenleri de test edilir.
- ▶ **White box test ile V_B , V_C , V_E değerleri de kontrol edilebilir.**



7

İçerik

- ▶ Test prensipleri
- ▶ Test oluşturma
 - ▶ Debugging
 - ▶ Unit testing
 - ▶ Matrix tests
 - ▶ Step-by-step tests
 - ▶ Automated test scripts
 - ▶ Integration testing
 - ▶ Acceptance testing

8

Test oluřturma

Debugging

- ▶ Sistemdeki **bir bileřen** nadiren de olsa **beklendiđi gibi çalışmayabilir (bug)**.
- ▶ **Debugging, problemin nedenini bulma ve düzeltme için gereklidir.**
- ▶ İki tür bug vardır:
 - ▶ **Bohrbugs:** Her zaman **aynı yerde** ve **aynı girişle ortaya çıkar.** Düzeltilmesi kolaydır.
 - ▶ **Heisenbugs:** Aynı girişle **aynı hata elde edilemeyebilir.** Hatanın yerini belirlemek zordur.

9

Test oluřturma

Debugging

- ▶ Debugging süreci iteratiftir ve ařađıdaki adımlardan oluşur:
 - ▶ **Farklı çalışma şartlarında** problem gözlemlenir.
 - ▶ **Potansiyel problemin ne olduđu tahmin edilir.**
 - ▶ Tahmin edilen **problemin nedeni ortadan kaldırılır.**
 - ▶ Problem giderilene kadar **test tekrar edilir.**

10

İçerik

- ▶ Test prensipleri
- ▶ Test oluşturma
 - ▶ Debugging
 - ▶ **Unit testing**
 - ▶ Matrix tests
 - ▶ Step-by-step tests
 - ▶ Automated test scripts
 - ▶ Integration testing
 - ▶ Acceptance testing

11

Test oluşturma

Unit testing

- ▶ Birim testi ile **bir modülün tüm işlevleri test edilir.**
- ▶ Aşağıdaki örnekte `if` ifadesi ile `else` ifadesinin **olası tüm değerler için test edilmesi gereklidir.**

```
if (16 < input < 32)
    output = ROM[input - 16];
else
    output = (2 * input) + 32;
```

12

İçerik

- ▶ Test prensipleri
- ▶ Test oluşturma
 - ▶ Debugging
 - ▶ Unit testing
 - ▶ **Matrix tests**
 - ▶ Step-by-step tests
 - ▶ Automated test scripts
 - ▶ Integration testing
 - ▶ Acceptance testing

13

Test oluşturma

Matrix tests

- ▶ Matris testi **yapısal** olarak **aynı** ancak **değerleri farklı olan girişlerin olduğu durumlara uygundur.**
- ▶ **Girişler** ile beklenen **çıkış değerlerini içeren bir liste oluşturulur.**

14

Test oluřturma

Matrix tests

- ▶ Analog digital convertor için ařađıdaki test matrisi oluřturulabilir.

Test Writer: Sue L. Engineer							
Test Case Name:		ADC unit test		Test ID #: ADC-UT-01			
Description:		Verify that each bit of the output can be set independently of the other outputs.		Type: <input type="checkbox"/> white box <input checked="" type="checkbox"/> black box			
Tester Information							
Name of Tester:			Date:				
Hardware Ver: 1.0			Time:				
Setup: Isolate the ADC from the system by removing configuration jumpers. Connect the clk input to a 10-kHz clock source and the Din input to a high-precision voltage course. Connect the output from the ADC to a logic analyzer.							
Test	V _T	Expected output		Pass	Fail	N/A	Comments
		Decimal	Hexadecimal				
1	0.000 V	0	0x000				
2	0.004887 V	1	0x001				
3	0.00977 V	2	0x002				
4	0.01955 V	4	0x004				
...				
10	2.502 V	512	0x200				
Overall test result:							

15

İçerik

- ▶ Test prensipleri
- ▶ Test oluřturma
 - ▶ Debugging
 - ▶ Unit testing
 - ▶ Matrix tests
 - ▶ Step-by-step tests
 - ▶ Automated test scripts
 - ▶ Integration testing
 - ▶ Acceptance testing

16

Test oluřturma

Step-by-step tests

- ▶ Adım adım test yapmak için **önceden test ve sonuçlar tanımlanır.**
- ▶ Matris testindeki tablo oluřturulur, ancak **her adımda test edenin yapacağı işlem belirlenir.**

17

Test oluřturma

Step-by-step tests

- ▶ Örnekte, miktar \$0.25 deęerine ulařınca řeker veren makine test ediliyor.

Test Writer: Sue L. Engineer						
Test Case Name:		Finite State Machine Path Test #1		Test ID #: FSM-Path-01		
Description:		Simulate insertion of money with a mix of nickels and dimes. Verifies FSM outputs candy in response to a total deposit of \$0.30.		Type: <input checked="" type="checkbox"/> white box <input type="checkbox"/> black box		
Tester Information						
Name of Tester:				Date:		
Hardware Ver:		1.0		Time:		
Setup:		Make sure that the system was reset sometime prior and is in state \$0.00.				
Step	Action	Expected Result	Pass	Fail	N/A	Comments
1	Strobe Nickel	State should go to \$0.05				
2	Strobe Dime	State should go to \$0.15				
3	Wait	State should remain \$0.15				
4	Strobe Nickel	State should go to \$0.20				
5	Strobe Dime	State should go to \$0.25				
6	Nothing	State should go to \$0.00				
Overall test result:						

18

İçerik

- ▶ Test prensipleri
- ▶ Test oluşturma
 - ▶ Debugging
 - ▶ Unit testing
 - ▶ Matrix tests
 - ▶ Step-by-step tests
 - ▶ **Automated test scripts**
 - ▶ Integration testing
 - ▶ Acceptance testing

19

Test oluşturma

Automated test scripts

- ▶ Otomatik test betiği, sistemi test etmek için **kullanıcı olmaksızın sağlanan bir dizi komuttur.**
- ▶ **Elde edilen çıkış ile beklenen çıkış otomatik** olarak **kontrol edilir.**
- ▶ Regresyon testi ile, sistemin bir parçasının değiştirilerek hata verip vermediği test edilir.

20

İçerik

- ▶ Test prensipleri
- ▶ Test oluşturma
 - ▶ Debugging
 - ▶ Unit testing
 - ▶ Matrix tests
 - ▶ Step-by-step tests
 - ▶ Automated test scripts
 - ▶ **Integration testing**
 - ▶ Acceptance testing

21

Test oluşturma

Integration testing

- ▶ **Alt sistemler (modüller)** birim testinden sonra daha **büyük modülleri oluşturmak için birleştirilir.**
- ▶ En sonunda tüm sistemi oluşturmak için **tüm modüller entegre edilir.**
- ▶ **Entegrasyon testinde** büyük modüllerin ve **tüm sistemin birlikte doğru çalışıp çalışmadığı kontrol edilir.**
- ▶ Tüm modüller, modüller arasındaki arayüzler ve zamana bağlı işlevler test edilmelidir.

22

İçerik

- ▶ Test prensipleri
- ▶ Test oluşturma
 - ▶ Debugging
 - ▶ Unit testing
 - ▶ Matrix tests
 - ▶ Step-by-step tests
 - ▶ Automated test scripts
 - ▶ Integration testing
 - ▶ **Acceptance testing**

23

Test oluşturma

Acceptance testing

- ▶ **Kabul testleri**, müşterinin sistemi hangi şartlar altında kabul edeceğini belirleyen **formal dokümandır**.
- ▶ **Test senaryoları** gereksinimleri karşılayacak şekilde oluşturulur.
- ▶ Kabul testleri **herkesin anlaştığı sonuçları içermelidir**.
- ▶ Kabul testlerinde **objektif geçti/kaldı kriterleri olmalıdır**.
- ▶ Her test için **başlama ve bitiş tarihleri olmalıdır**.
- ▶ **Her test sonucu kayıt altına alınmalıdır**.
- ▶ **Her test sıralı olmalıdır ve unique ID ye sahip olmalıdır**.
- ▶ **Hata olması durumunda**, sistemin kabul edilmesi için yapılması **gereken düzeltmeler belirtilmelidir**.

24

Ödev

- ▶ Bir otonom araç için kabul testlerini, entegrasyon testlerini ve birim testlerini oluřturunuz.