

# Software Quality and Testing

TURKCELL TECHNOLOGY  
15.12.2011



TURKCELL  
AKADEMİ

## TEST DESIGN TECHNIQUES

- IDENTIFYING TEST CONDITIONS & DESIGNING TEST CASES
- CATEGORIES OF TEST DESIGN TECHNIQUES
- SPECIFICATION-BASED / BLACK-BOX TECHNIQUES
- STRUCTURE-BASED / WHITE-BOX TECHNIQUES
- EXPERIENCE-BASED TECHNIQUES



TURKCELL  
AKADEMİ

## IDENTIFYING TEST CONDITIONS & DESIGNING TEST CASES

### Formality of Test Documentation

- Very formal testing would have extensive documentation which is well controlled, and would expect the documented detail of the tests to include the exact and specific input and expected outcome of the test.
- Very informal testing may have no documentation at all, or only notes kept by individual testers, but we'd still expect the testers to have in their minds and notes some idea of what they intended to test and what they expected the outcome to be.



TURKCELL  
AKADEMİ

## IDENTIFYING TEST CONDITIONS & DESIGNING TEST CASES

### Formality of Test Documentation

The level of formality

- Context
- Organization
  - ❖ Culture
  - ❖ The people working there
  - ❖ How mature the development process is
  - ❖ How mature the testing process is
- Time constraints



TURKCELL  
AKADEMİ

## IDENTIFYING TEST CONDITIONS & DESIGNING TEST CASES

### Test Analysis:Identifying Test Conditions

- Test analysis is the process of looking at something that can be used to derive test information – system requirement, a technical specification, the code itself (for structural testing), or a business process.
- A test condition is simply something that we could test.
- A testing technique helps us select a good set of tests from the total number of all possible tests for a given system.
- Test conditions should be able to be linked back to their sources in the test basis - this is called traceability.



TURKCELL  
AKADEMİ

## IDENTIFYING TEST CONDITIONS & DESIGNING TEST CASES

### Test Design:Specifying Test Cases

- A test case needs to have input values
- The tester needs to determine what the expected result of entering that input would be and document it as part of the test case.
- In order to know what the system should do, we need to have a source of information about the correct behavior of the system - this is called an 'oracle' or a test oracle.
- The test case should also say why it exists - the objective of the test it is part of or the test conditions that it is exercising (traceability)



TURKCELL  
AKADEMİ

## TEST CASES



### Test Case Components

- Test case title
- Test case summary
- Test case priority
- Test module
- Test case steps
- Expected results
- Actual results



TURKCELL  
AKADEMİ

## IDENTIFYING TEST CONDITIONS & DESIGNING TEST CASES

### Test Implementation:Specifying Test Procedures or Scripts

- Group the test cases in a sensible way for executing them and to specify the sequential steps that need to be done to run the test.
- The document that describes the steps to be taken in running a set of tests is called a test procedure, and is often also referred to as a test script.
- Writing the test procedure is opportunity to prioritize the tests, to ensure that the best testing is done in the time available.

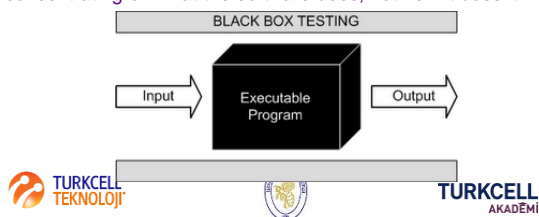


TURKCELL  
AKADEMİ

## BLACK BOX TESTING

### Specification-based (Black Box) Testing

View the software as a black-box with inputs and outputs, but have no knowledge of how the system or component is structured inside the box. In essence, the tester is concentrating on what the software does, not how it does it.



## BLACK BOX TESTING

### Black Box Testing

- Both functional and non-functional testing.
  - Functional testing is concerned with what the system does, its features or functions.
  - Non-functional testing is concerned with examining how well the system does something, rather than what it does.
- Particularly focuses on the functionality of the system
- Test scenarios generated by the expected behavior of the system (expected result)



## BLACK BOX TESTING

### Black Box Test Techniques

#### 1. Equivalence Partitioning

- Good all-round specification-based blackbox technique. It can be applied at any level of testing and is often a good technique to use first.
- Divide (i.e. to partition) a set of test conditions into groups or sets that can be considered the same. By this test coverage percentage will be increased.
- The software should correctly handle values from the invalid partition, by replying with an error message.



## BLACK BOX TESTING

### Black Box Test Techniques

#### 1. Equivalence Partitioning

##### Example :

Some of equivalence classes for a field that must be entered an integer value from -10 to 10

##### Positive Tests

Negative integers -10,-1  
0  
Positive integers 1,10

##### Negative Tests

Negative integers -100,-11  
Floating-point numbers -1/2,1/2,-1/5,1/5  
Positive integers 11,100



## BLACK BOX TESTING

### Black Box Test Techniques

#### 2. Boundary Value Analysis

- Based on testing at the boundaries between partitions.
- Both valid boundaries (in the valid partitions) and invalid boundaries (in the invalid partitions).
- In tests, the end points are areas where errors most frequently encountered.
- Min. and max. values are endpoints



TURKCELL  
AKADEMİ

## BLACK BOX TESTING

### Black Box Test Techniques

#### 3. Decision Table

- Decision tables aid the systematic selection of effective test cases and can have the beneficial side-effect of finding problems and ambiguities in the specification.
- Complex scenarios and complex requirements that need to be tested.
- Decision tables are prepared by taking into consideration the cause and effect relationships.



TURKCELL  
AKADEMİ

## BLACK BOX TESTING

### Decision Table

Conditions	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8
New customer (15%)	T	T	T	T	F	F	F	F
Loyalty card (10%)	T	T	F	F	T	T	F	F
Coupon (20%)	T	F	T	F	T	F	T	F
<b>Actions</b>								
Discount (%)	X	X	20	15	30	10	20	0



TURKCELL  
AKADEMİ

## BLACK BOX TESTING

### Black Box Test Techniques

#### 4. State Transition Testing

- The system can be in a (finite) number of different states, and the transitions from one state to another are determined by the rules of the 'machine'.
- Any system where you get a different output for the same input, depending on what has happened before, is a finite state system.
- In order to see the total number of combinations of states and transitions, both valid and invalid, a state table is useful.



TURKCELL  
AKADEMİ

## BLACK BOX TESTING

### Black Box Test Techniques

#### 4. State Transition Testing

A state transition model has four basic parts:

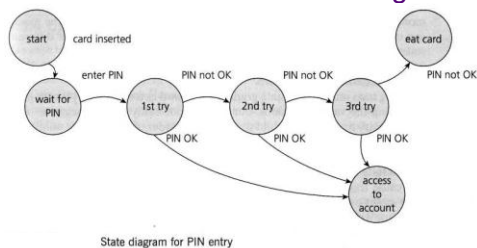
- the states that the software may occupy (open/closed or funded/insufficient funds);
- the transitions from one state to another (not all transitions are allowed);
- the events that cause a transition (closing a file or withdrawing money);
- the actions that result from a transition (an error message or being given your cash).



TURKCELL  
AKADEMİ

## BLACK BOX TESTING

### State Transition Testing



TURKCELL  
AKADEMİ

## BLACK BOX TESTING

### State Transition Testing

State table for the PIN example

	Insert card	Valid PIN	Invalid PIN
S1) Start state	S2	—	—
S2) Wait for PIN	—	S6	S3
S3) 1st try invalid	—	S6	S4
S4) 2nd try invalid	—	S6	S5
S5) 3rd try invalid	—	—	S7
S6) Access account	—	?	?
S7) Eat card	S1 (for new card)	—	—



TURKCELL  
AKADEMİ

## BLACK BOX TESTING

### Black Box Test Techniques

#### 5. Use Case Testing

- Identify test cases that exercise the whole system on a transaction by transaction basis from start to finish.
- Defined in terms of the business user, not the system, describing what the actor does and what the actor sees rather than what inputs the system expects and what the system outputs.
- Serve as the foundation for developing test cases mostly at the system and acceptance testing levels.



TURKCELL  
AKADEMİ

## BLACK BOX TESTING

### Use Case Testing

	Step	Description
<b>Main Success Scenario</b> A: Actor S: System	1	A: Inserts card
	2	S: Validates card and asks for PIN
	3	A: Enters PIN
	4	S: Validates PIN
	5	S: Allows access to account
<b>Extensions</b>	2a	Card not valid S: Display message and reject card
	4a	PIN not valid S: Display message and ask for re-try (twice)
	4b	PIN invalid 3 times S: Eat card and exit

Partial use case for PIN entry



**TURKCELL**  
AKADĒMİ

## WHITE BOX TESTING

### Structure-based (White Box) Testing

- Use the internal structure of the software to derive test cases.
- You can see into the system.
- Require knowledge of how the software is implemented, that
- is, how it works. For example, a structural technique may be concerned with exercising loops in the software. This may be done regardless of the functionality of the software.



**TURKCELL**  
AKADĒMİ

## EXPERIENCE BASED TECHNIQUES

### Experience Based Test Techniques

- People's knowledge, skills and background are a prime contributor to the test conditions and test cases.
- The experience of both technical and business people is important, as they bring different perspectives to the test analysis and design process.
- Due to previous experience with similar systems, they may have insights into what could go wrong, which is very useful for testing.



**TURKCELL**  
AKADĒMİ

## EXPERIENCE BASED TECHNIQUES

### Error Guessing

- Always be used as a complement to other more formal techniques.
- Dependent on the skill of the tester, as good testers know where the defects are most likely to lurk.
- There are no rules for error guessing. The tester is encouraged to think of situations in which the software may not be able to cope.
- List possible defects or failures and to design tests that attempt to produce them.



**TURKCELL**  
AKADĒMİ

## EXPERIENCE BASED TECHNIQUES

### Exploratory Testing

- Hands-on approach in which testers are involved in minimum planning and maximum test execution.
- The planning involves the creation of a test charter, a short declaration of the scope of a short time-boxed test effort, the objectives and possible approaches to be used.
- The test design and test execution activities are performed in parallel typically without formally documenting the test conditions, test cases or test scripts.
- Most useful when there are no or poor specifications and when time is severely limited



TURKCELL  
AKADEMİ

## TEST DESIGN TECHNIQUES

### Apply Different Categories of Techniques

Selection of test techniques based on various factors :

- Type of system
- Regulatory standards & contract terms
- Risks
- Customer & contractual requirements
- The level of documentation
- The test team's knowledge & experience
- Test budget and time constraints
- Software development model



TURKCELL  
AKADEMİ

## TEST DESIGN TECHNIQUES

### Apply Different Categories of Techniques

**Specification-based techniques** are appropriate at all levels of testing (component testing through to acceptance testing) where a specification exists.

**Structure-based techniques** can also be used at all levels of testing. Developers use structure-based techniques in component testing and component integration testing, especially where there is good tool support for code coverage.

**Experience-based techniques** are used to complement specification-based and structure-based techniques, and are also used when there is no specification, or if the specification is inadequate or out of date.



TURKCELL  
AKADEMİ