



TOPMOTIVE

SOFTWARE TESTING

// Internship 2023 DVSE RO

Targu Mures, 26.06.2023

AGENDA

THE IMPORTANCE OF SOFTWARE TESTING

WHAT IS SOFTWARE TESTING

TEST CASES AND BUG REPORTS

QUALITY CHARACTERISTICS

TESTING TYPES AND TECHNIQUES

UI AUTOMATED TESTING WITH SELENIUM WEBDRIVER

EFFECTS OF SOFTWARE DEFECTS

- Major financial losses
- Loss of customers
- Loss of reputation
- Danger to people health and security
- Danger to people freedom
- Danger to people lives
- Risk of war

WHAT IS SOFTWARE TESTING?

WHAT'S ITS PURPOSE?

SOFTWARE TESTING OBJECTIVES

SOFTWARE TESTING OBJECTIVES

**DETERMINE IF
REQUIREMENTS ARE
SATISFIED**

FIND DEFECTS

**PROVIDE CONFIDENCE
IN THE PRODUCT'S
QUALITY**

**MEASURE
QUALITY**

TEST CASE TEMPLATE EXAMPLE

ID: _____ Title: _____ Tested by: _____ Date: _____ System: _____ Objective: _____ Environment: _____			Precondition: _____		
Step	Step Description	Test Data	Expected Results	Actual Results	Comments
1					
2					
3					
4					
5					

**ARE THERE ANY BUG FREE
SOFTWARE PRODUCTS?**

HOW TO REPORT A DEFECT

A software bug constitutes a coding error or fault in the program or system which can produce errors, unintended behavior or application failures.

Title:

Project:

Application Version:

Module:

Defect Severity:

Defect Priority:

Reporter:

Description:

- **Steps to replicate:**
- **Actual result:**
- **Expected result:**

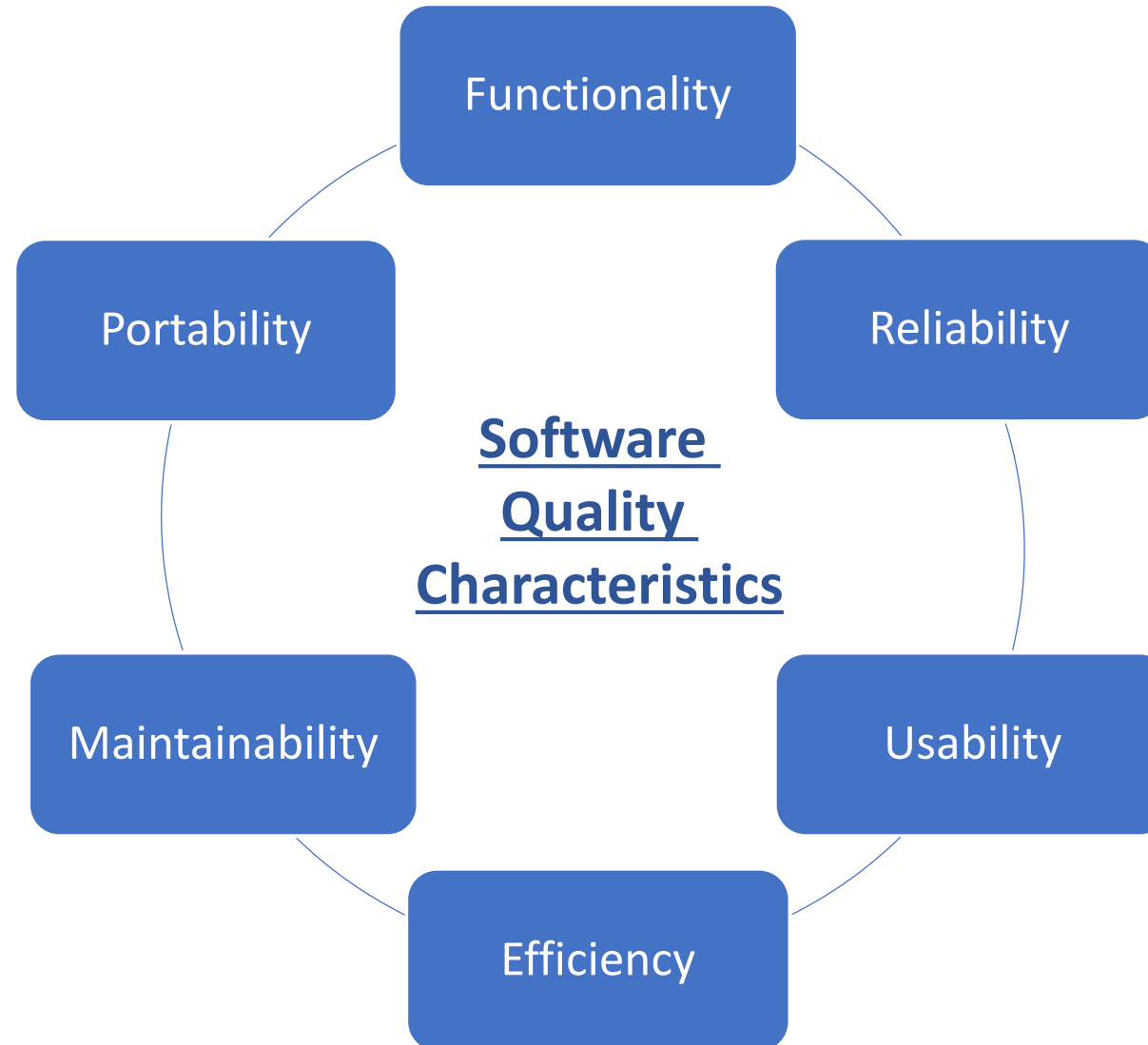
Attachments:

Part II

QUALITY CHARACTERISTICS

SOFTWARE TESTING TYPES AND TECHNIQUES

SOFTWARE QUALITY CHARACTERISTICS



SOFTWARE TESTING TYPES

Functional Testing

Performance Testing

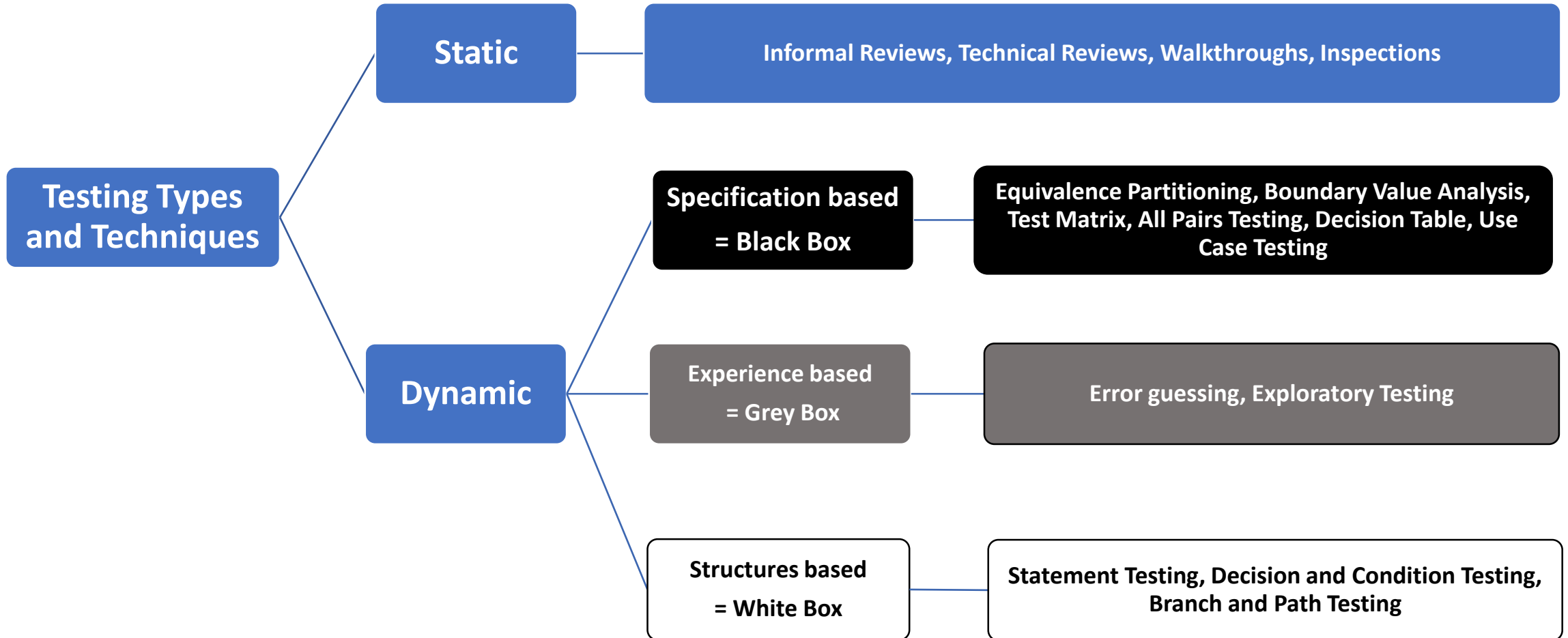
Smoke Test

Exploratory Testing

Security Testing

Regression Testing

SOFTWARE TESTING TYPES AND TECHNIQUES



The main purpose of using testing techniques:

- Reduce the number of test cases and to provide the necessary level of confidence that the software product can go on the market.

BLACK BOX – TEST CASE DESIGN TECHNIQUES

EQUIVALENCE CLASS PARTITIONING - EQP

- It is designed to **minimize the number of test cases** by dividing tests in such a way that the system is expected to act the same way for all sets of inputs or outputs of each equivalence class partition.
- Derive test cases based on equivalence classes:
 - One class that satisfies the condition – the valid class.
 - Second class that doesn't satisfy the condition – the invalid class.

Situation: Bank deposits

A student will get different interests rate depending on the value of their bank deposit. The ranges are:

- 100\$ - 499\$ >> 5%
- 500\$ - 999\$ >> 7%
- 1000\$ - 2000\$ >> 9%

- What are the equivalence classes?
- How many inputs do we test?
- **Do we test almost 2000 inputs?**

BLACK BOX – TEST CASE DESIGN TECHNIQUES

EQUIVALENCE CLASS PARTITIONING - EXAMPLE

100\$ - 499\$ >> 5%

500\$ - 999\$ >> 7%

1000\$ - 2000\$ >> 9%

Test #	Test data	Expected result	Actual result
1.		deposit with 5% interest rate	
2.		deposit with 7% interest rate	
3.		deposit with 9% interest rate	
4.		no deposit	
5.		no deposit	

BLACK BOX – TEST CASE DESIGN TECHNIQUES

EQUIVALENCE CLASS PARTITIONING - EXAMPLE

100\$ - 499\$ >> 5%

500\$ - 999\$ >> 7%

1000\$ - 2000\$ >> 9%

Test #	Test data	Expected result	Actual result
1.	150	deposit with 5% interest rate	
2.	750	deposit with 7% interest rate	
3.	1500	deposit with 9% interest rate	
4.	80	no deposit	
5.	2500	no deposit	

We test only 5 inputs, this is the minimum necessary number of test cases.

BLACK BOX – TEST CASE DESIGN TECHNIQUES

BOUNDARY VALUES ANALYSIS - BVA

- Boundary Values Testing is a method that **tests the boundary whether it's an input, an output or a performance boundary.**
- **If you have a range: (a to b), you will test the following:**



Test Case #	Value	Expected result
1	a-1	Invalid
2	a	Valid
3	a+1	Valid
4	b-1	Valid
5	b	Valid
6	b+1	Invalid

Situation: Bank deposits

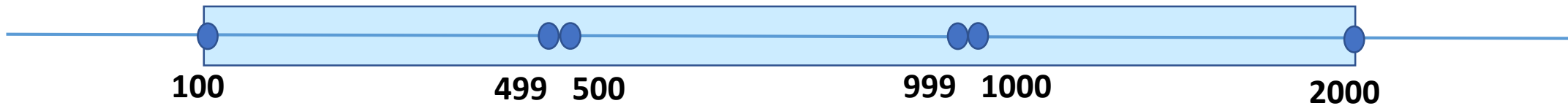
A student will get different interests rate depending on the value of their bank deposit. The ranges are:

- 100\$ - 499\$ >> 5%
 - 500\$ - 999\$ >> 7%
 - 1000\$ - 2000\$ >> 9%
- Which are the boundaries for each class?
 - How many inputs do we test in total?

BLACK BOX – TEST CASE DESIGN TECHNIQUES

BOUNDARY VALUES ANALYSIS - EXAMPLE

A student will get different interests rate depending on the value of their bank deposit.



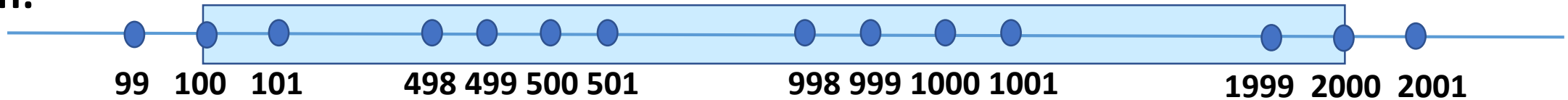
Test #	Test data	Expected result
1.		Invalid = no deposit
2.		Valid = 5% rate
3.		Valid = 5% rate
4.		Valid= 5% rate
5.		Valid = 5% rate
6.		Valid = 7% rate
7.		Valid = 7% rate

Test #	Test data	Expected result
8.		Valid = 7% rate
9.		Valid = 7% rate
10.		Valid = 9% rate
11.		Valid= 9% rate
12.		Valid = 9% rate
13.		Valid = 9% rate
14.		Invalid = no deposit
15.		Invalid = no deposit

BLACK BOX – TEST CASE DESIGN TECHNIQUES

BOUNDARY VALUES ANALYSIS - EXAMPLE

Solution:



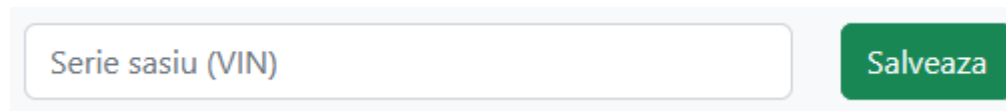
Test #	Test data	Expected result
1.	99	Invalid = no deposit
2.	100	Valid = 5% rate
3.	101	Valid = 5% rate
4.	498	Valid = 5% rate
5.	499	Valid = 5% rate
6.	500	Valid = 7% rate
7.	501	Valid = 7% rate

Test #	Test data	Expected result
8.	998	Valid = 7% rate
9.	999	Valid = 7% rate
10.	1000	Valid = 9% rate
11.	1001	Valid = 9% rate
12.	1999	Valid = 9% rate
13.	2000	Valid = 9% rate
14.	2001	Invalid = no deposit
15.	0	Invalid = no deposit

Important: Always test value 0 (zero)!

BOUNDARY VALUES ANALYSIS

- A garage application can identify vehicles by the VIN number.
- The VIN number consists of 17 alphanumeric characters
- In some cases, vehicles can also be identified by the last 7 characters of the VIN number
- How do we test the functionality?

A screenshot of a web form. It features a light gray rectangular container. Inside, on the left, is a white text input field with a thin gray border and the placeholder text "Serie sasiu (VIN)" in a light gray font. To the right of the input field is a green rectangular button with rounded corners and the text "Salveaza" in white.

BOUNDARY VALUES ANALYSIS

Input	Expected results	Actual results
Valid 17 char VIN number	Valid	
Valid 7 char VIN number	Valid	
16 characters	Invalid	
18 characters	Invalid	
6 characters	Invalid	
8 characters	Invalid	
Special chars: ?`!@#\$/	Invalid	
Nothing	Invalid	
Space	Invalid	
0	Invalid	

BLACK BOX – TEST CASE DESIGN TECHNIQUES

DECISION TABLE

- A decision table is a good way to deal with combinations of inputs/conditions that have different outputs/actions.
- Each rule of a decision table defines an unique combination of conditions which lead to one or more actions.

BLACK BOX – TEST CASE DESIGN TECHNIQUES

DECISION TABLE - EXAMPLE

Situation:

A bank offers services only for residents persons with age between 18 and 55 years old.

The bank offers an interest rate of 10% for students.

Using decision table testing technique create the test cases and then reduce their number.

BLACK BOX – TEST CASE DESIGN TECHNIQUES

DECISION TABLE - EXAMPLE

Solution: $2^3 = 8$ test cases

Conditions	r1	r2	r3	r4	r5	r6	r7	r8
Resident person?	T	T	T	T	F	F	F	F
Age between 18-55?	T	T	F	F	T	T	F	F
Student?	T	F	T	F	T	F	T	F
Actions								
Offer service?	yes	yes	no	no	no	no	no	no
10% interest rate?	no	yes	no	no	no	no	no	no

BLACK BOX – TEST CASE DESIGN TECHNIQUES

DECISION TABLE - EXAMPLE

Reduction of test cases number

What are the mandatory relevant conditions for the insurance to take place? If one of the mandatory conditions is False, the rest of the conditions do not matter.

Conditions	r1	r2	r3	r4	r5	r6	r7	r8
Resident person?	T	T	T	T	F	F	F	F
Age between 18-55?	T	T	F	F	T	T	F	F
Student?	T	F	T	F	T	F	T	F
Actions								
Offer service?	yes	yes	no	no	no	no	no	no
10% interest rate?	no	yes	no	no	no	no	no	no

BLACK BOX – TEST CASE DESIGN TECHNIQUES

DECISION TABLE - EXAMPLE

Conditions	r1	r2	r3	r5
Resident person?	T	T	Don't care	F
Age between 18-55?	T	T	F	Don't care
Student?	T	F	Don't care	Don't care
Actions				
Offer service?	yes	yes	no	no
10% interest rate?	no	yes	no	no

We reduced the number of test cases from 8 to 4 (50% less).
The minimum necessary number of test cases is 4.

BLACK BOX – TEST CASE DESIGN TECHNIQUES

ALL-PAIRS TESTING

- "All Pairs" is a method of testing several variables together.
- The test cases include all the pairs of values for every variable.
- The main role of "All Pairs" is that every value is paired with another value at least once.
- "All Pairs" drastically reduces the number of test cases required to test and it can help you find most errors, with the smallest number of test cases.

Examples of testing several variables:

- A window with several fields.
- A search screen with several search criteria.
- Environment configuration of software (operation system, browsers)

BLACK BOX – TEST CASE DESIGN TECHNIQUES

ALL-PAIRS TESTING - EXAMPLE

Example:

We need to test the application on the following environments and language settings:

- Operating system: Win8 and Win10
- Web Browsers: Firefox and Chrome
- Language: English and German

=> The number of combinations that must be verified are:

OS X Browser X Language = {Win7, Win8} X {FF, Chrome} X {En, De}

BLACK BOX – TEST CASE DESIGN TECHNIQUES

ALL-PAIRS TESTING - EXAMPLE

Reducing the number of test cases: Pick the last row. If all pairs exist in the other rows, then delete the last row and go up, if not keep the row and go up to the next row.

Test Case #	OS	Browser	Language
1	Win 8	FF	English
2	Win 8	FF	German
3	Win 8	Chrome	English
4	Win 8	Chrome	German
5	Win 10	FF	English
6	Win 10	FF	German
7	Win 10	Chrome	English
8	Win 10	Chrome	German

BLACK BOX – TEST CASE DESIGN TECHNIQUES

ALL-PAIRS TESTING - EXAMPLE

Test Case #	OS	Browser	Language
1	Win 7	FF	English
4	Win 7	Chrome	German
6	Win 8	FF	German
7	Win 8	Chrome	English

We have 4 test cases to test.

We reduced the number of test cases from 8 to 4 (50% less).

Q & A

Q & A