

# Test Design Techniques

A thick orange horizontal line that spans the width of the slide, starting from the left edge and ending with a small orange circle on the right side.

# Lesson roadmap

To do



In Progress



Done

# Agenda



Test design techniques



Dynamic Test Design techniques



Boundary value analysis



Equivalence partitioning



Decision table



State transition



Use cases

# Why do we need test design techniques?

Practically, due to time and budget considerations, it is not possible to perform exhausting testing for each set of test data, especially when there is a large pool of input combinations.

- To reduce a number of Test Cases to a necessary minimum
- To select the right Test Cases to cover all possible scenarios

# What is **Dynamic** Testing?

This is a testing method of testing of software program giving input and examining outputs (I/O). This testing requires the code to be run

**Goal: reduce number of defects in working program**

# Dynamic Testing techniques

## Experience Based

Error guessing

Exploratory testing

## Structure Based

Statement

Decision

Condition

Multiple Condition

## Specification Based

Equivalence partitioning

Boundary value analysis

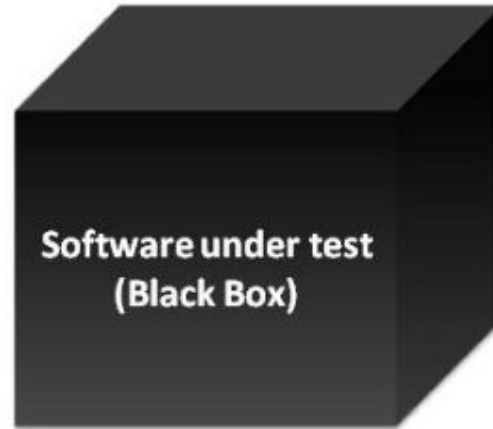
Use case testing

State transition

Decision table

# Specification Based or Black Box

Requirements Document



Validate output



# Boundary Value Analysis

## Equivalence Class Partitioning



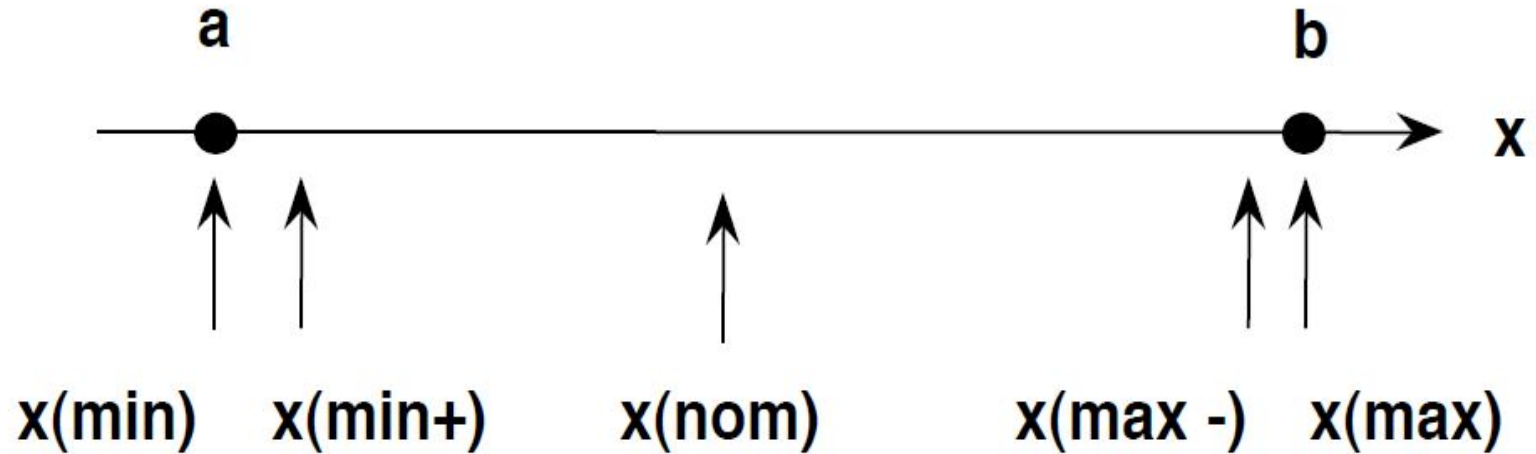
# What Boundary Testing?

Boundary testing is the process of testing boundaries between partitions of the input values

The basic idea in boundary value testing is to select input variable values at their:

- a. Minimum
- b. Just above the minimum
- c. A nominal value
- d. Just below the maximum
- e. Maximum

# What Boundary Testing?







Boundary Testing comes after the Equivalence Class Partitioning

# What is Equivalence Class Partitioning?

In this technique, you divide the set of test condition into a partition that can be considered the same.

- It divides the input data of software into different equivalence data classes
- You can apply this technique, where there is a range in input field

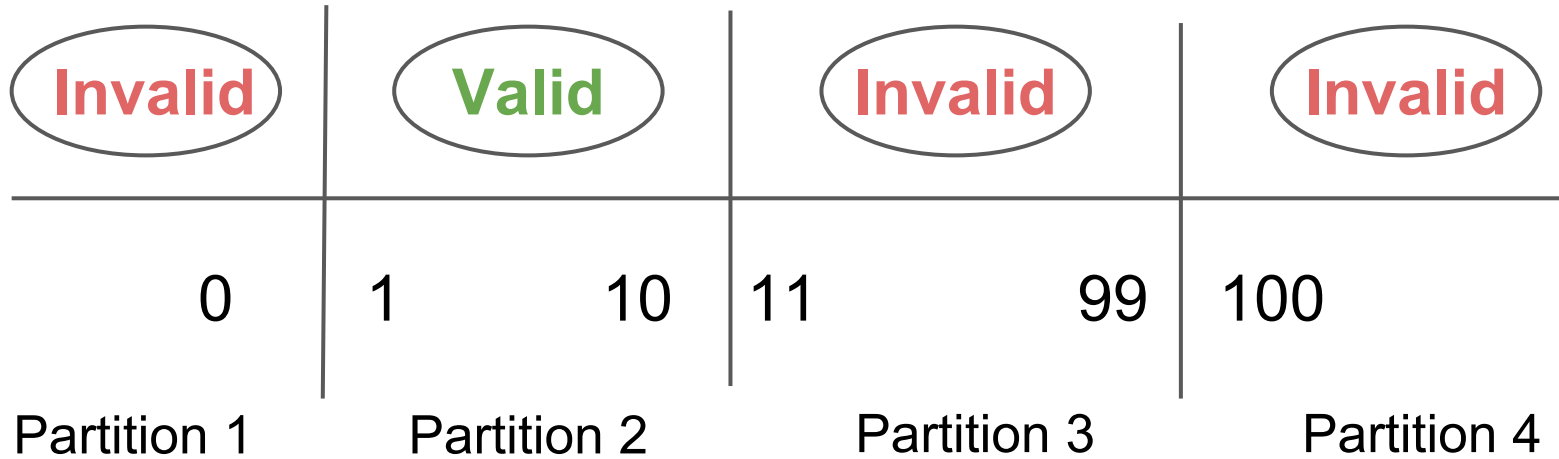
# Equivalence Class Partitioning

Тип номера	Число гостей		Условия	Кол-во
		Только самое главное <b>US\$1 553</b>	• Оплата не возвращается ?	<input type="text"/>
▶ <a href="#">Двухместный номер Делюкс с 1 кроватью или 2 отдельными кроватями</a>		<del>US\$1 832</del>	• Оплата не ?	0 

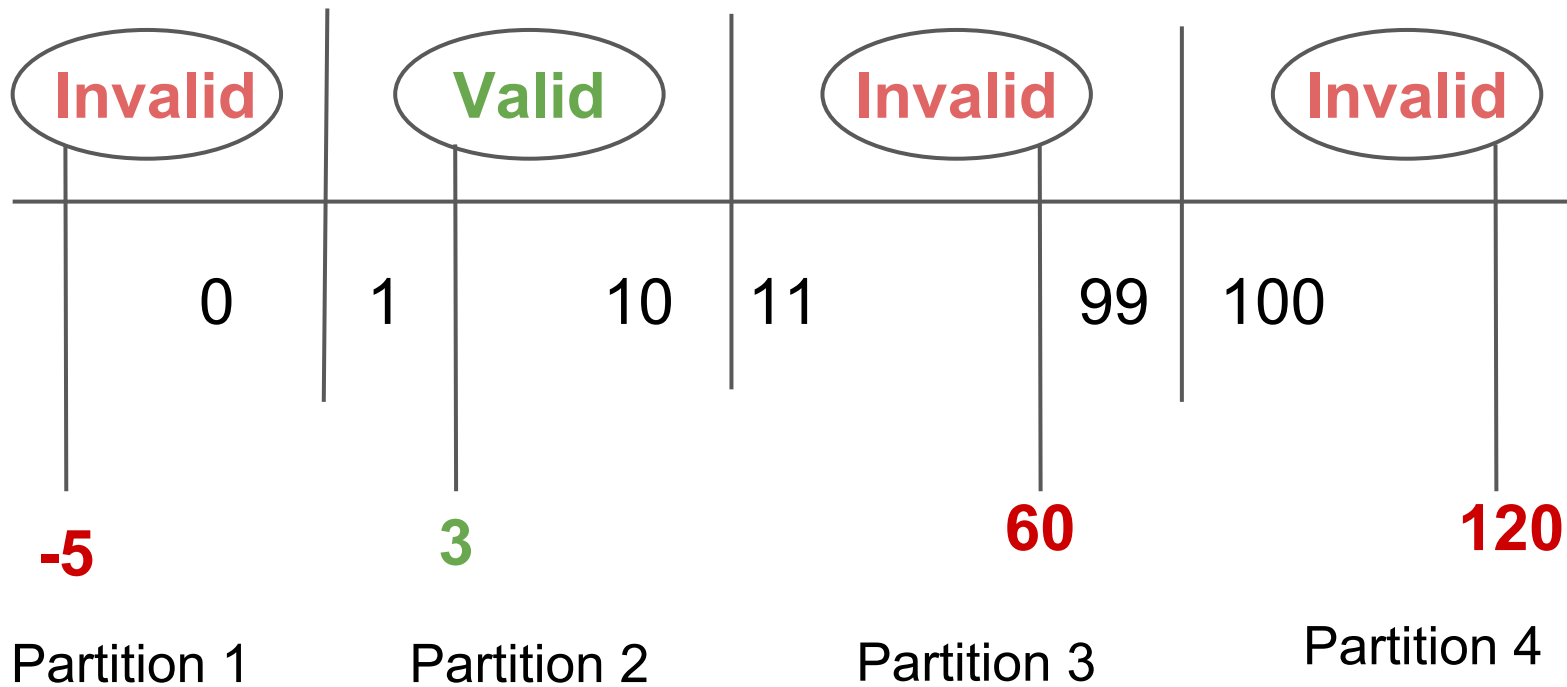
How system decides how many reservations user should be allowed?

Rooms values 1 to 10 are considered valid & room is booked.

While value 11 to 99 are considered invalid for reservation and error message will appear "***Only 10 rooms may be ordered at one time***".

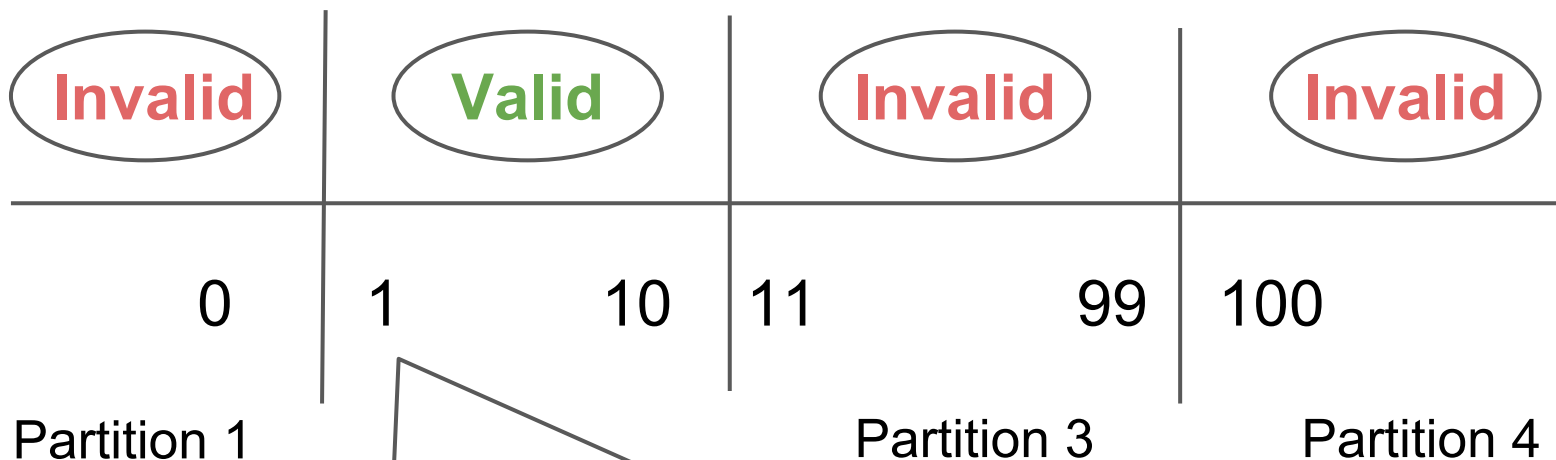


- The divided sets are called Equivalence Partitions or Equivalence Classes.
- We pick only one value from each partition for testing.
- The hypothesis behind this technique is that if one condition/value in a partition passes/fails all others will also pass/fail.



# Boundary Value Analysis

In Boundary Value Analysis, you test boundaries between equivalence partitions



In Boundary Value analysis you will check the boundary values like 0, 1, 10, 11, 99, 100

# EP and BVA Tips

Makes sense for numeric ranges and can be applied to **non-numeric fields as well** (field length for alphabetic fields can come into play as boundaries)

When we test **INVALID** partitions, the safest option is try to cover **only one invalid test condition per test**. This is because programs can stop processing input as soon as they encounter the first problem.

So if we have invalid customer name, address...we can get error “invalid input” but we won’t know whether the test has detected only one invalid input or all of them.



# EP and BVA Examples

Examples of types of partitions:

If you are booking a flight, you have a choice of Economy/Coach, Premium Economy, Business or First Class tickets.

Each of these is an equivalence partitions has its own rights and should be tested.

It doesn't have sense to talk about boundaries for this type.

The invalid partition would be an attempt to type in any other type of flight class (i.e. Staff)

If this is a drop down fields, it should not be possible to add anything, but still good to test and try add.

# Why do both EP and BVA?

1. This testing is used to reduce very large number of test cases to manageable chunks.
2. Very clear guidelines on determining test cases without compromising on the effectiveness of testing.
3. Appropriate for calculation-intensive applications with large number of variables/inputs

**NOTE:** When we use partitions separately from boundaries try to choose partition values that are NOT boundaries, but from the middle of diapason.

# Practical part (1)

---

## Sign up for free now!

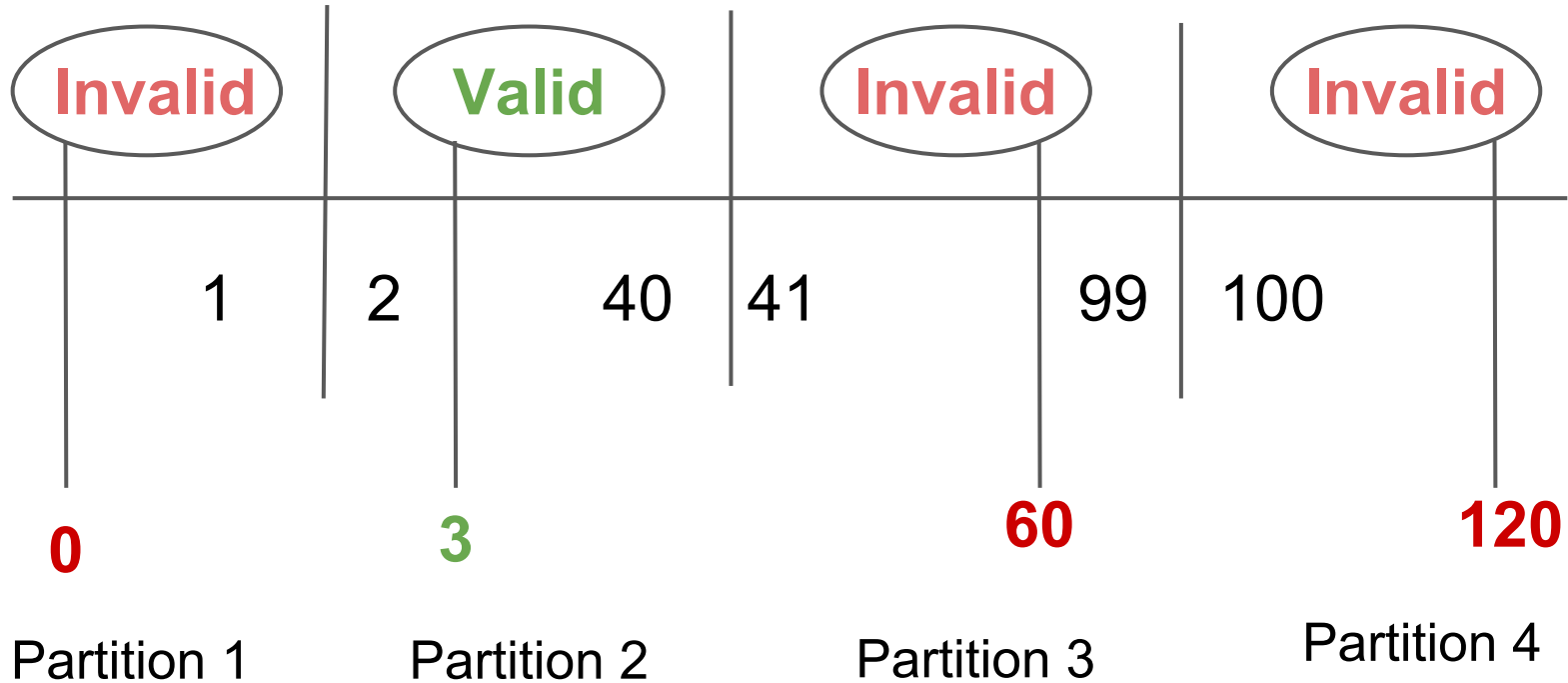
Date of birth

Sign Up Now

Your name should start with a capital letter and should not contain special characters. It should be from 2 to 40 characters in length

1. Define Equivalence partitioning
2. Define Boundary values

Equivalence partitioning and boundary values for field: “ **First Name**” and “**Last Name**”



# Practical part (2)

---

Minimum age should be 18, maximum age should be 120

Your name should start with a capital letter and should not contain special characters. It should be from 2 to 40 characters in length

## Sign up for free now!

Date of birth

Sign Up Now

Your password should contain:

- between 8 and 25 characters
- at least 1 uppercase letter
- at least 1 lowercase letter
- at least 1 number
- at least 1 special character, e.g. %, \$, ?

# Lesson road map

To do

Part #2  
60 mins

Lesson  
Review  
15 mins

In Progress

Coffee  
break  
15 mins

Done

Part #1  
90 mins



User Case

State transitions

Decision table

# What a Use Case?

A use case is a description of a particular use of the system by an actor or user.

It is used widely in developing tests at system or acceptance level.

## **What is Use Case Testing?**

Use Case Testing, is a technique that helps identify test cases that cover the entire system, on a transaction by transaction basis from start to the finishing point.

Use case testing is # of steps that describe the relations between system and user//business user//other system.

# Use Case testing

Main Success Scenario A: Actor S: System	Step	Description
	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
Extensions	2a	Card not valid. S: Display 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

# State transition testing

State Transition is a graphical representation of use cases.

There are 4 basics of state transition:

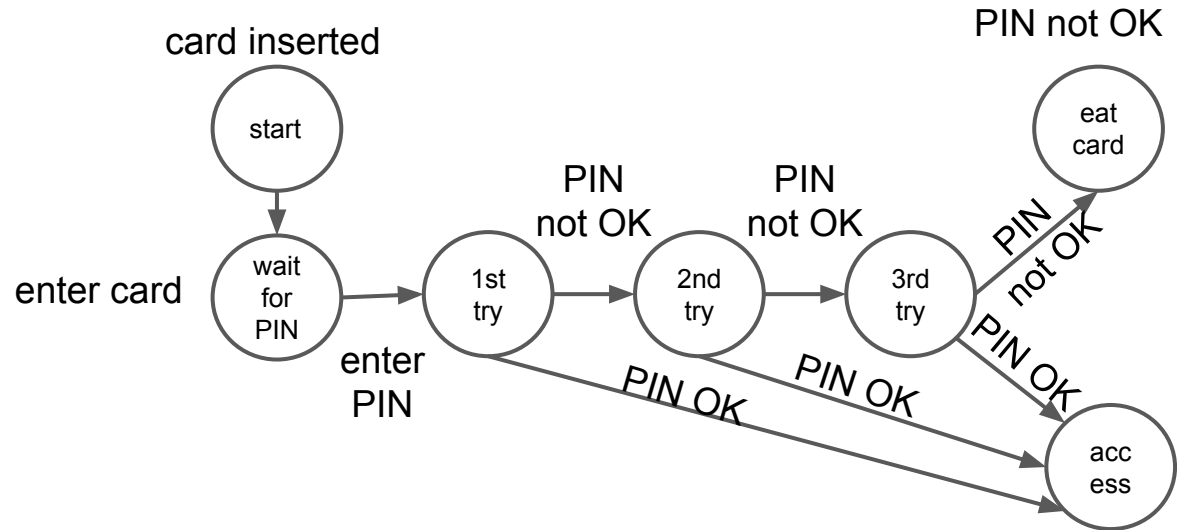
1. The states that the software may occupy
2. The transitions from one state to another (not all transitions are allowed)
3. The events that cause a transition (closing a file, withdrawing money)
4. The actions that result from a transition (an error message)

# Let's see how to use state transitions

System, there use entering a Personal Identity Number (PIN) to access bank account.

There are 4 possible events:

1. Card inserted
2. Enter PIN
3. PIN OK
4. PIN not OK



# Decision table

Decision Table Testing is a good way to deal with a combination of inputs, which produce different results.

It helps reduce test effort in verifying each and every combinations of test data, at the same time ensuring complete coverage.

How to use decision tables for test designing:

- Identify a suitable function or subsystem which reacts according to a combination of inputs or events
- Identify all of the combinations of True and False
- Identify the correct outcome for each combination
- Ensure that the Table is Complete

# Let's see how to use decision table!

You are a new customer opening a credit card account. You'll get 15% discount on all purchases today. If you're an existing customer which hold a loyalty card, you will get 10% discount. If you have a coupon, you can get 20% off today (but it can't be used with the new customer's discount).

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
Actions								
Discount (%)	--	--	20	15	30	10	20	0
Error	Y	Y						

# Practical part (1)

---



# Decision Tables – Task 1/2

**Task:** Using the rules for procurement of a loan below prepare a decision table for the list of people (see the next slide) and make a decision will they get a loan or not

**Work rules:**

- Loan is not granted to unemployed people
- Loan is granted for people with permanent legal work with no restrictions

**Age rules:**

- For people under 18 loan is not granted
- People of 18 – 40 age are the target group for loan granting
- For people over 40 loan is not granted

**Income rules:**

- Loan is not granted for people with income less than 10000 hrivnas/month
- Loan is granted for people with income of 10000-50000 hrivnas/month, loan amount is < 20000 hrivnas
- Loan is granted for people with income over 50000 hrivnas/month, loan amount is > 20000 hrivnas

## Decision Tables – Task 2/2

- User 1: Student, unemployed, 17 years old, income is 800 hrivnas / month (scholarship)
- User 2: Post graduate, unemployed, 24 years old, no income
- User 3: Retired, unemployed, 62 years old, income is 2000 hrivnas / month (pension)
- User 4: Office Manager, legally employed, 50 years old, income is 15000 hrivnas / month
- User 5: Lawyer, legally employed, 45 years old, income is 55000 hrivnas / month
- User 6: Office Worker, legally employed, 34 years old, income is 8500 hrivnas / month
- User 7: Marketing Specialist, legally employed, 29 years old, income is 28000 hrivnas / month
- User 8: Private Entrepreneur, legally employed, 38 years old, income is 120000 hrivnas / month

# Decision Tables – Solution 1/3

[illegible]

## Decision Tables – Solution 2/3

<b>Conditions</b>		<b>User 1</b>	<b>User 2</b>	<b>User 3</b>	<b>User 4</b>	<b>User 5</b>	<b>User 6</b>	<b>User 7</b>	<b>User 8</b>
<b>Work</b>	<b>Unemployed</b>								
	<b>Legal work</b>								
<b>Age</b>	<b>&lt; 18</b>								
	<b>18 - 40</b>								
	<b>&gt; 40</b>								
<b>Income</b>	<b>&lt; 10000</b>								
	<b>10000 - 50000</b>								
	<b>&gt; 50000</b>								
<b>Loan amount</b>	<b>No loan</b>								
	<b>&lt; 20000</b>								
	<b>&gt; 20000</b>								

## Decision Tables – Solution 3/3

Conditions		User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8
Work	Unemployed	YES	YES	YES					
	Legal work				YES	YES	YES	YES	YES
Age	< 18	YES							
	18 - 40		YES				YES	YES	YES
	> 40			YES	YES	YES			
Income	< 10000	YES	YES	YES			YES		
	10000 - 50000				YES			YES	
	> 50000					YES			YES
Loan amount	No loan	YES	YES	YES	YES	YES	YES		
	< 20000							YES	
	> 20000								YES

# Difference in Static & Dynamic testing

---

# Difference between testing



Done without executing	Done by executing the program
Does verification process	Does validation process
Is about prevention of defects	Testing is about finding and fixing defects
Gives assesment of code and docs	Gives bugs/bottlenecks in the software
Involves checklist and process to be followed	Involves Test Cases for execution

Can be performed before compilation	Is performed after compilation
Covers the structural and statement coverage testing	Covers the executable file of the code
Cost of finding defects and fixing is less	Cost of finding and fixing defects is high
Return on investment will be high as this process involved at early stage	Return on investment will be low as this process involves after the dev phase
More reviews comments are highly recommended for good quality	More defects are highly recommended for good quality.
Requires loads of meetings	Comparatively requires lesser meetings



# Lesson road map

To do

In Progress

Done



# Lesson review questions

1. Назовите техники тест дизайна?
2. Чем BVA отличается от EP?
3. На каких уровнях тестирования применяется BVA?
4. Чем статическое тестирование отличается от динамического?

1. Quiz
2. Тест кейсы с учетом техник для Password Forgotten
3. Тест кейсы для функциональности Login