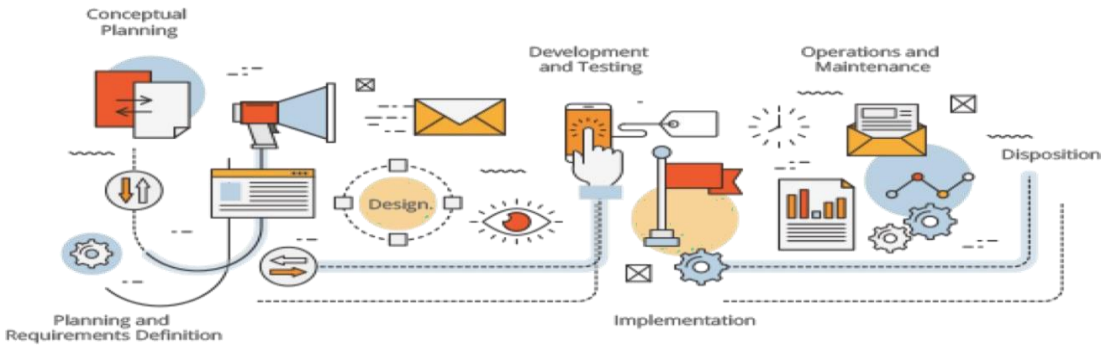
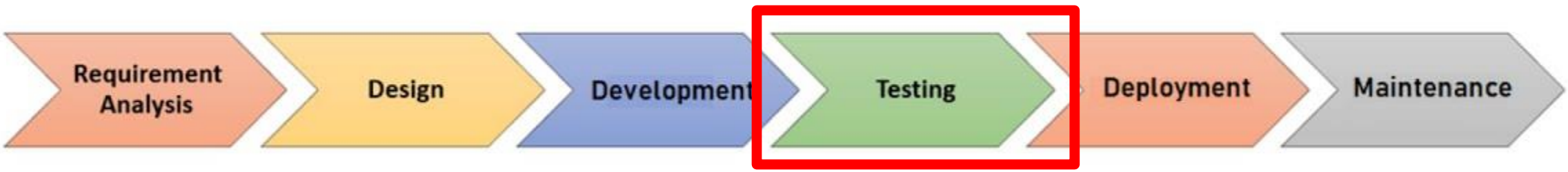
Software Engineering

Requirement Engineering



Software Development Life Cycle (SDLC)



Equivalence Partitions analysis

Coals:

- To reduce the number of test cases to a necessary minimum
- To select the right test cases to cover all possible scenarios (of course not to be absolutely sure);
- Maintaining acceptable test coverage.

Using this technique, the tester should be remembered that:

- Too many equivalence classes increases the likelihood that multiple tests would be superfluous (excessive);
- Too few equivalence classes increases the likelihood that the error will be skipped product.

Equivalence Partitions analysis

Test case design by EP proceeds into 2 steps:

- Identify equivalence classes;
- Choose one representative from each equivalence classes;
- Define the test cases.

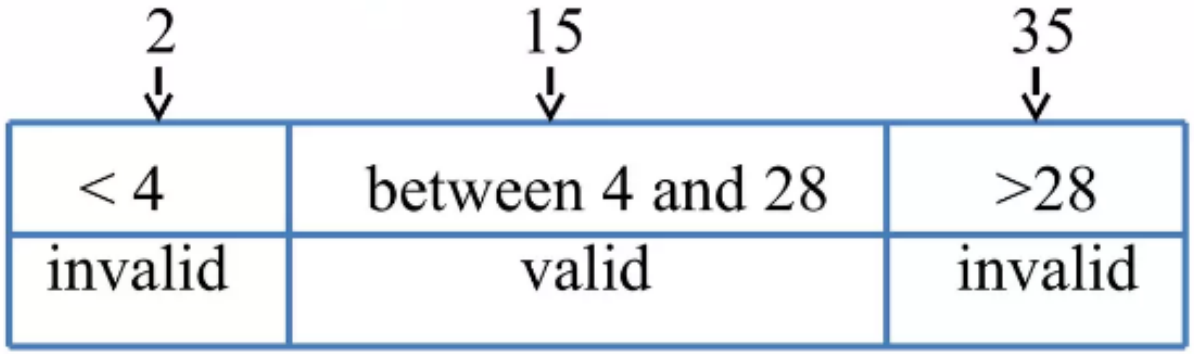
Equivalence Classes – Rules

- Input condition is a range: one valid and two invalid classes are defined
- Input condition requires specific value: one valid and two invalid classes are defined
- Input condition is boolean: one valid and one invalid class are defined

☞ Then define one test case for each equivalence class

Example: Requirement for 'Password' field from "Add users modal window" of Admins actions menu: password field can not be shorter than 4 and longer than 28 (including) characters (numeric and alphabetic)

Equivalance classes



Define and execute the test cases:

1. Password field contain 2 characters – Fail;
2. Password field contain 15 characters – Pass;
3. Password field contain 35 characters – Fail.

Example: Requirement for 'Count' field from "Order page" of Manager: Count field is numeric field, only integer number (0-255), not Alphabetic (A - Z), (a-z) and special characters.

Equivalance classes	Case
less than 0	invalid
beetwen 0 and 255	valid
more than 255	invalid
string	invalid
illegal characters	invalid

Define and execute the test cases:

1. Enter '-5' into 'Count' field – Fail
2. Enter '50' into 'Count' field– Pass
3. Enter '330' into 'Count' field– Fail
4. Enter any word into 'Count' field – Fail
5. Enter '(!@#$%^&' into 'Count' field – Fail

Example: In a system designed for postal services' payment: Letters up to 100g are called as 'light'.Postal rates for sending the light letters up to 10g are $25. The next 40g should be played by $35. Each next 25g up to 100g should be played by an extra $10. Partitions should be designed for "grams".

Equivalance classes:	Price
1) 0 – 10g:	25$
2) 11-50g:	35$
3) 51-75g:	45$
4) 76-100g:	55$
5) >100g	-

Define and execute the test cases:

Letter has '-5' g – Fail
Letter has '6' g – Pass (25$)
Letter has '33' g – Pass (35$)
Letter has '64' g – Pass (+10$)
Letter has '88' g – Pass (+10$)
Letter has '105' g – Fail

Equivalence classes Examples

Input or Output Event	Valid Equivalence Classes	Invalid Equivalence Classes
Enter a non-zero digit	1 – 9	< 1, > 9
Enter the first letter of a name	First character is a capital letter First character is a lower case letter	First character is not a letter
Draw a line **Max. 4 inches**	From 1 to 4 inches long	No line Longer than 4 inches

Example: Customer-acc-number

Input or Output Event	Valid Equivalence Classes	Invalid Equivalence Classes
Customer => single character	Character	Non char
Acc (number) => Three digits	All three are numbers/digits	One char two digits
Number => any number in range 100-500	With in range 100 -500	Above /below range

Consider percentage field that will accept percentage only between 50 to 90 %, more and even less than not be accepted, and application will redirect user to an error page. If percentage entered by user is less than 50 %or more than 90 %, that equivalence partitioning method will show an invalid percentage. If percentage entered is between 50 to 90 %, then equivalence partitioning method will show valid percentage.

Equivalence Classes:

1.Invalid Equivalence Class:

1. Percentage less than 50% (e.g., 49%)
2. Percentage greater than 90% (e.g., 91%)

2.Valid Equivalence Class:

1. Percentage between 50% and 90% (inclusive)

Equivalence Partitioning		
Invalid	Valid	Invalid
<=50	50-90	>=90

Boundary Test Cases:

1.Invalid Equivalence Class:

1. Test with percentage less than 50% (e.g., 49%)
2. Test with percentage greater than 90% (e.g., 91%)
3. **on-numeric values** Inputs that are not numbers should be invalid (if the field accepts only numeric input). [abc", "@$%", "fifty“]

2.Valid Equivalence Class:

1. Lowest valid percentage: 50% (e.g., 50%)
2. Highest valid percentage: 90% (e.g., 90%)

Test Cases Based on ECP

Test Case	Input Value	Expected Output
Valid Percentage	50	Accepted
Valid Percentage	75	Accepted
Valid Percentage	90	Accepted
Invalid Percentage (Low)	49	Rejected
Invalid Percentage (High)	91	Rejected
Invalid Input (Negative)	-10	Rejected
Invalid Input (Non-numeric)	"abc"	Rejected

Let us consider an example of an online shopping site. In this site, each of products has a specific product ID and product name. We can search for product either by using name of product or by product ID. Here, we consider search field that accepts only valid product ID or product name. If the product ID entered by user is invalid then application will redirect customer or user to error page. If product ID entered by user is valid i.e. 45 for mobile, then equivalence partitioning method will show a valid product ID.

Product	Product ID
Mobiles	45
Laptops	54
Pen Drives	67
Keyboard	76
Headphones	34

Equivalence Classes:

1.Invalid Equivalence Class:

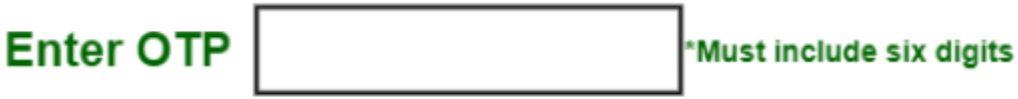
1. Product ID that is not in the list (e.g., 123)
2. Invalid Product ID (Non-existent or Out of Range)
   - Example: 9999, -1, abc
   - Expected Result: Redirect to error page
3. Invalid Product Name (Non-existent Product)
   - Example: "xyz123", "Fake Product“
   - Expected Result: Redirect to error page
4. Empty Input (Blank Search Field)
   - Example: "" (empty string)
   - Expected Result: Error message or prompt to enter valid data
5. Special Characters or Unexpected Input
   - Example: @#$%, null, <>?
   - Expected Result: Error message or input validation failure

2.Valid Equivalence Class:

1. Product ID that is in the list (e.g., 45 for Mobiles)

Equivalence Partioning		
Invalid	Invalid	Valid
77	84	45

Consider an OTP number that contains only 6 digit number, greater and even less than six digits will not be accepted, and the application will redirect customer or user to error page.



Equivalence Classes:

- **Invalid Equivalence Class:** Input with a number of digits less than 6 or greater than 6.
- **Valid Equivalence Class:** Input with exactly 6 digits.

Equivalence Partioning			
Invalid	Invalid	Valid	Valid
Digits>=7	Digits<=5	Digits=6	Digits=6
67545678	9754	654757	213309

According to the specification, a program accepts 4 to 10 inputs, which are five-digit integers greater than 10,000. Identify equivalence partitions and possible test cases.

Test Case Description	Input	Expected Outcome
Invalid: Fewer than 4 integers	3 integers	Redirect to error page
Invalid: More than 10 integers	11 integers	Redirect to error page
Invalid: Integer < 10,000	9999	Redirect to error page
Invalid: Non-integer input	"abc"	Redirect to error page
Valid: 4 integers	10001, 20002, 30003, 40004	Accept input
Valid: 10 integers	10001, 20002, ..., 100010	Accept input
Valid: 5 integers	10001, 20002, ..., 50005	Accept input
Valid: 6 integers	10001, 20002, ..., 60006	Accept input
Valid: 7 integers	10001, 20002, ..., 70007	Accept input
Valid: 8 integers	10001, 20002, ..., 80008	Accept input
Valid: 9 integers	10001, 20002, ..., 90009	Accept input

In an Examination a candidate has to score minimum of 24 marks in order to clear the exam. The maximum that he can score is 40 marks. Identify the Valid Equivalence values if the student clears the exam.

The classes will be as follows:
Class I: values < 24 => invalid class
Class II: 24 to 40 => valid class
Class III: values > 40 => invalid class

Marks Obtained	Outcome	Valid/Invalid
0 - 23	Fail	Invalid
24 - 40	Clear	Valid

Equivalence Partitions analysis

Pluses of EP technique:

- reducing the number of tests;
- reduction in testing time;
- improvement structured testing.

Minuses of EP technique:

- if misused technology we risk losing bugs.

Equivalence partitioning is no standalone method to determine test cases. It has to be supplemented by boundary value analysis.