Function and Domain testing

Andrei Kutsko

AGENDA

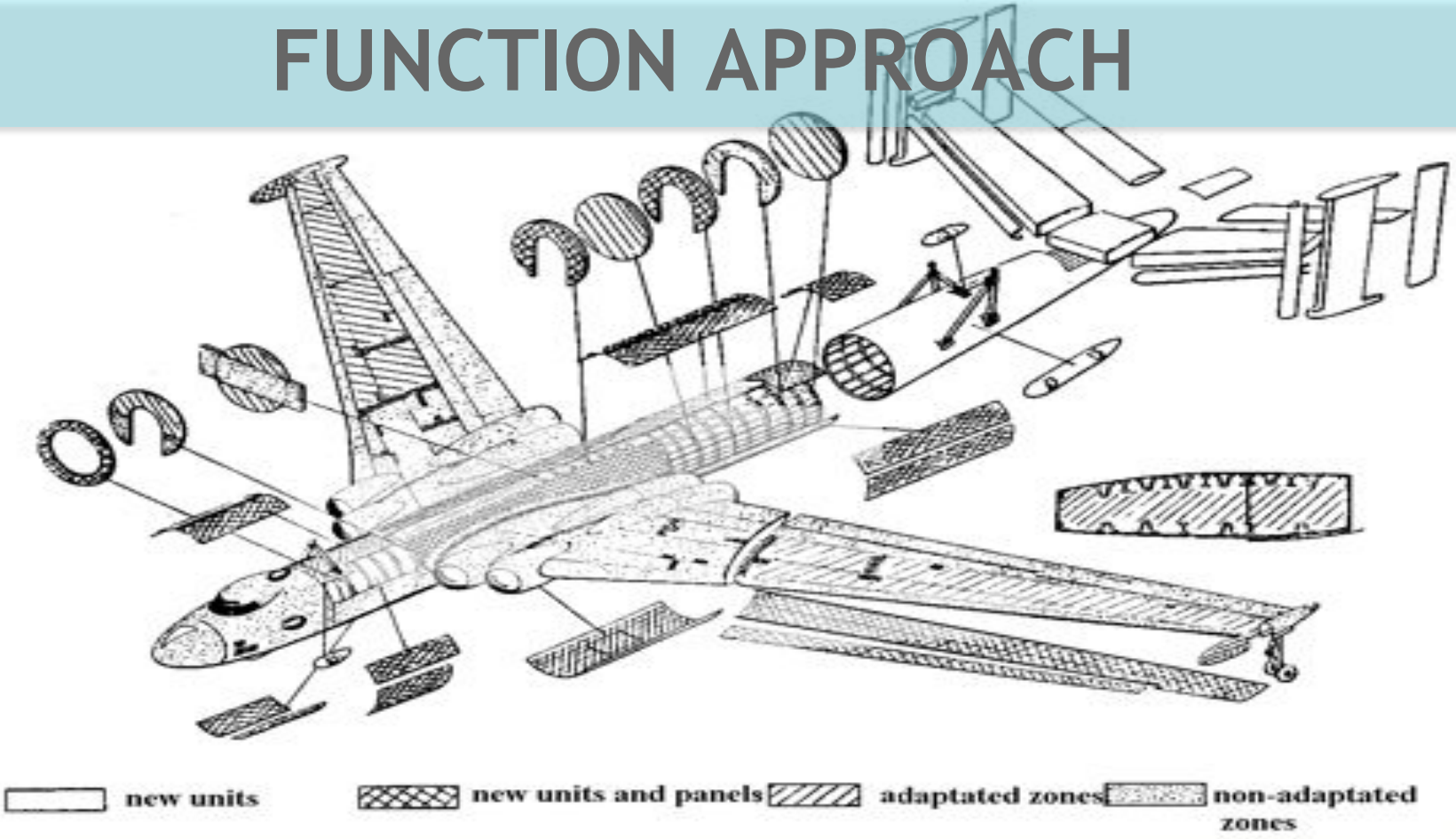
Function Approach

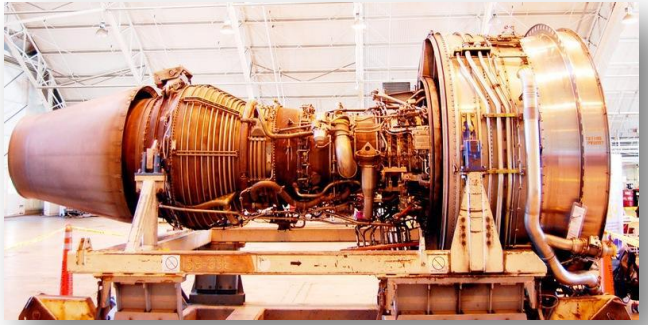
- Idea and Steps
- Example

Domain Approach

- Problem Description
- Equivalence Classes
- Idea and Steps
- Examples

FUNCTION APPROACH

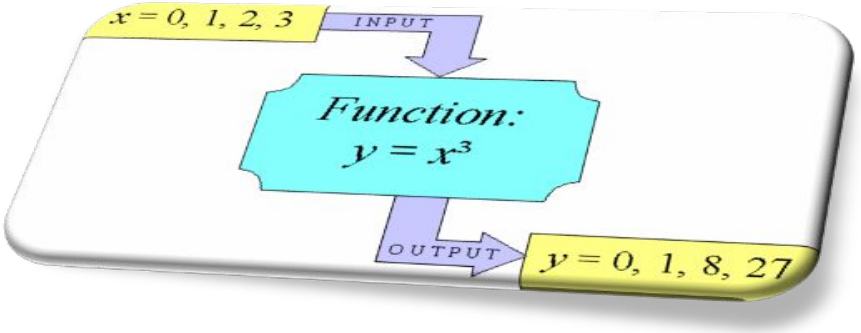




Test each function thoroughly, one at a time

FUNCTIONS

Function is something the product can do



Functions **may be called features / commands or they may be identified only by what they do**

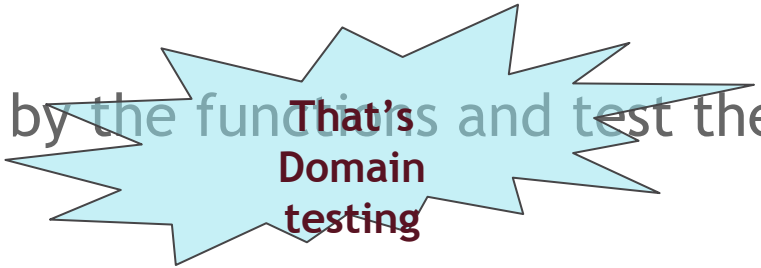
THE FUNCTION LIST

- Category 1
  - Function 1
    - Input of the function
    - Output of the function
    - Possible scope of the function
    - Options / configurations of the function
    - Conditions in which the function behaves differently
  - Function 2
    - Input…

Steps

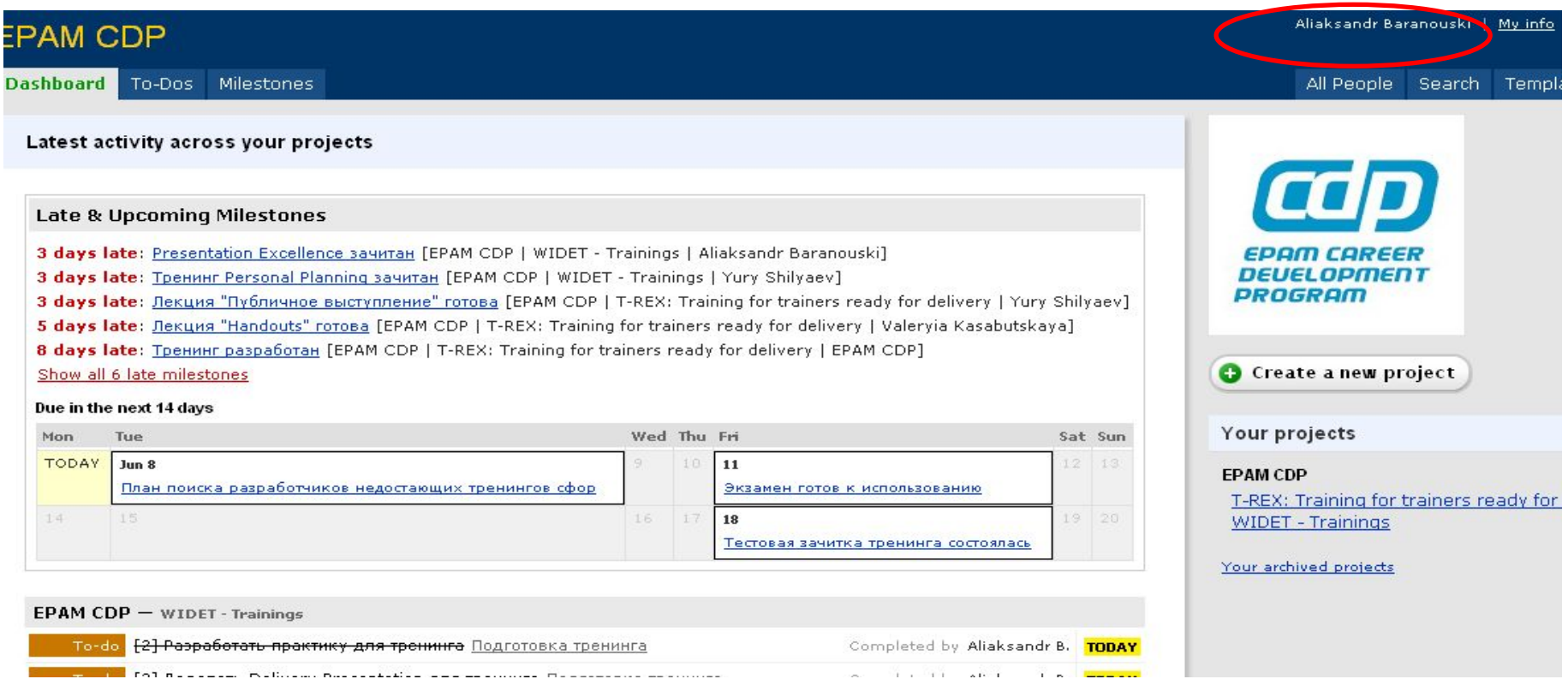
STEPS

1. Identify the program's **features** / commands
2. Identify **variables** used by the functions and test their **boundaries**
3. Identify possible **dependent** variables
4. Use each function in a **mainstream way** and push it in as many ways as possible, as **hard as possible**

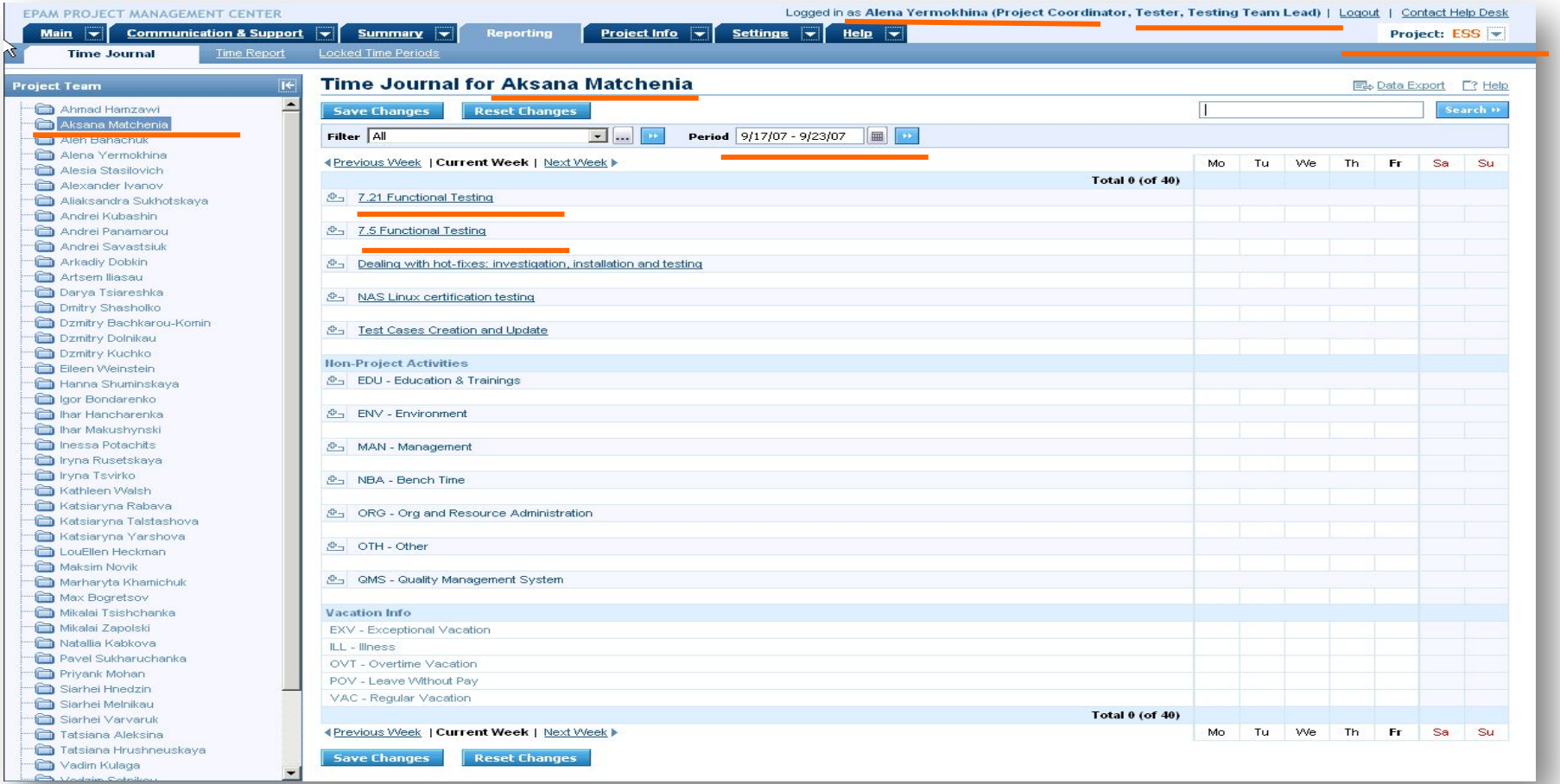


HIDDEN TEXT FIELD

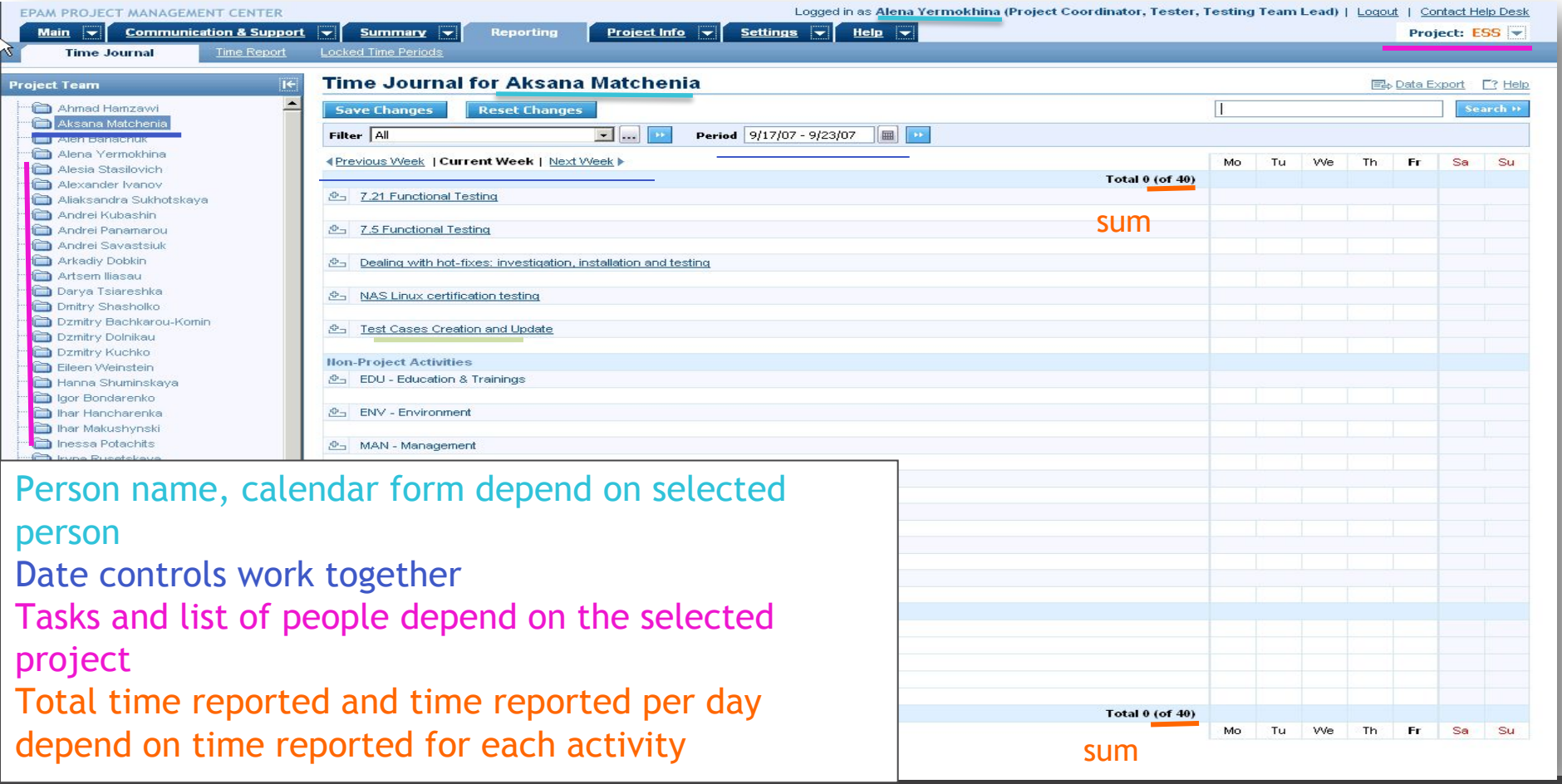
A text field, that takes value from another part of application



IDENTIFY HIDDEN TEXT FIELDS



IDENTIFY DEPENDENT CONTROLS



Person name, calendar form depend on selected person

Date controls work together

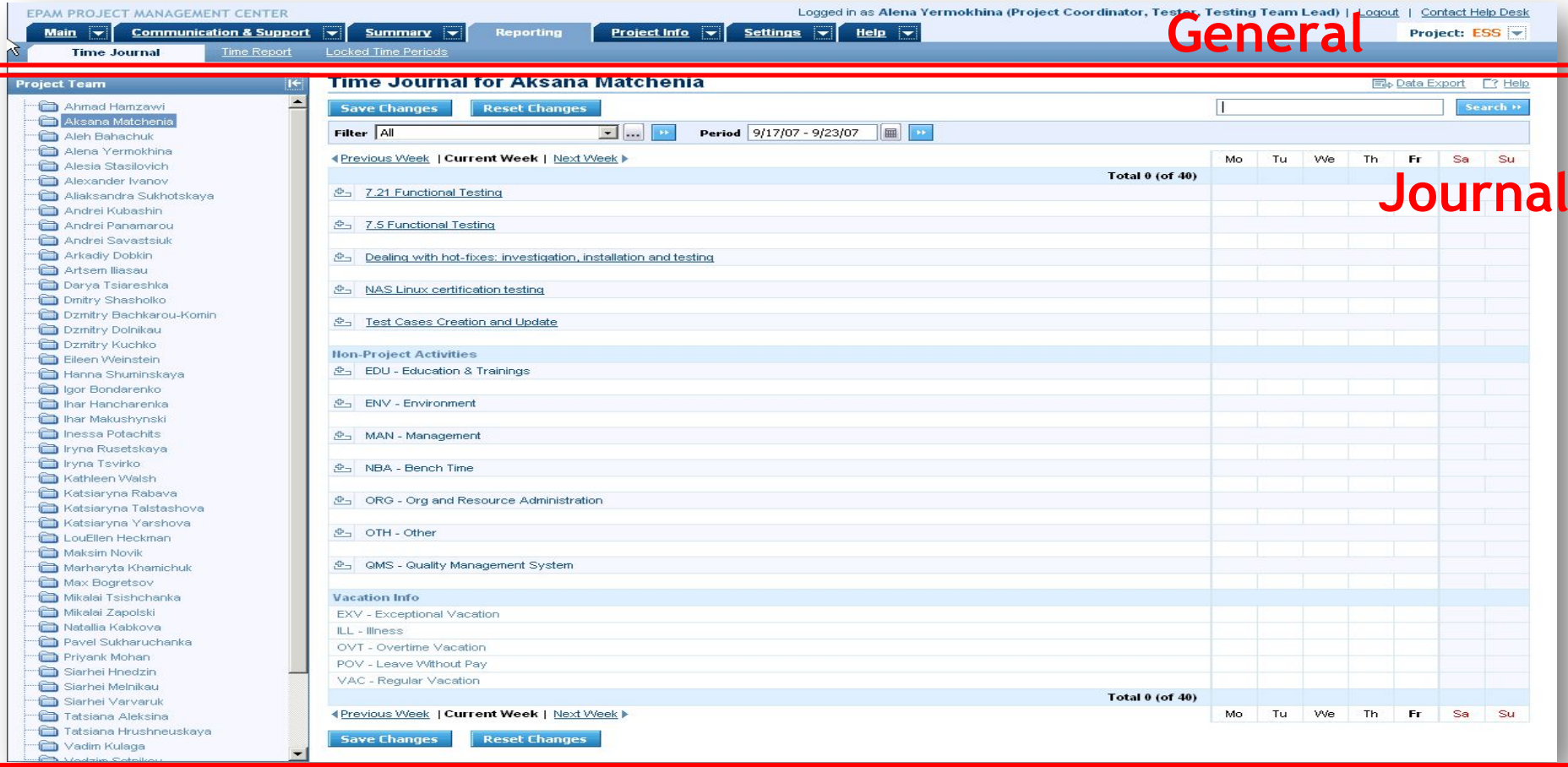
Tasks and list of people depend on the selected project

Total time reported and time reported per day depend on time reported for each activity

STEPS

1. Start from upper level
2. Use classes
3. Add questions where it is necessary
4. Collaborations with other apps?
5. Pairs dependencies

PMC TIME JOURNAL



GENERAL

Area	Function (Class)	Comment
General		
	Logged in	
		Name and Roles as text
		Dependence of functions on the rights
	Logout as link	
	Contact Help Desk as link	Mail to Help Desk
	Navigation tabs	
		Main (as link, as list)
		Communication & Support (as link, as list)
		Summary (as link, as list)
		Reporting (as link, as list)
		Project Info (as link, as list)
		Settings (as link, as list)
		Help (as link, as list)
	Project switcher	Projects as list
		Dependence of Project Team and Activities on the selected project

WHEN?

- ✔ **Initial testing of product**
  - Test new features of product
  - Fast scan for serious problems

- ✔ **Coverage issues**
  - Provide status for the whole app
  - Coverage for particular area

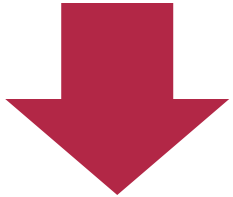
STRENGTH?

- ✔ Thorough analysis of each item tested

Blind spots?

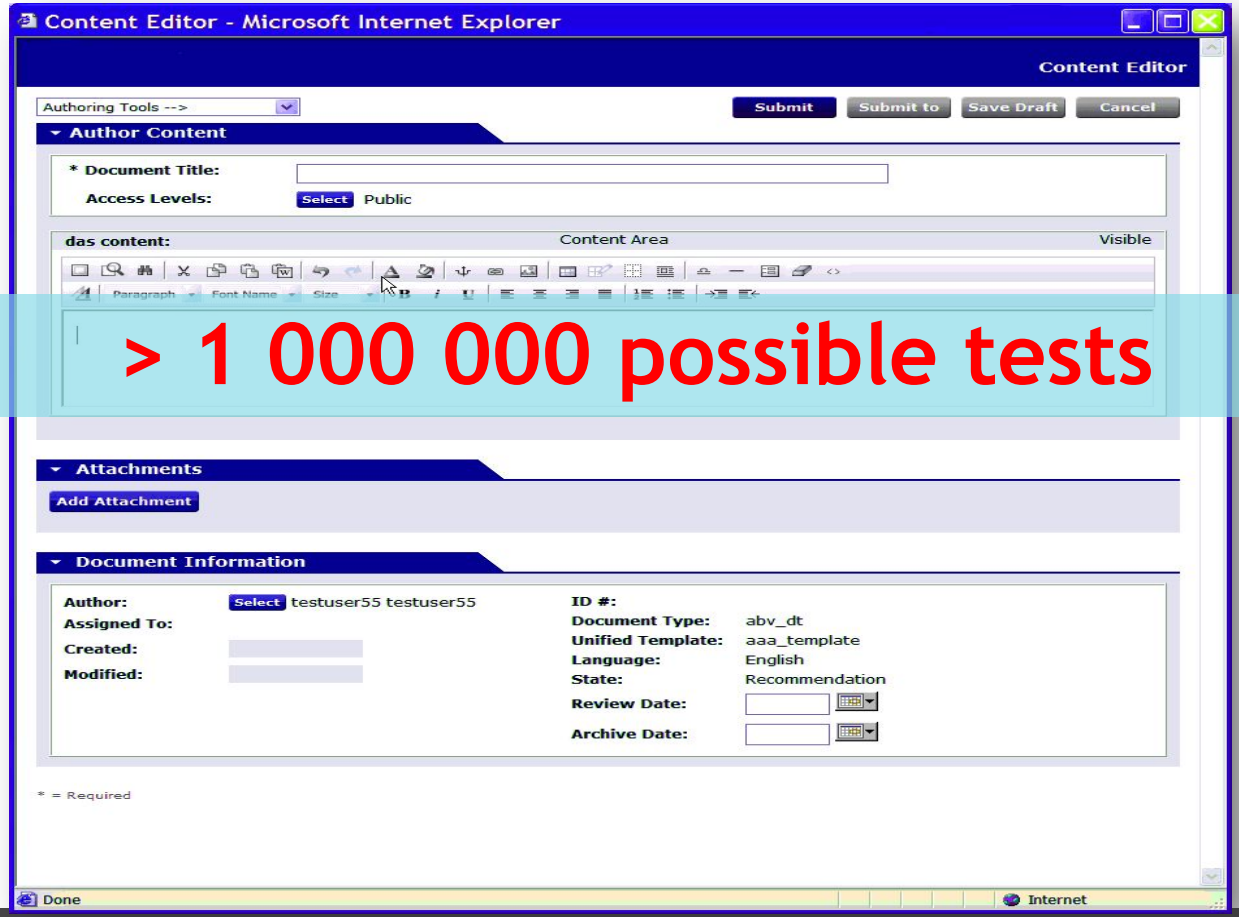
- ✔ Misses feature interactions
- ✔ Misses load-related issues, interaction with background tasks, effects of interrupts
- ✔ Doesn’t address user tasks

2. Identify **variables** used by the functions and test their **boundaries**



Domain Approach

PROBLEM DESCRIPTION



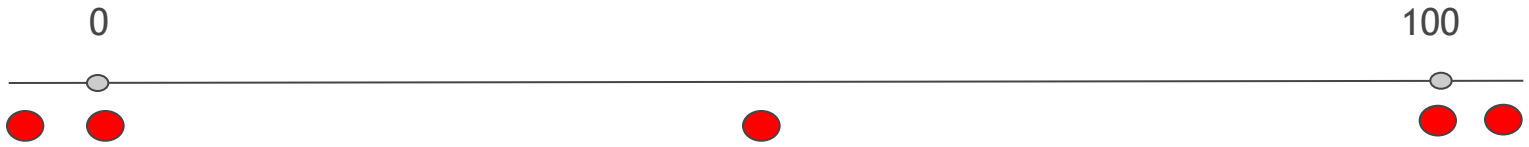
IDEA

We need a **strategy**, that provides **a rationale for selecting** a few test cases from a huge number

Equivalence Classes

EQUIVALENCE CLASSES

Two values are equivalent if the program would take the **same path in response** to each.



DIMENSIONS

Producer

Size

Chocolate

Shape

Cover

Filling

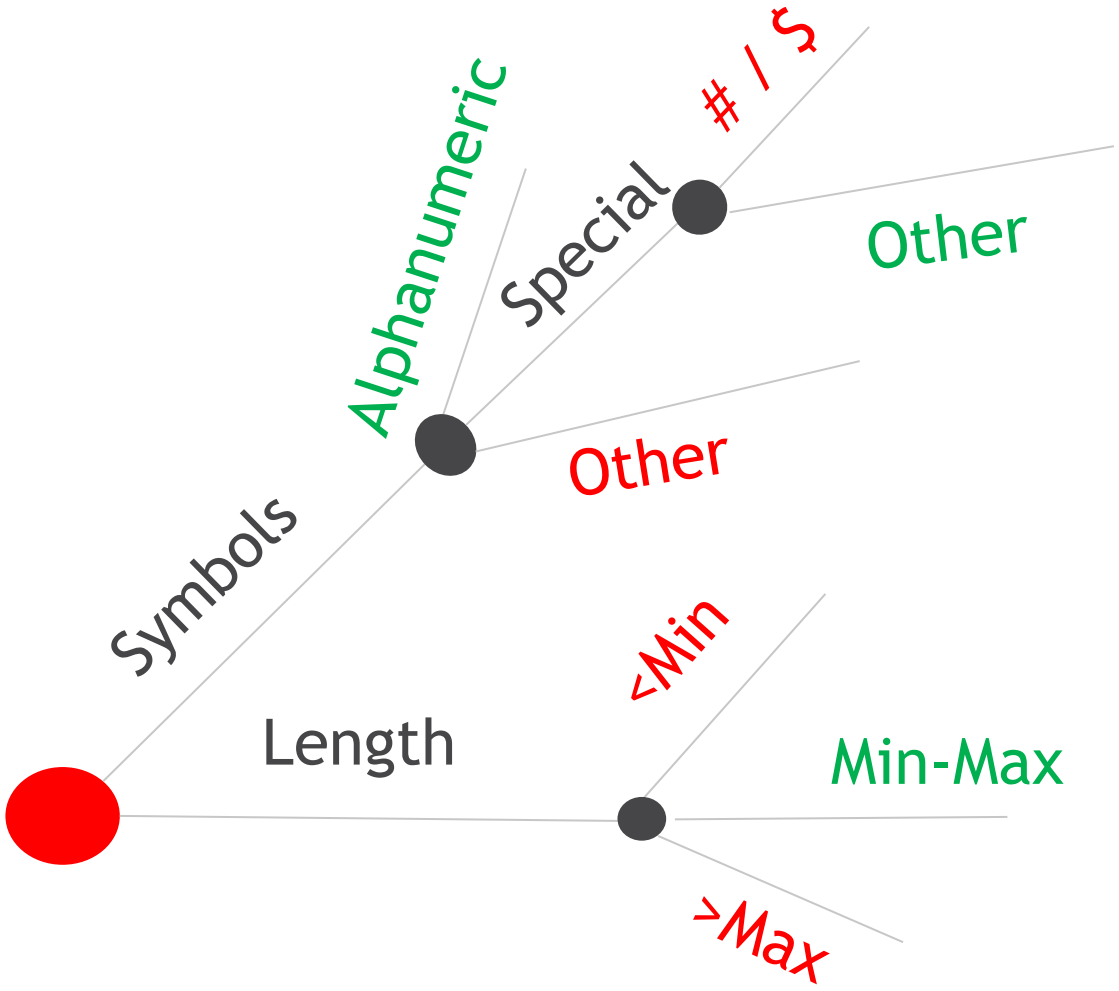
STEPS

- Identify variables
- Identify dimensions
- Define Borders (E.C.)
- Select values

EXAMPLE

В поле вводится имя пользователя на английском языке. Недопустимо использование символов /, #, $. Имя не должно содержать пробелов. Пробелы в начале и в конце «съедаются», для разделения используется _

TREE


Symbols
Alphanumeric
Special
/ $
Other
Other
Length
<Min
Min-Max
>Max

RESULTS

Dimension	Classes	Sample values
Length, char	min-max, inclusive	min, max
	0-min-1	0, min-1
	max+1- infinity	max +1, very big
Symbols	alphanumeric	123abc
	special	!@#$%^&*()_+|-=\{}[]:";'<>?,./
	not English	ÀÇÈÌÑÒÙßàçèìíñò
	double-byte	hieroglyphs
Spaces	no spaces	a
	leading, trailing	_a, _a_, a_
	in the middle	a_a
Unique + Case	different	aaa \ bbb
	exactly the same	aaa\ aaa
	same but in different case	aaa\aAa

LET'S TRY...

Доменное имя должно содержать от двух до 63 символов, начинаться и заканчиваться буквой латинского алфавита или цифрой.
Промежуточными символами могут быть буквы латинского алфавита, цифры или дефис.
Доменное имя не может содержать дефисы одновременно в 3-й и 4-й позициях.

ОБЛАСТИ И КЛАССЫ

- Длина имени:

  0-1, 2-63, 64-∞
- Символы:

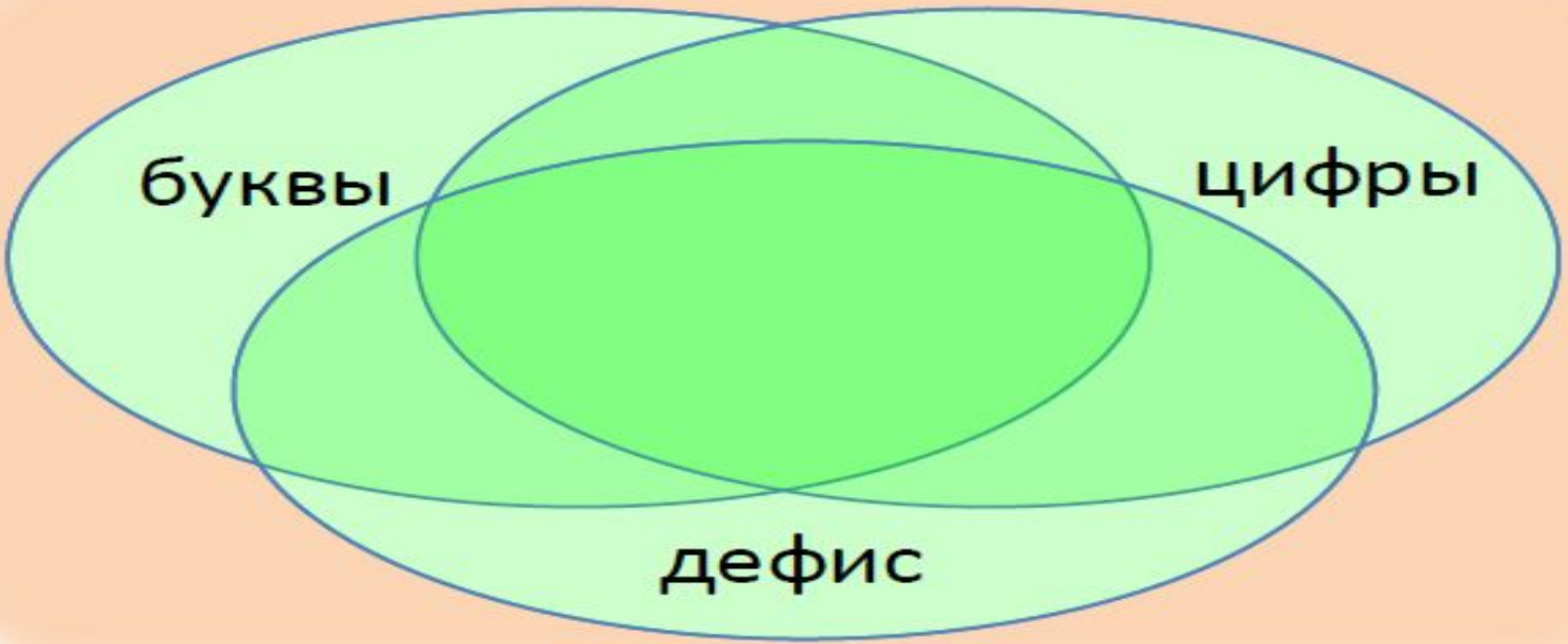
  буквы латинского алфавита, цифры, дефис, все остальные
- Структура

  в начале/конце стоит дефис/не дефис,

  в 3 и 4 позиции дефисы/не дефисы

Интервалы

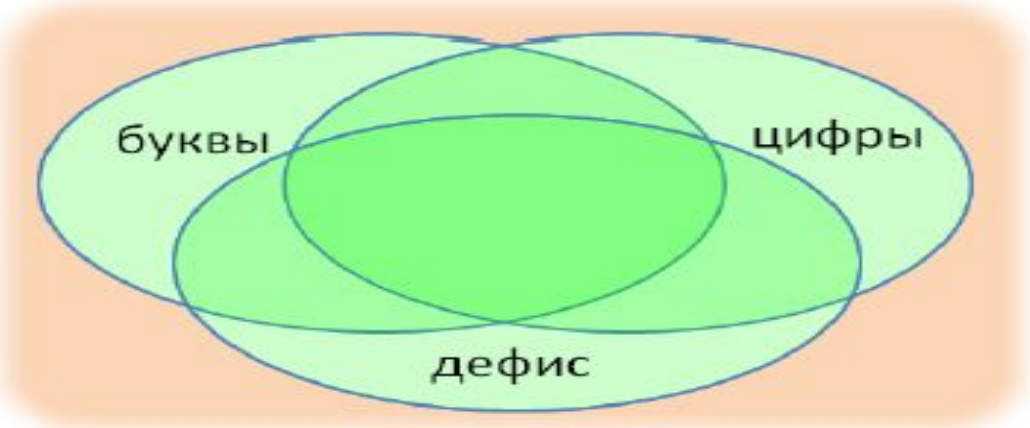
Покрытие и разбиение

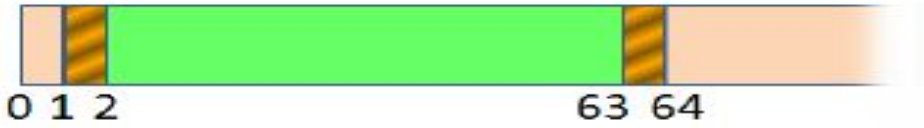


Матрицы

Поз.3 / Поз.4	Дефис	Не дефис
Дефис		
Не дефис		

Поз.4 \ Поз.3	Дефис	Не дефис
Дефис		
Не дефис		





BOUNDARIES

DIFFERENT CASES

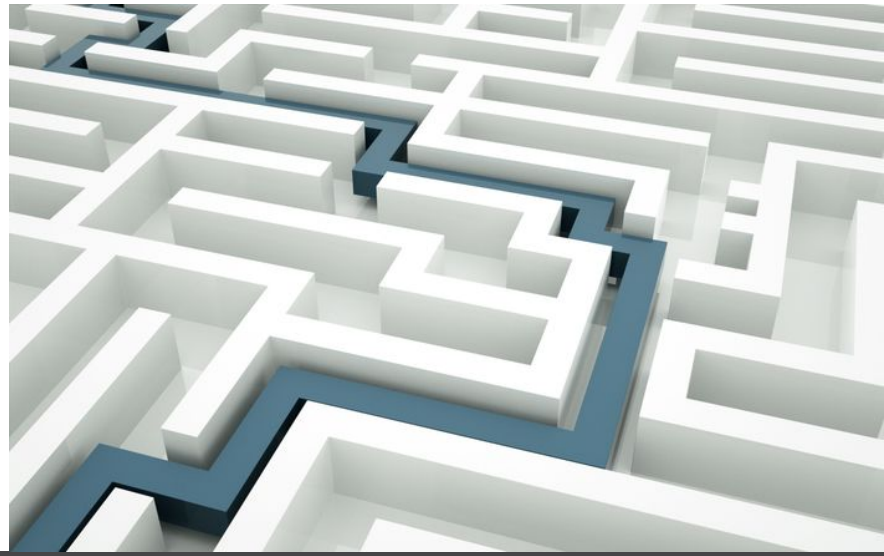
IF WE DO NOT KNOW EXACT BOUNDARIES:

If requirements do not specify allowed length, symbols, format, formula, etc:

- Developers may decline valid bugs
- Application may be inconsistent
- Bugs when not mainstream tests cause system down remain

WE DO NOT KNOW EXACT BOUNDARIES - SOLUTION

1. Create list of items and properties (not fields!)
2. Make reasonable assumptions
   - valid values
   - behavior when values are invalid
   - consult internet
3. Agree with developers
4. Approve with customer



WE DO NOT KNOW EXACT BOUNDARIES - EXAMPLE

Item property	Type	Required?	Unique?	Min	Max	Other (symbols allowed, format, formula,…)
Name	string	yes	yes	1	100	may contain any symbols from supported languages, not case sensitive
Password	string	yes	no	6	50	may contain any symbols from supported languages except spaces, case sensitive

WHEN?

- ✔ Identify ambiguities in descriptions of fields
- ✔ Find biggest / smallest values of a field and catch bugs on boundaries
- ✔ Pick few cases from huge number of equivalent cases

STRENGTH?

- ✔ Find highest probability errors with a relatively small set of tests.
- ✔ Intuitively clear approach, generalizes well

BLIND SPOTS?

- ✔ Errors that are not at boundaries or in obvious special cases.
- ✔ Also, the actual domains are often unknowable

Всем спасибо за внимание!