

Why Testing?

Software Testing

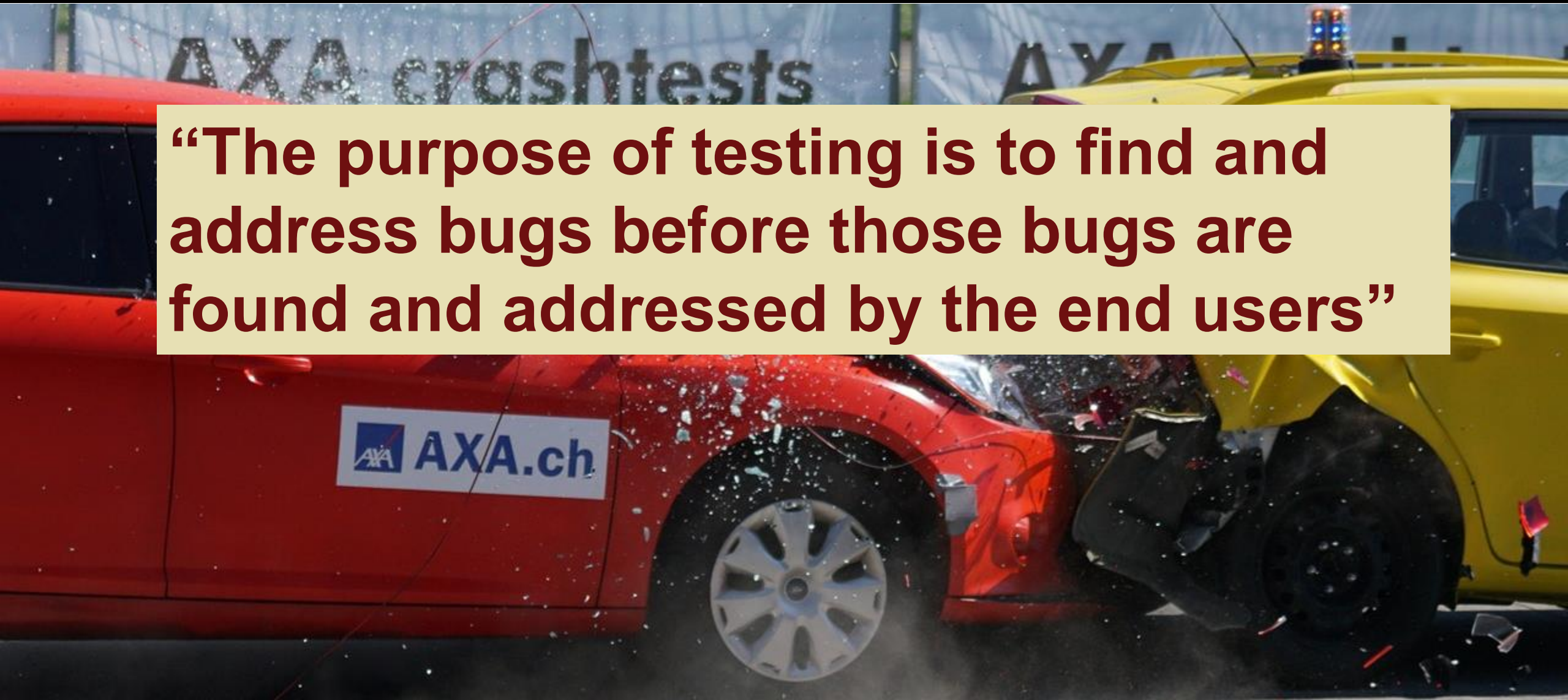
Trainer: Aleksandar Karamfilov
Email: alex@pragmatic.bg
Facebook: facebook.com/a.karamfilov
Copyright © Pragmatic LLC



www.pragmatic.bg

Software Testing

“The purpose of testing is to find and address bugs before those bugs are found and addressed by the end users”



Testing is applied epistemology!

$\Sigma = mc^2$

A close-up photograph of a hammer about to strike a bolt held by a wrench. The hammer's head is dark and worn, positioned above a silver-colored bolt. A black adjustable wrench is clamped around the bolt. The background is a blurred workshop setting with yellow and brown tones.

You don't assure quality by testing!

Beizer's levels of testing mentality

Different meanings of testing

Beizer's levels of testing mentality

No difference between testing and debugging.

```
css
> fonts
> skins
custom.css
ie.css
theme.css
theme-animate.css
theme-blog.css
theme-elements.css
theme-shop.css

83      );
84      'image_src' => $image_src
85  }
86
87
88

95      $color_val == 6 ) {
96          $rgb_array['r'] = hexdec($hex_str);
97          $rgb_array['g'] = 0xFF & ($color_val >> 0x10);
98          $rgb_array['b'] = 0xFF & ($color_val >> 0x8);
99      } elseif( strlen($hex_str) == 3 ) {
100          $rgb_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));
101          $rgb_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));
102          $rgb_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));
103      } else {
104          return false;
105      }
106  }
107  }
108  // Draw the image
109  if( isset($_GET['c']) ) {
110      $return_string = implode($separator, $rgb
111  }
```


Beizer's levels of testing mentality

Purpose of testing is to show that software works.



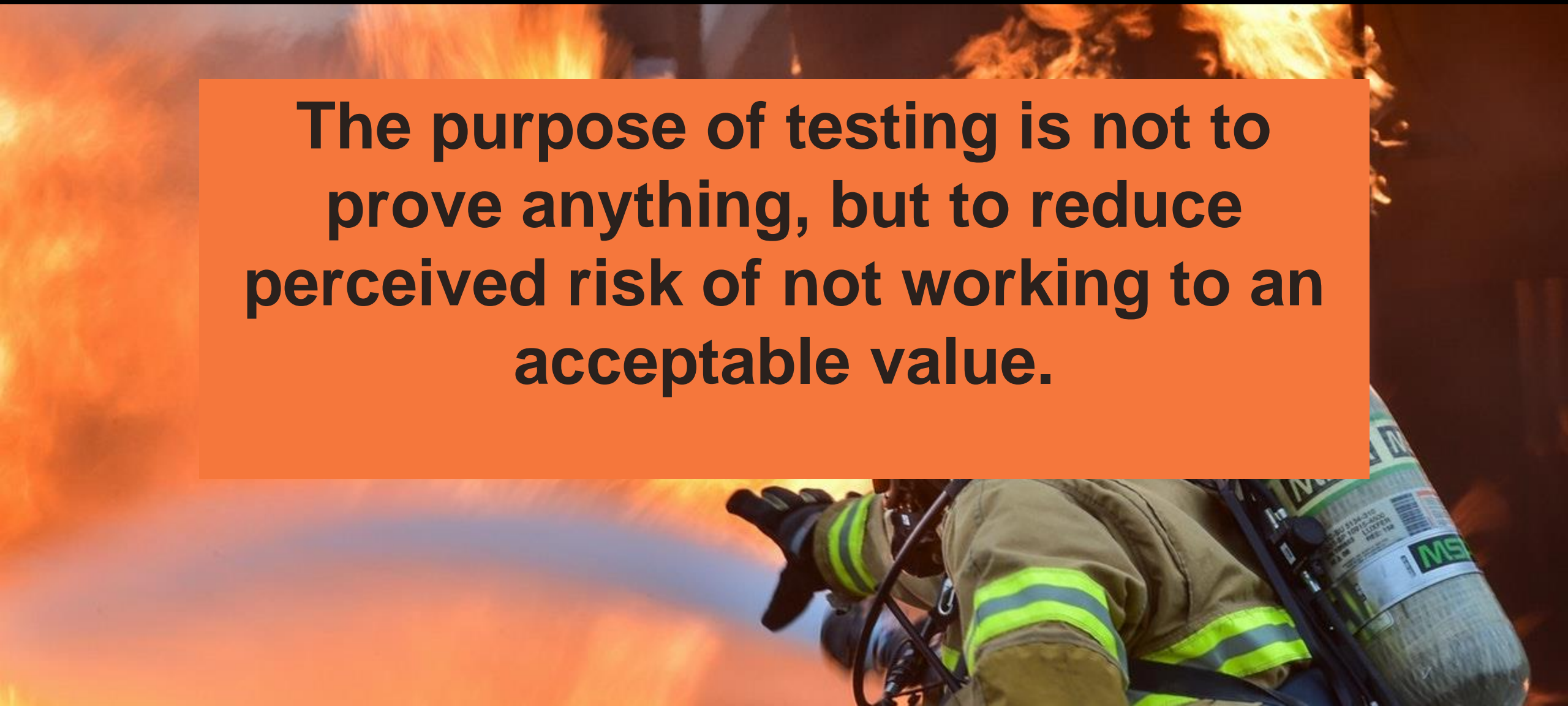
Beizer's levels of testing mentality



Purpose of testing is to show that software does **not** work.

Beizer's levels of testing mentality

The purpose of testing is not to prove anything, but to reduce perceived risk of not working to an acceptable value.



Beizer's levels of testing mentality

Testing is not an act. It is mental discipline that results in low-risk software without much effort.



Testing principles

7 principles of software testing

Software testing principles

- Exhaustive testing is impossible
- Testing shows the presence of bugs
- Early testing
- Defect clustering (80/20)
- The pesticide paradox
- Testing is context dependent
- Absence of errors fallacy

Software Testing Life Cycle

Phases in testing

Software Testing Life Cycle (STLC)

```
graph TD; A[Test planning and control] --> B[Test analysis and design]; B --> C[Test implementation and execution]; C --> D[Evaluating exit criteria and reporting]; D --> E[Test closure activities];
```

Test planning and control

Test analysis and design

Test implementation and execution

Evaluating exit criteria and reporting

Test closure activities

Software Testing Life Cycle (STLC)

Planning

- ⦿ Determine scope
- ⦿ Determine risks
- ⦿ Identify objectives
- ⦿ Select test approach
- ⦿ Use test policy/strategy
- ⦿ Required resources
- ⦿ Schedule tasks
- ⦿ Exit criteria

Analysis

- ⦿ Review test basis
- ⦿ Identify test conditions
- ⦿ Evaluate testability requirements/system

Implementation

- ⦿ Develop/prioritize test cases
- ⦿ Create test suits
- ⦿ Implement/verify environment

Evaluating exit criteria

- ⦿ Check test logs against exit criteria
- ⦿ Assess if more test are needed

Test closure activities

- ⦿ Check which planned deliverables we actually delivered
- ⦿ Finalize and archive testware
- ⦿ Hand over testware to the maintenance organization
- ⦿ Evaluate how the testing went and analyze lessons learned for future releases and projects.

Design

- ⦿ Design tests
- ⦿ Design test environment

Execution

- ⦿ Execute test suits/cases
- ⦿ Log the outcome
- ⦿ Compare actual/expected results
- ⦿ Report discrepancies
- ⦿ Confirmation/re-testing

Reporting

- ⦿ Write a test summary report for stakeholders

Control

- ⦿ Measure/analyse results
- ⦿ Monitor/document progress
- ⦿ Provide information on testing
- ⦿ Initiate corrective actions
- ⦿ Make decisions

Classic example

Weinberg-Myers triangle problem

Triangle problem

Consider the following software program. It reads three input data values. These values represent the three lengths of the sides of a triangle. The purpose of this program is to display a message which states whether the triangle is scalene (i.e., no two sides are equal), isosceles (two sides equal) or equilateral (all sides equal). As software goes, this is very low in complexity.

How do we test it?

Create 10 tests to test the program

[illegible]

Example

Side 1	Side 2	Side 3	Expected Result
1	1	1	Equilateral
10.5	10.5	10.5	Equilateral
2	3	4	Scalene
4	2	2	Scalene
4	3	2	Scalene
10	10	20	Isosceles
20	10	10	Isosceles
-1	1	1	Error
1		1	Error
A	B	C	Error

Questions

