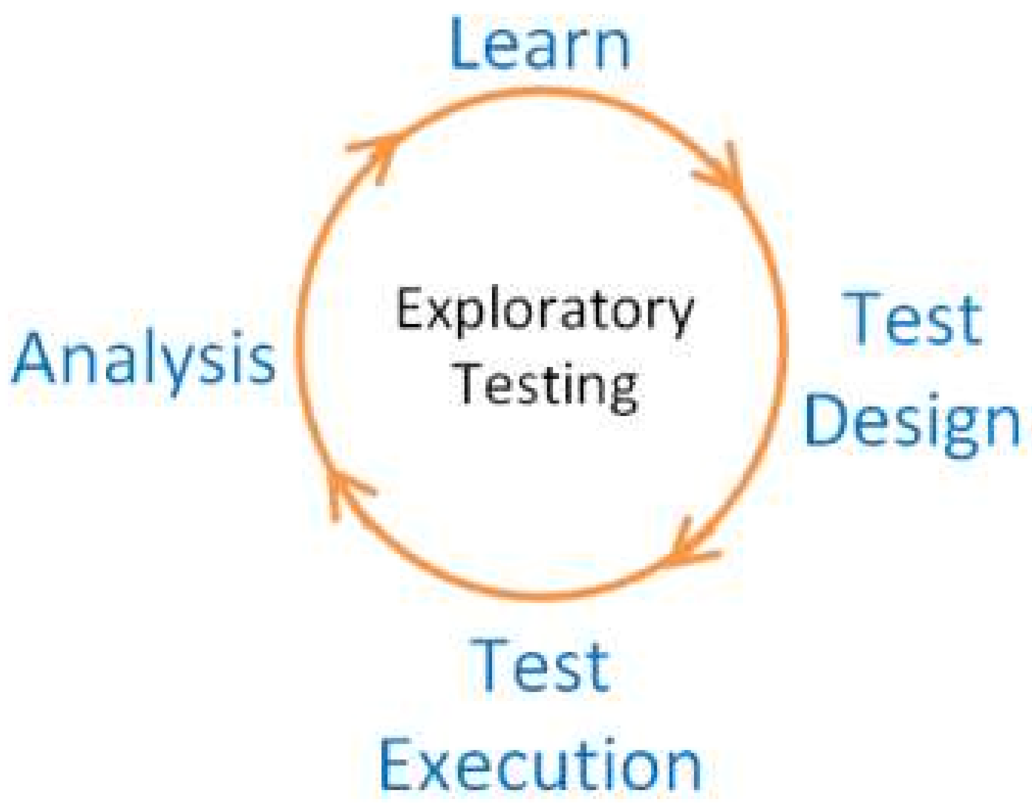
Oracles and heuristics

Lector: Viktor Slavchev

E-mail: mr.slavchev@gmail.com
Twitter:@TheTestingTroll
Blog: mrslavchev.com

www.pragmatic.bg

ET diagram


Learn
Test Design
Test Execution
Analysis
Exploratory Testing

Task - Jigsaw

Main quest:
Complete the jigsaw with Shrek:
https://goo.gl/MLsrhV

Side quest:
Take notes on your strategy.

Time to complete: 10 min.

Techniques used in solving jigsaw

- Make all pieces visible
- Find the corner pieces
- Compare with image
- Build the edges
- Build distinct features (eyes, nose, fingers)
- Match patterns where we don’t understand
- Match color

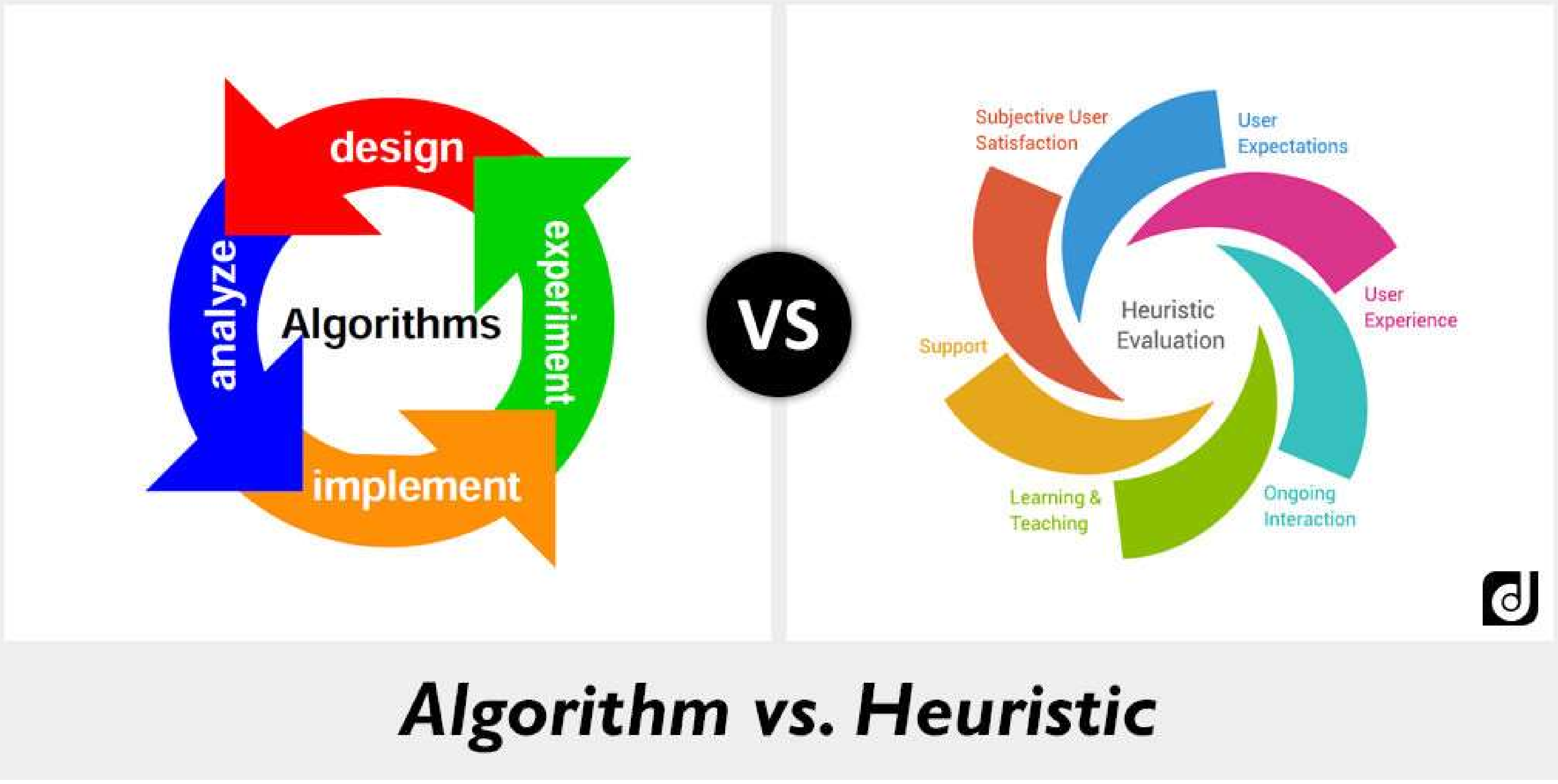
Building bigger pieces helps us solve faster.

Testing heuristics

- *"Heuristic is any approach to problem solving, learning, or discovery that employs a practical method not guaranteed to be optimal or perfect, but sufficient for the immediate goals"*
- Or
  "A heuristic is a fallible method for solving a problem or making a decision"

When we know how to solve a problem, we might follow a rule. When we don't know how to solve it, we try different heuristics.

Algorithm vs. heuristic



IMPORTANT !!!

- Heuristics are fallible
- Don’t treat them as “best practices”
- Conclusions we make need to be tested as well.
- We rarely can use one heuristic to solve a problem, we use combination of them.
- Do not assume, but rather make an educated guess

Assumptions

Assumptions are not heuristics, they have to be tested

Common heuristics

From everyday life

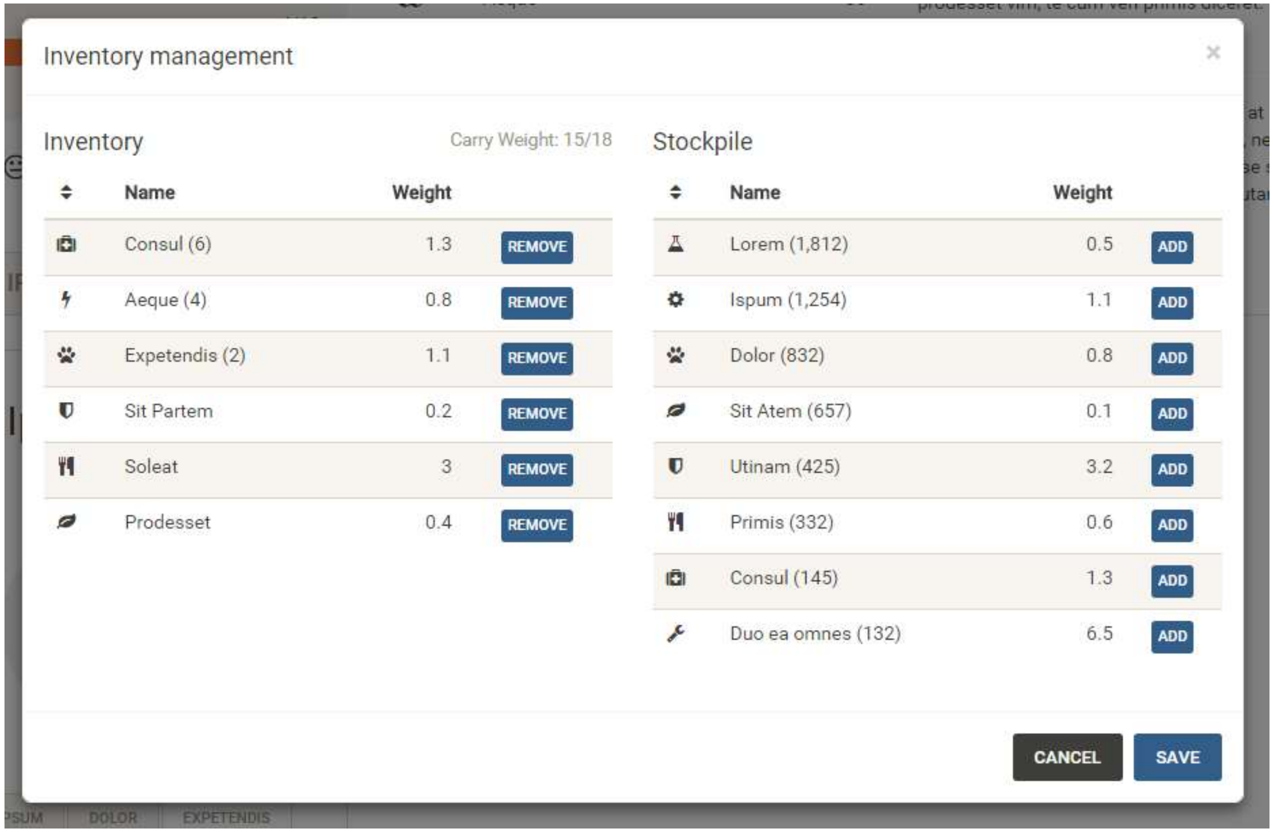
- Rule of thumb
- Guesstimation
- Educated guess
- Stereotyping
- Common sense
- Trial and error

Testing heuristics

Examples

Ex. - "Beginning, middle, end"

- Try to append or remove element in the beg/mid/end of a collection



Inventory management

Inventory — Carry Weight: 15/18

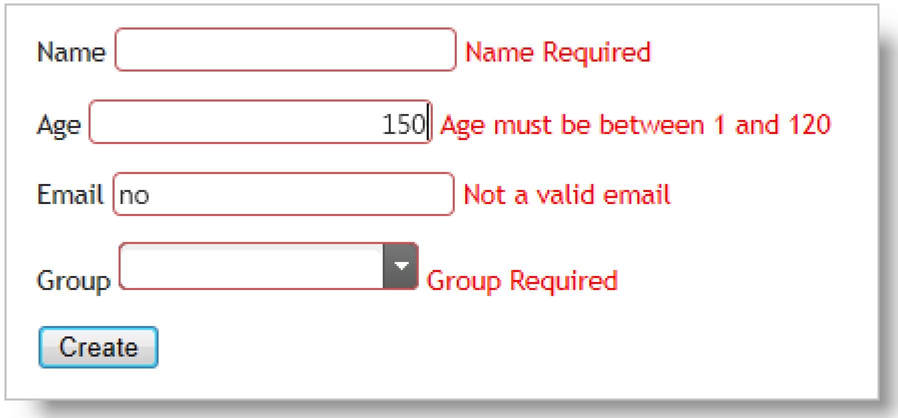
Name	Weight	
Consul (6)	1.3	REMOVE
Aeque (4)	0.8	REMOVE
Expetendis (2)	1.1	REMOVE
Sit Partem	0.2	REMOVE
Soleat	3	REMOVE
Prodesset	0.4	REMOVE

Stockpile

Name	Weight	
Lorem (1,812)	0.5	ADD
Ispum (1,254)	1.1	ADD
Dolor (832)	0.8	ADD
Sit Atem (657)	0.1	ADD
Utinam (425)	3.2	ADD
Primis (332)	0.6	ADD
Consul (145)	1.3	ADD
Duo ea omnes (132)	6.5	ADD

Ex. "Goldilocks"

- Too big, too small or just right - might be used for all types of ranges - dates, numbers, strings.



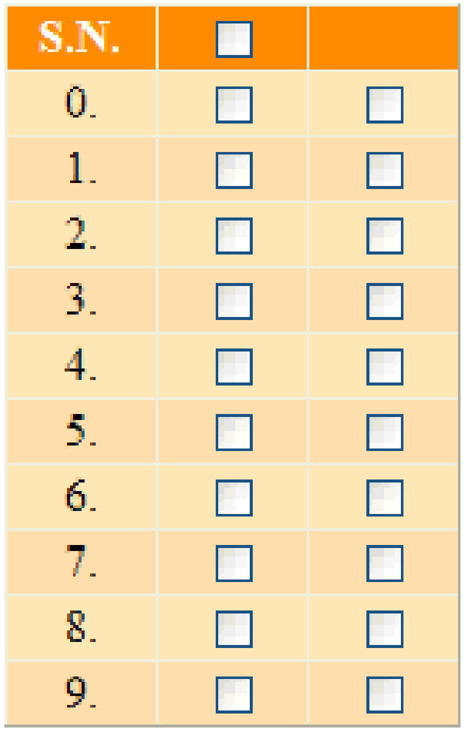
Name		Name Required
Age	150	Age must be between 1 and 120
Email	no	Not a valid email
Group		Group Required

Ex. “Interrupt”

- Interrupt the system doing important stuff, turn it off, turn off internet, force it to shut down, kill the app, etc.

Ex. “Select none, all, many”

- when applying actions to a list of items
- select all
- select none
- select many of them
- select few of them



S.N.	☐	
0.	☐	☐
1.	☐	☐
2.	☐	☐
3.	☐	☐
4.	☐	☐
5.	☐	☐
6.	☐	☐
7.	☐	☐
8.	☐	☐
9.	☐	☐

Testing heuristics examples

- *“Never and always”* - find out places in requirements/docs/help where word “always” and “never” are used. Challenge them.
- *CRUD* - create/read/update/delete - invoke these actions in conjunction with any other heuristic.
- *“Reverse”* - undo your actions one by one, see if sessions are persisted, if values are not changed or lost.

Testing oracle heuristic

Question

Context: You are testing an application (desktop, web or mobile). You observe weird behavior.

Question: How do you make sure it is a bug? What or who you can consult?

Try to name 5 ways to validate if what you see is correct behavior or a bug.

Time: 5 mins

Testing oracles heuristic

- Testing oracle is a “a means by which we recognize a problem that we encounter during testing” (Bolton)
- Helps us to model ideas how application should work.
- Oracle creation can be based on:
  - Feelings and mental models
  - Artifacts - documents, system requirements, design docs
  - Experience - previous encounters with similar/same issues
  - Conference - conversation with stakeholders
  - Inconsistencies with similar products.

Testing oracles

- From scientific perspective - oracles are our “hypothesis” that we aim to prove wrong by experimenting.
- Good series of articles on testing oracles by Michael Bolton:
  Oracles from the Inside Out, Part 1: Introduction
  Oracles from the Inside Out, Part 2: Experience, Mental Models, and Feelings
  Oracles From the Inside Out, Part 3: From Experience Directly to Conference

HICCUPPS

Example for oracle might be the consistency oracle, used when we doubt if what we found is a bug.

Software should be consistent with:

- **History** - previous versions
- **Image** - the image that company is trying to build
- **Comparable products -** reference oracle
- **Claims** - product claims to do what?
- **User's expectations/desires** - what the user wants
- **Product itself** - internal consistency
- **Purpose** - what it is intended to do
- **Statutes/Standards** - legal compliance

...and its extension

FEW HICCUPS

Familiarity - inconsistency with bugs we already know

Explainability - system should be easy to explain to ourselves

World - system should be consistent of what we know in the world

Read more at: [FEW HICCUPS by M. Bolton]

Assignment

Weinberg-Myers triangle problem

The program reads three integer values from an input dialog. The three values represent the lengths of the sides of a triangle. The program displays a message that states whether the triangle is scalene (no equal sides), isosceles (two equal sides), or equilateral (three equal sides).

List possible input values and expected results. As much as you can:

Input - a, b, c	Expected result
3, 3, 5	isosceles
5, 5, 5	equilateral

Quest - Triangle calc

Now try it on this application

https://goo.gl/Z1n5Fz

http://www.3eck.org/triangle/en/calculator_simple.php

Use the cases you wrote. Are they relevant? What weird things do you see?

For the next lecture

- Try to bring your laptop, if you can’t, let me know before the beginning.
- You will need something to take notes.
- Try to be accurate.
- Review the current lecture and the additional materials.

Questions