

Contemporary Exploratory Testing Foundations

 CC BY-NC-SA 4.0

Maaret Pyhäjärvi

November 2025

CGI



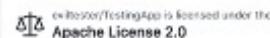
How Would You Test This? **E-Primer**

by Alan Richardson

<https://qe-at-cgi-fi.github.io/ETF2025/app/>

<https://github.com/QE-at-CGI-FI/pft-eprimer-agentic-github-copilot-demo>

© 2025 CGI Inc.



Apache License 2.0

A permissive license whose main conditions require preservation of copyright and license notices. Contributors provide an express grant of patent rights. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

Permissions

- ✓ Commercial use
- ✓ Modification
- ✓ Distribution
- ✓ Patent use
- ✓ Private use

Limitations

- ✗ Trademark use
- ✗ Liability
- ✗ Warranty

Conditions

- ① License and copy
- ② State changes

This test target is from collections of [Alan Richardson](#), [eviltester](#), a brilliant exploratory tester.

E-Primer an e-prime checking tool

Do you want to write without using the verb "to be"?

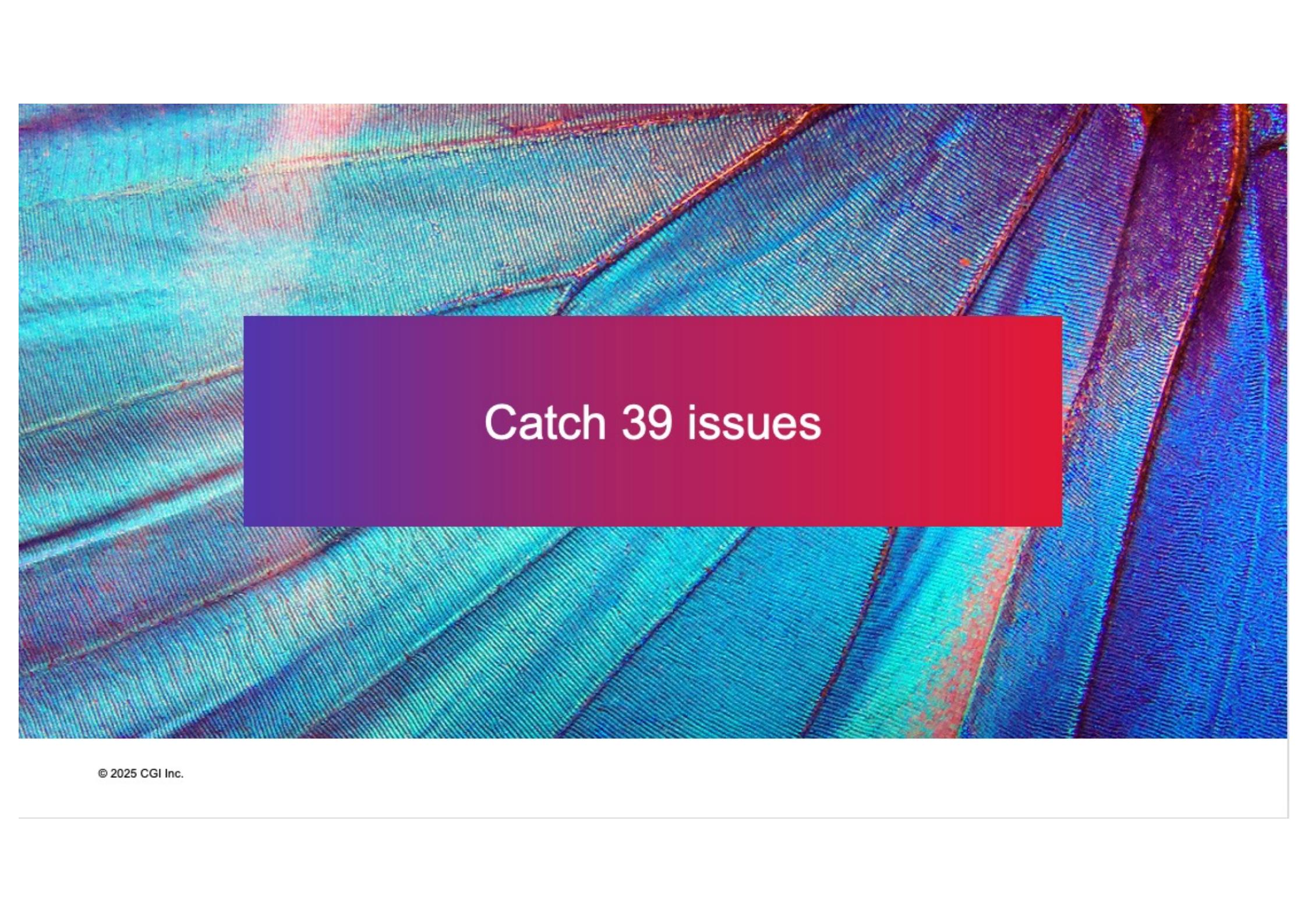
Do you want to master [e-prime](#)?

Use our online tool to check your writing.

- Word Count:
- Discouraged Words:
- Possible Violations:

Text:

[Check For E-Prime](#)



Catch 39 issues



Contemporary Exploratory Testing Foundations

Maaret Pyhäjärvi

v. 2.4 (2025-11-17)



by [Maaret Pyhäjärvi](#) is licensed under [CC BY 4.0](#)



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Optimizing the value of testing

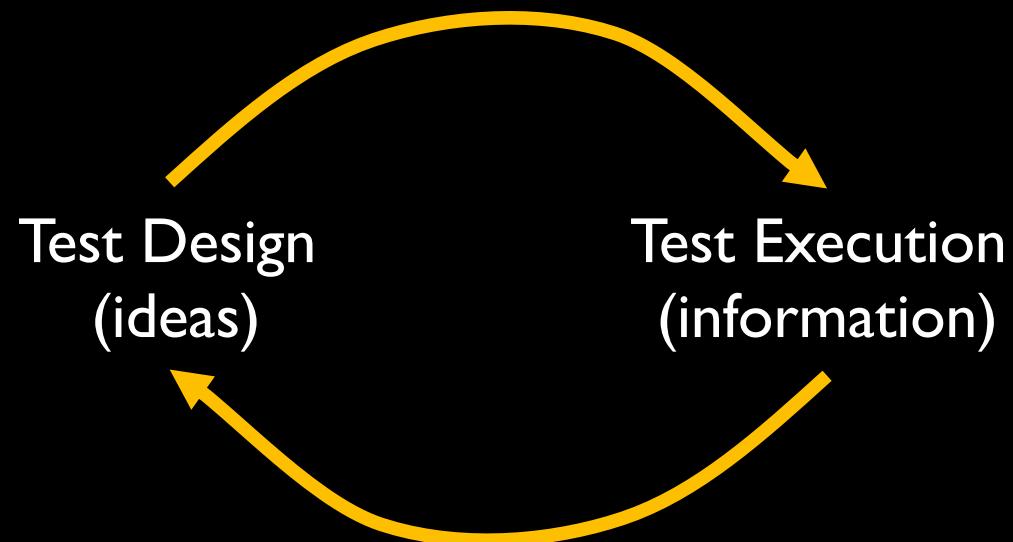


<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Learning



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Exploratory Testing the Verb



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to



Input

Tester

Domain knowledge

Requirements and specifications

Testing knowledge

Miscellaneous knowledge



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to



Output

Better tester

Coverage

Information incl. defects and change requests

Documentation: Strategy

Documentation: Tests



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Course Outline

Chapter 1: Test target and our options for exploring
Chapter 2: Self-management basics on setting yourself
constraints
Chapter 3: The moment of first impression
Chapter 4: Recognizing and learning a domain
Chapter 5: Recognizing functionality
Chapter 6: Recognizing data
Chapter 7: Recognizing application and execution
environment
Chapter 8: Documenting in a mindmap

Chapter 9: pytest the very basics
Chapter 10: Documenting as skeleton test automation
Chapter 11: Playwright library and CSS selectors on web
pages
Chapter 12: Documenting as executable test automation
Chapter 13: Why this is not about any specific tool
Chapter 14: Use of time
Chapter 15: Coverage
Chapter 16: Test Strategy
Chapter 17: Full results and reproducing from customer
feedback
Chapter 18: Closing remarks

The Self-Study
Edition



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Course Outline

Section I:

Options for Exploring

1

3

Section II:

Control through Choices

2

Section III:

Documenting (with Automation)

4-13

Extending with Function, Data,
Environment and Domain

Section IV:

Use of time and coverage

14-17

The Classroom
Edition



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Test Target and Our Options for Exploring

Chapter I



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

The screenshot shows a web browser window with the URL exploratorytestingacademy.com/app/. The page content includes:

- A header section with a back arrow, forward arrow, and refresh icon.
- A Apache License 2.0 badge and text describing the license.
- A table with three columns: Permissions, Limitations, and Conditions.
- A text block stating: "This test target is from collections of [Alan Richardson](#), [eviltester](#), a brilliant exploratory tester."
- A section titled **E-Primer an e-prime checking tool**.
- Text asking: "Do you want to write without using the verb 'to be'?"
- Text asking: "Do you want to master e-prime?"
- Text asking: "Use our online tool to check your writing."
- A bulleted list: "Word Count:", "Discouraged Words:", and "Possible Violations:".
- A "Text:" label followed by a text input area.
- A "Check For E-Prime" button.



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Stop-and-Think: Options for Exploring

What would you do first, and soon after you get started?

List all things that come to your mind about how you could test this. What would you start from? What you would not do?



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Options for Exploring

Research the Domain

Use test target *with a constraint*



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Self-management Basics on Setting Yourself Constraints

Chapter 2



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Charters

Charter template

- *target*: where you're exploring
- *resources*: what you're using/how you're exploring
- *information*: what question you want to answer

*Elizabeth Zagroba's concise template adapted
from Elizabeth Hendrickson's template*



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Choose Your Own Constraint

Deliberately excluding perspectives!
Never Be Bored!



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Explore with Intent

INTENT

Mission

Charter

Other
Charters

Details

LEARNINGS



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Stop-and-Think: Charters, Constraints, Intent

You're approaching the moment of first impression. How do you want to frame your moment of first impression?



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

The Moment of First Impression

Chapter 3



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Options Expire

Capture First Impression

Borrow someone else's First Impression

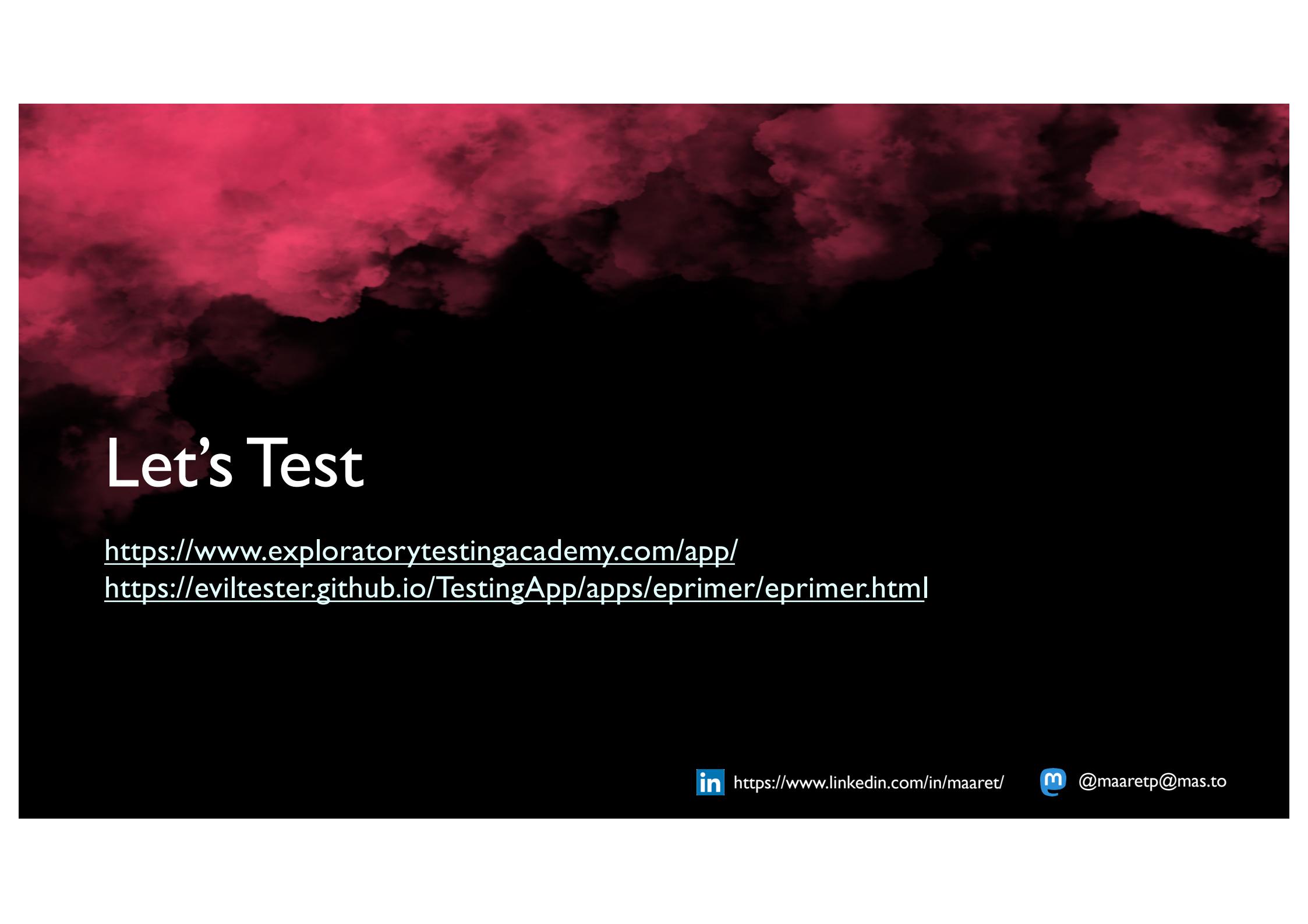
Timing of feedback changes reaction to it!



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to



Let's Test

<https://www.exploratorytestingacademy.com/app/>

<https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html>

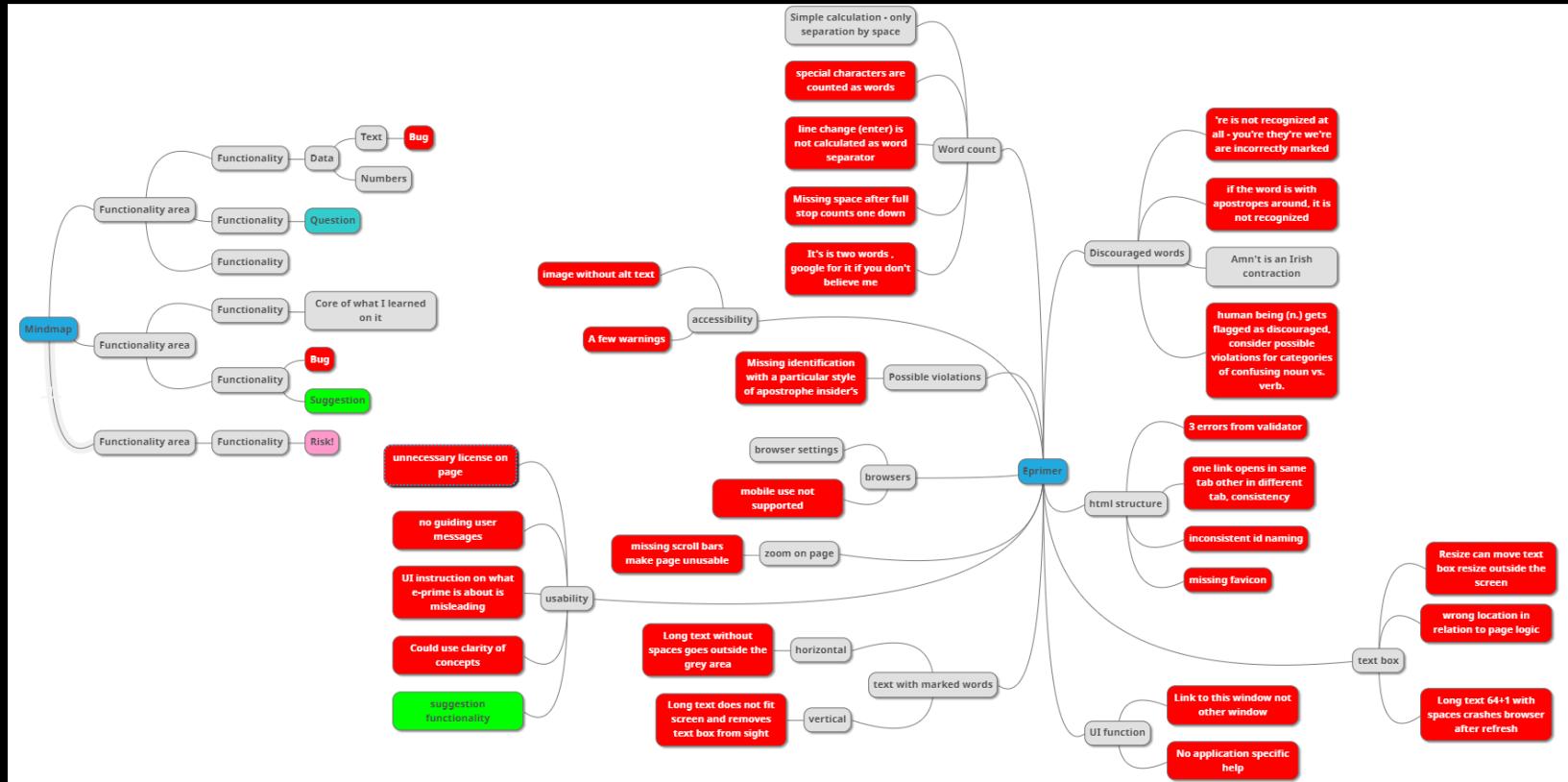


<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Example: Test Results, Red is Bug



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Bugs are Conversation Starters

Bug is anything that might bug a user.
You start conversations about defects and
change requests.



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Recognizing and Learning a Domain

Chapter 4



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Conference Reference Inference



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to



eviltester/TestingApp is licensed under the
Apache License 2.0

A permissive license whose main conditions require preservation of copyright and license notices.
Contributors provide an express grant of patent rights. Licensed works, modifications, and larger
works may be distributed under different terms and without source code.

Permissions

- ✓ Commercial use
- ✓ Modification
- ✓ Distribution
- ✓ Patent use
- ✓ Private use

Limitations

- ✗ Trademark use
- ✗ Liability
- ✗ Warranty

Conditions

- ⓘ License and copyright notice
- ⓘ State changes

This test target is from collections of [Alan Richardson](#), [eviltester](#), a brilliant
exploratory tester.

E-Primer an e-prime checking tool

Do you want to write without using the verb "to be"?

Do you want to master [e-prime](#)?

Use our online tool to check your writing.

- Word Count: 9
- Discouraged Words: 3
- Possible Violations: 1

To be or not to be is Hamlet's dilemma

Text:

To be or not to be is Hamlet's dilemma

Check For E-Prime



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

exploratorytestingacademy.com/app/eprime.js

```
function inEPrimeOutputFormat(aWord){  
    return '<span class="ep_violation">' + aWord + "</span>";  
}  
  
function inImpossibleEPrimeOutputFormat(aWord){  
    return '<span class="ep_warning">' + aWord + "</span>";  
}  
  
function isDiscouragedWord(aWord){  
  
    var discouragedWords = new Array();  
    discouragedWords['be'] = 'be';  
    discouragedWords['being'] = 'being';  
    discouragedWords['been'] = 'been';  
    discouragedWords['am'] = 'am';  
    discouragedWords["isn't"] = "isn't";  
    discouragedWords["are"] = "are";  
    discouragedWords["aren't"] = "aren't";  
    discouragedWords["was"] = "was";  
    discouragedWords["wasn't"] = "wasn't";  
    discouragedWords["were"] = "were";  
    discouragedWords["weren't"] = "weren't";  
    discouragedWords["is"] = "is";  
    discouragedWords["ain't"] = "ain't";  
    discouragedWords["i'm"] = "i'm";  
    discouragedWords["amn't"] = "amn't";  
  
    return (discouragedWords[aWord.toLowerCase()] == aWord.toLowerCase());  
}
```



<https://www.linkedin.com/in/maaretp>



@maaretp@mas.to



Let's Test

<https://www.exploratorytestingacademy.com/app/>

<https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html>



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Learning of Domain of E-Primer

Core Idea	Writing English language avoiding verb “be” in all its forms
Why?	Someone claims it had benefits, intellectual challenge
Examples	Used in sentences Listed examples
Sample texts	The Bible!



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Recognizing Functionality

Chapter 5



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Naming of Function

Functions in Code
Expected Features
Visible Features



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to



Let's Test

<https://www.exploratorytestingacademy.com/app/>

<https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html>



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Learning of Function of E-Primer

Input	Text field and button
Output	Three numbers, text area
Containers	Resizable text field, resizable browser window, page
Presentation	Fonts, text and element sizes, order of functions
Browser	Settings, zoom
Algorithm	Recognizing eprime



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Recognizing Data

Chapter 6

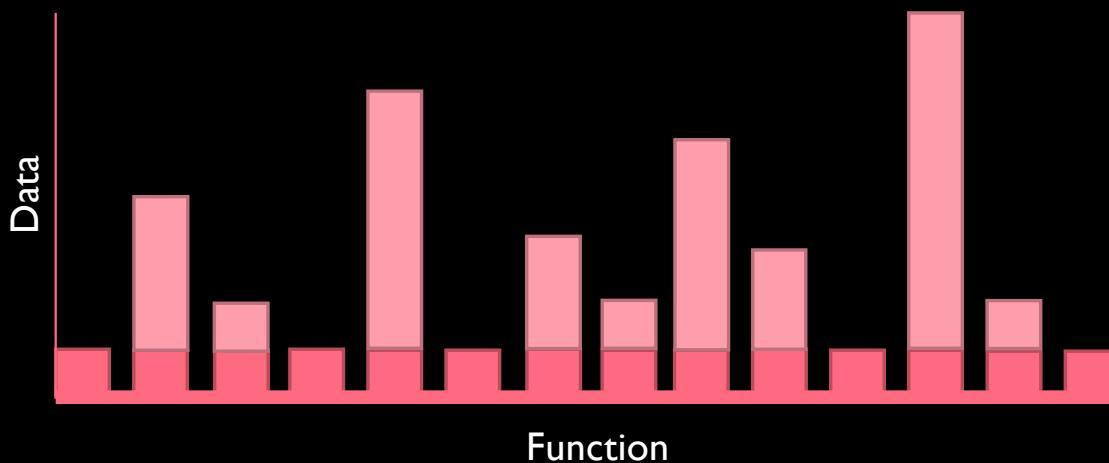


<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Data or Variables



Versatile Data

Lifecycle of Data: Create, Read, Update, Delete
Known problematic inputs: GitHub Naughty Strings

<https://github.com/minimaxir/big-list-of-naughty-strings/blob/master/blns.txt>



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to



Let's Test

<https://www.exploratorytestingacademy.com/app/>

<https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html>



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Learning of Data of E-Primer

Word delimiter	Space, wordcount breaks with characters and line change
Types of apostrophes	Typesetter / typewriter
Long text	Copied / tool generated
Valid eprime	Recognizing right as right
Eprime violations	Recognizing wrong as wrong



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Recognizing Application and Execution Environment

Chapter 7



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

What You Coded is a Bad Constraint



Naomi Wu 机械妖姬
@RealSexyCyborg

...

You can't say "Signal is secure, it's the OS that's not" if Signal cannot operate without an OS. They are a system-can only be used as a system, they need to be evaluated as a system, and their effectiveness as a system disclosed to customers.

3:58 AM · Jan 16, 2021 · Twitter Web App



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Execution Environment

Different browsers: web and mobile

Browser functionality and add-ons

HTML standard compatibility

Accessibility standard compatibility



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to



Let's Test

<https://www.exploratorytestingacademy.com/app/>

<https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html>



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Learning of Application and Execution Environment of E-Primer

Browser	Chrome, Brave, ...
Screen size	Web, Mobile
Browser Settings	Zoom, Security, ...
Add-ons	BugMagnet
Validators	HTML, Accessibility, ...



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Documenting in a Mindmap

Chapter 8

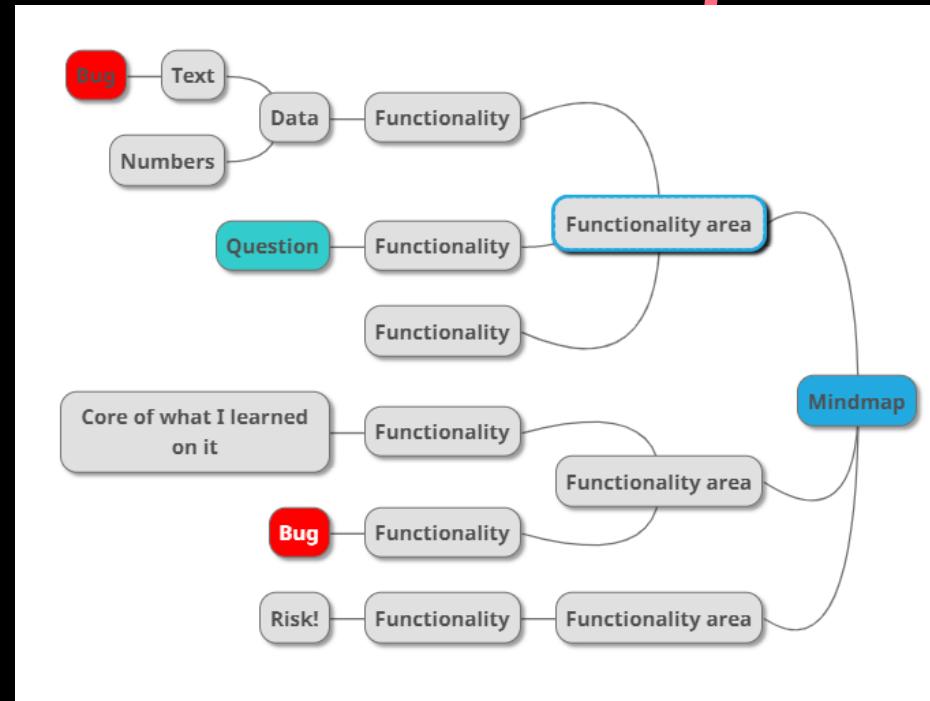


<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Mindmap



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Cem Kaner. Bug Reporting Heuristic.

Bug Reports

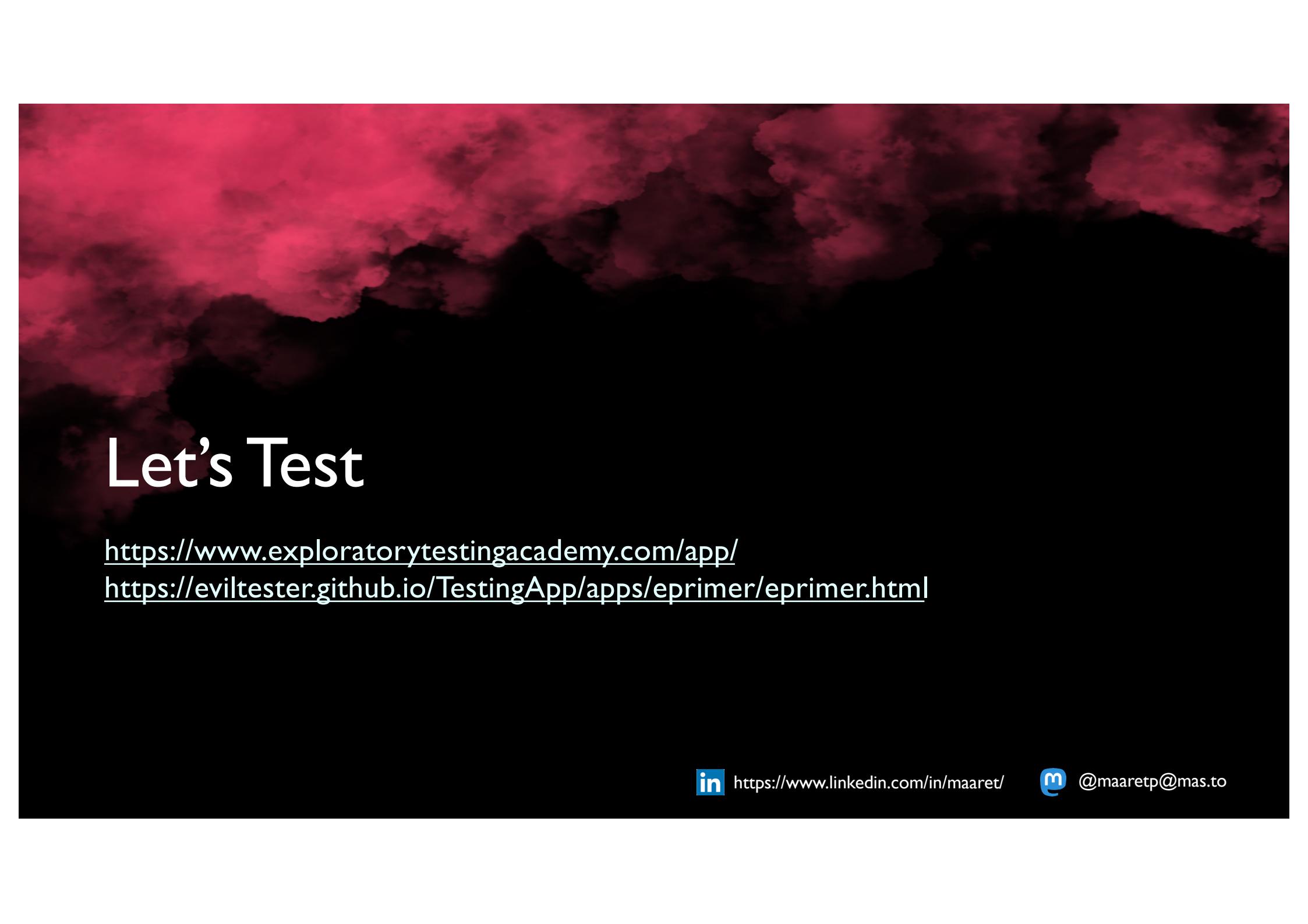
R eplicate
Q uolate
M azimize
G eneralize
E xternalize
N eutral tone



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to



Let's Test

<https://www.exploratorytestingacademy.com/app/>

<https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html>



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Mindmapping as Future Reference

Notetaking in the moment

Restructure as you learn

Documentation for the future

General purpose mindmaps



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to



pytect the Very Basics

Chapter 9



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

pytest

Popular test runner for python language
Used both for unit tests and orchestrated tests
Integrates with libraries in python ecosystem



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Documenting as Skeleton Test Automation

Chapter 10



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Document in Context of Code

```
def this_is_a_test():
    #first thing to do
    #second thing to do
    #third thing to do
    pass
```

```
===== test session starts =====
platform win32 -- Python 3.10.2, pytest-7.0.0, pluggy-1.0.0
rootdir: C:\avimet\avimet-bdd, configfile: pytest.ini
plugins: anyio-3.5.0, approvaltests-0.2.3, base-url-1.4.2, bdd-5.0.0, playwright-0.2.2
collected 1 item

test.py .

===== 1 passed in 0.02s =====
PS C:\avimet\avimet-bdd\tests> █
```

Scenario: Eprime analysis

* Runnable with pytest-bdd

Given the eprime page is displayed

When user analyses sentence to be or not to be

Then user learns sentence has 2 be-verbs, 0 possible be-verbs and total 6 words



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to



Let's Test

<https://www.exploratorytestingacademy.com/app/>

<https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html>



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Skeleton Test Automation

Stepwise Test Cases as
Automation Placeholders

Like test cases but version
controlled as code

Handoff to a task that is
decomposing testing
differently



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Playwright Library and css Selectors on Web Page

Chapter 11



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Playwright Library

Browser driver by Microsoft
Speed – Reliability – Visibility
Automatic waits

```
from playwright.sync_api import Page
```



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Library / Methods

```
1 import pytest
2 from playwright.sync_api import Page
3
4 def test_example(page: Page):
5     page.goto("https://www.exploratorytestingacademy.com/app/")
6     page.screenshot(path="example.png")
```

<https://playwright.dev/python/docs/intro>



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

css selectors

css=

#id

.class

tag

[attribute='value']

[part_of_attribute_value_contains*='value']



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Documenting as Executable Test Automation

Chapter 12



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to



Let's Test

<https://www.exploratorytestingacademy.com/app/>

<https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html>



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

first_test.py ×

```
1  from playwright.sync_api import Page
2
3
4 ► ⌄ def test_example(page: Page):
5     page.goto("https://www.exploratorytestingacademy.com/app/")
6     page.fill("#inputtext", "To be or not to be - Hamlet's dilemma")
7     page.click("#CheckForEPrimeButton")
8     # it should really be 8, but it's not
9     assert page.inner_text("#wordCount") == "9" 
10    assert page.inner_text("#discouragedWordCount") == "2"
11    assert page.inner_text("#possibleViolationCount") == "1"
```

Run:

Python tests for first_test.test_example ×



Tests passed: 1 of 1 test – 3 sec 622 ms



Test Results

3 sec 622 ms

C:\pythonProject\eprime-bdd\venv\Script

Testing started at 21:56 ...

C:\Users\mapyh\AppData\Local\JetBrains

from distutils import version

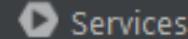
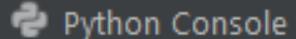
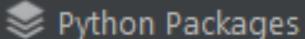
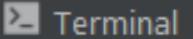
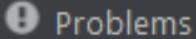
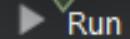
Launching pytest with arguments first_

===== test ses

collecting ... collected 1 item

first test.py::test example[chromium]

Tests passed: 1



Tests passed: 1 (a minute ago)

```
eprime_tests.py ×
1 import pytest
2 from playwright.sync_api import Page
3
4 url = "https://www.exploratorytestingacademy.com/app/"
5
6 def this_is_file(text):...
7
8 @pytest.mark.parametrize('input_text, expect_wordcount, expect_discouraged, expectViolation',
9 [
10     ("To be or not to be - Hamlet's dilemma", 8, 2, 1),
11     ("nothing", 1, 0, 0),
12     ("to be or not to be", 6, 2, 0),
13     ("", 0, 0, 0),
14     ("be, being, been, am, is, isn't, are, aren't, was, wasn't, were, and weren't.", 17, 12, 0),
15     ("\n", 0, 0, 0),
16     ("I'm, you're, we're, they're, he's, she's, it's, there's, here's, where's, how's, what's, who's, aint's, that's.", 30, 4, 11),
17     ("first\nsecond", 2, 0, 0),
18     ("ain't, amn't", 4, 2, 0),
19     ("Hanna's Esa's Meena's Süess-O'Reggie's or Okechukwu's", 6, 0, 5),
20     ("'be'", 1, 1, 0),
21     ("human being", 2, 0, 0),
22     ("typewriter's apostrophe", 2, 0, 1),
23     ("typesetter's apostrophe", 2, 0, 1),
24     (1000 * "x", 1, 0, 0),
25     (this_is_file('sample'), 508, 2, 0),
26     (this_is_file('bible'), 31172, 21, 1),
27     ('cat sat\n', 2, 0, 0)
28 ],
29 ids=['demo', 'not eprime', 'hamlet', 'empty', 'be in forms', 'newline', 'contractions', 'newline with words', 'slang',
30 'possessive', 'quoted be', 'not verb', 'typewriters apostrophe', 'typesetters apostrophe', 'long word', 'file', 'bible', 'ending newline']
31 )
32
33 def test_parametrized_test(page: Page, input_text, expect_wordcount, expect_discouraged, expectViolation):
34     page.goto(url)
35     page.fill("#inputtext", input_text)
36     page.click("#CheckForEPrimeButton")
37     assert page.inner_text("#wordCount") == str(expect_wordcount)
38     assert page.inner_text("#discouragedWordCount") == str(expect_discouraged)
39     assert page.inner_text("#possibleViolationCount") == str(expectViolation)
```

18 tests,
3 browsers

Run:



Python tests for eprime_tests.test_parametrized_test

	✓		✗																	
	✗	✗	Test Results																	
			✗	eprime_tests																
				✗	test_parametrized_test															
					✗	(chromium-demo)														
						✓	(chromium-not eprime)													
						✓	(chromium-hamlet)													
						✓	(chromium-empty)													
						✗	(chromium-be in forms)													
						✗	(chromium-newline)													
						✗	(chromium-contractions)													
						✗	(chromium-newline with words)													
						✗	(chromium-slang)													
						✗	(chromium-possessive)													
						✗	(chromium-quoted be)													
						✗	(chromium-not verb)													
						✓	(chromium-typewriters apostrophe)													
						✗	(chromium-typesetters apostrophe)													
						✓	(chromium-long word)													
						✓	(chromium-file)													
						✓	(chromium-bible)													
						✓	(chromium-ending newline)													

Tests failed: 10, passed: 8 of 18 tests – 8 sec 470 ms

C:\pythonProject\eprime-bdd\venv\Scripts\

Testing started at 22:29 ...

C:\Users\mapyh\AppData\Local\JetBrains\Py

from distutils import version

Launching pytest with arguments eprime_te

===== test session

collecting ... collected 18 items

eprime_tests.py::test_parametrized_test[0]

eprime_tests.py:10 (test_parametrized_te

9 != 8

Expected :8

Actual :9

<Click to see difference>

page = <Page url='https://www.exploratory

input_text = "To be or not to be - Hamlet

expect_discouraged = 2, expectViolation

@pytest.mark.parametrize('input_text',

[

("To be or not to be - Hamlet's o

("nothing", 1, 0, 0),

10 bugs

Requires pytest-bdd +
refactoring to step-
methods

```
1 ► Feature: Eprime text analysis
  2   As a user,
  3     I want to verify my text for violations of eprime,
  4     So I learn to write proper English
  5
  6 ► Scenario Outline: Eprime samples are correctly analyzed
    Given the eprime page is displayed
    When user analyses sentence <sentence>
    Then user learns sentence has <count_certain> be-verbs, <count_possible> possible be-verbs and total <count_total> words
  10
  11 Examples:
  12 | sentence           | count_certain | count_possible | count_total |
  13 | to be or not to be - Hamlet's dilemma | 2           | 1             | 8           |
  14 | cat is hat         | 1           | 0             | 3           |
  15 | nothing            | 0           | 0             | 1           |
  16 | be, being, been, am, is, isn't       | 6           | 0             | 7           |
  17 | are, aren't, was, wasn't, were, weren't | 6           | 0             | 9           |
  18 | I'm, you're, we're, they're, he's, she's | 6           | 0             | 12          |
  19 | it's, there's, here's, where's, how's   | 5           | 0             | 10          |
  20 | what's, who's, aint's, that's        | 4           | 0             | 8           |
  21
```

```
tests\step_defs\test_eprime.py ..F..FFFFF
```

```
===== short test summary info =====
FAILED tests\step_defs\test_eprime.py::test_eprime_samples_correctly_analyzed[to be or not to be - Hamlet's dilemma-2-1-8] - AssertionError: assert '9' == '8'
FAILED tests\step_defs\test_eprime.py::test_eprime_samples_correctly_analyzed[be, being, been, am, is, isn't-6-0-7] - AssertionError: assert '6' == '7'
FAILED tests\step_defs\test_eprime.py::test_eprime_samples_correctly_analyzed[are, aren't, was, wasn't, were, weren't-6-0-9] - AssertionError: assert '6' == '9'
FAILED tests\step_defs\test_eprime.py::test_eprime_samples_correctly_analyzed[I'm, you're, we're, they're, he's, she's-6-0-12] - AssertionError: assert '6' == '12'
FAILED tests\step_defs\test_eprime.py::test_eprime_samples_correctly_analyzed[it's, there's, here's, where's, how's-5-0-10] - AssertionError: assert '5' == '10'
FAILED tests\step_defs\test_eprime.py::test_eprime_samples_correctly_analyzed[what's, who's, aint's, that's-4-0-8] - AssertionError: assert '4' == '8'
===== 6 failed, 4 passed in 9.43s =====
```

Documenting as Executable Test Automation

- Single line
 - See it fail
 - First test
 - Same test but variables
 - Same test but templates
 - **Failing test with a bug**
 - Spec to tests
 - Guess the values that are likely to fail
 - Multiple browsers

Throwaway automation?



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Why This is not about Any Specific Tool

Chapter 13



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Documentation as a Constraint

A Balancing Act between Now and Future
Never be bored is not possible without
automation



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Automation in Frame of Exploratory Testing

-  Documenting
-  Extending reach
-  Alerting to attend
-  Guiding to detail

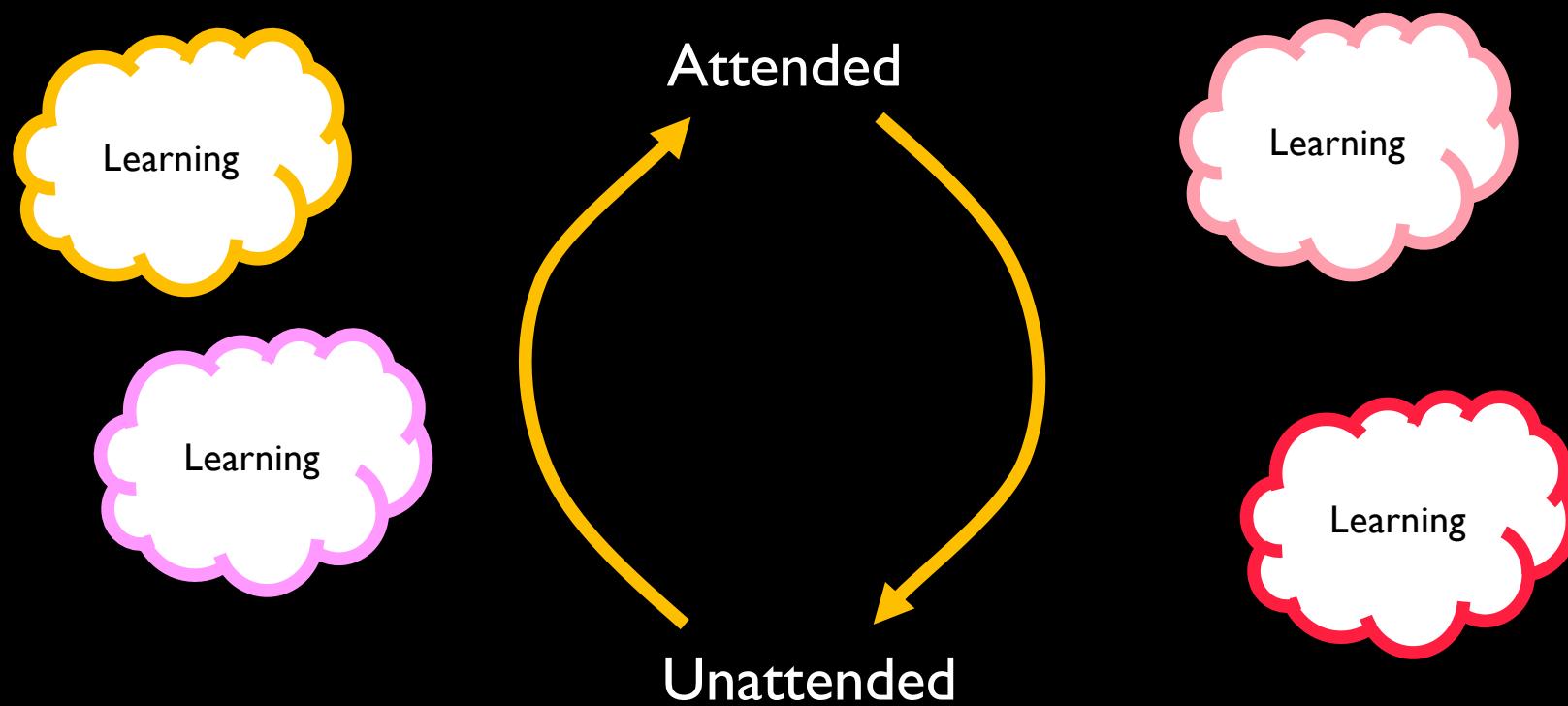


<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Moving Focus



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Stop-and-Think: Test Automation as Documentation

How would the testing you did before
this have been different if you were to
start with this?



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Use of Time

Chapter 14



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Test, Bug, Setup

Software with little bugs is faster to test
Setup is configuring, learning and
documenting
Test grows coverage



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

The screenshot shows the E-Primer tool interface. At the top, there is a license notice for Apache License 2.0, followed by sections for Permissions, Limitations, and Conditions. Below this, a note credits Alan Richardson and eviltester. The main title is **E-Primer an e-prime checking tool**. The interface includes the following text and lists:

Do you want to write without using the verb "to be"?

Do you want to master e-prime?

Use our online tool to check your writing.

- Word Count: 9
- Discouraged Words: 2
- Possible Violations: 1

to be or not to be - hamlet's dilemma

Text:
to be or not to be - hamlet's dilemma

Check For E-Prime

Data trap (indicated by a blue arrow pointing to the text input field)

Test Cases trap (indicated by a blue arrow pointing to the title)

Bug trap (indicated by a blue arrow pointing to the word count statistics)

Algorithm trap (indicated by a blue arrow pointing to the analysis results)



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Stop-and-Think: Time and Traps

Where did your time go on testing of
the application?



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Coverage

Chapter 15



<https://www.linkedin.com/in/maaret/>

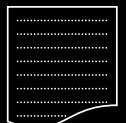


@maaretp@mas.to

Setting the Stage for Testing

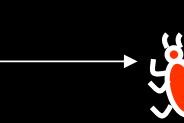
WHAT
WHEN
WHO
HOW
WHY

We target
these...



Test ideas

...to find
these...



Serious

B business

PM project (time)

T testing (time)

U user

...to tell if there's
more and what
level we know
things.



Coverage

REQUIREMENTS
RISKS (of relevant bugs)
CODE
ENVIRONMENTS



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Risk Coverage

Coverage of relevant bugs
Effectiveness – results of overall strategy
facilitate experience of quality for
stakeholders



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Stop-and-Think: Coverage of Today's Testing

Would the testing you thought of have
missed any of the bugs we have seen?

What did we not test?



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Test Strategy

Chapter 16



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Ideas that Guide Test Design

Specific to Application Under Test
Risks to ways of testing for them



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to



Let's Test

<https://www.exploratorytestingacademy.com/app/>

<https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html>



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Test Strategy for E-Primer

What is the product?

- E-Primer is an English text validator that checks text against specific rules around avoiding the verb 'to be'. It identifies rule breaking in two categories: one that can be checked by a rule, and another that needs human assessment (for now).

What are the key potential risks?

- It suggests the wrong corrections and misses corrections in realistic text samples
- It miscounts words in a way that leads us to underappreciate the scale of processing.
- It looks wrong on some browsers and data samples
- It requires too much effort to learn in relation to the value of proofreading it provides

How could we test the product so as to evaluate the *actual* risks associated with it?

- Understand the rules of e-prime through research
- Collect data samples (short and long ones) that represent both e-prime text and text that violates rules of e-prime and run them through the program.
- Verify common forms of 'to be' are systematically recognized across the samples
- Document specification as automation that shows the rules of e-prime and enables running subset of all tests across browsers.
- Try fooling word count to count less words or more words by specific data samples
- Run the web page through a set of html-validation tools
- Visually verify the page with realistic e-prime text samples
- Read the code of the application for inspiration focusing on names of functions rather than understanding implementation
- Summarize learning obstacles for user and value of the application as comparison sheet



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Full Results and Reproducing from Customer Feedback

Chapter 17



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Invisible ink

Customers will tell you of some of the things you missed
Observing customers (incl. telemetry) will tell you of
some of the things you missed
Limited reporting capability



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

BUGS:

- Css validator gives errors
 - Special character as start of line forces an extra line change in grey display box
 - Firefox text field does not clear with ctrl+R
 - html validator identifies 3 errors
 - You're / we're / They're contractions not recognised as violations of e-prime
 - Two words separated by line feed are counted as one
 - Human being is noun but recognised as violation
 - Space is considered only separator for words and special characters are counted as words
 - Long text moves button outside user's access as vertical scroll is disabled
 - id naming is inconsistent, some are camel case, others not
 - Long texts without spaces go outside the grey area reserved for displaying the texts
 - Red/blue on grey has bad contrast
 - Zoom or resize of browser renders page unusable due to missing scroll bars
 - Contractions for word count (I'm) count as two words as per general searchable rules of how word counting works
 - The possible violation's category takes possessives and leaves for human assessment and would probably be expected to be something to create programmatic rules on
 - Possible violations does not handle typesetter's apostrophe, only typewriter's apostrophe in calculation
 - Two part words (like people's last names) in possessive form are not recognised as possible violations
 - Images missing alt text necessary for accessibility
 - Accessibility warnings on contrast
 - Mobile use not supported, styles very non-responsive
 - UI instructions for user are unclear
 - if word is in single quotes, it is not properly recognised as e-prime.
 - text box location in UI is not where user would expect it to be as per the logic of how web pages are usually operating
 - Site is missing favicon and security.txt - both common conventions for web applications
 - Resizing the input text field can move it outside view so that it cannot be resized back
 - Choosing which links are to overload this app and which open new browser window are inconsistent
- | word + space + enter creates extra vertical space in grey area
'is are' both are not recognised
zoom with and horizontal bar has issues, seen on edge and chrome, on win and Mac with touchpad
Wave accessibility tool does a forced refresh on close [TOOL?](#)
On Chrome when settings panel scroll does not work. It works on Firefox. [TOOL?](#)
Enter in end of line moves last word to different line on grey box
Firefox does not come back from mobile use simulation without forced refresh with this site [TOOL?](#)



34 bugs



<https://www.linkedin.com/in/maarep/>



@maaretp@mas.to



Let's Test

<https://www.exploratorytestingacademy.com/app/>

<https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html>



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Finding (more of some) relevant Conversation Starters

You will never find all bugs.

Target finding only things that matter. Learn what matters.

Seeing and not reporting is better than not seeing problems.



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Closing Remarks

Chapter 18



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Course Outline

Chapter 1: Test target and our options for exploring
Chapter 2: Self-management basics on setting yourself constraints
Chapter 3: The moment of first impression
Chapter 4: Recognizing and learning a domain
Chapter 5: Recognizing functionality
Chapter 6: Recognizing data
Chapter 7: Recognizing application and execution environment
Chapter 8: Documenting in a mindmap

Chapter 9: pytest the very basics
Chapter 10: Documenting as skeleton test automation
Chapter 11: Playwright library and CSS selectors on web pages
Chapter 12: Documenting as executable test automation
Chapter 13: Why this is not about any specific tool
Chapter 14: Use of time
Chapter 15: Coverage
Chapter 16: Test strategy
Chapter 17: Full results and reproducing from customer feedback
Chapter 17: Closing remarks



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

Maaret Pyhäjärvi (from Finland)



Email: maaret@iki.fi

Mastodon: [@maaretp@mas.to](https://mastodon.social/@maaretp)

Web: maaretp.com

Blog: visible-quality.blogspot.fi
*(please connect with me through
Mastodon or LinkedIn)*



#PayToSpeak #TechVoices
#EnsembleTesting #EnsembleProgramming #StrongStylePairing
#ExploratoryTesting #TestAutomation
#ModernAgile
#ContemporaryExploratoryTesting



<https://www.linkedin.com/in/maaret/>

@maaretp@mas.to

