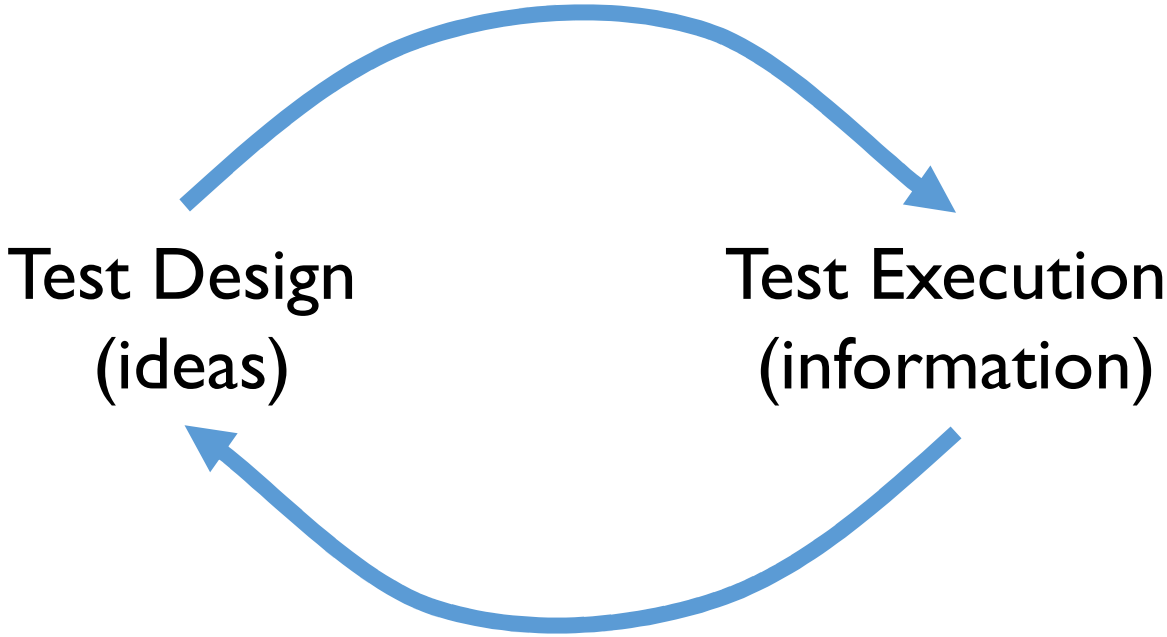
Exploratory Testing Foundations

Maaret Pyhäjärvi

@maaretp

Optimizing the value of testing

Learning



Exploratory Testing
the Verb

Input

Tester

Domain knowledge

Requirements and specifications

Testing knowledge

Miscellaneous knowledge

Output

Better tester

Coverage

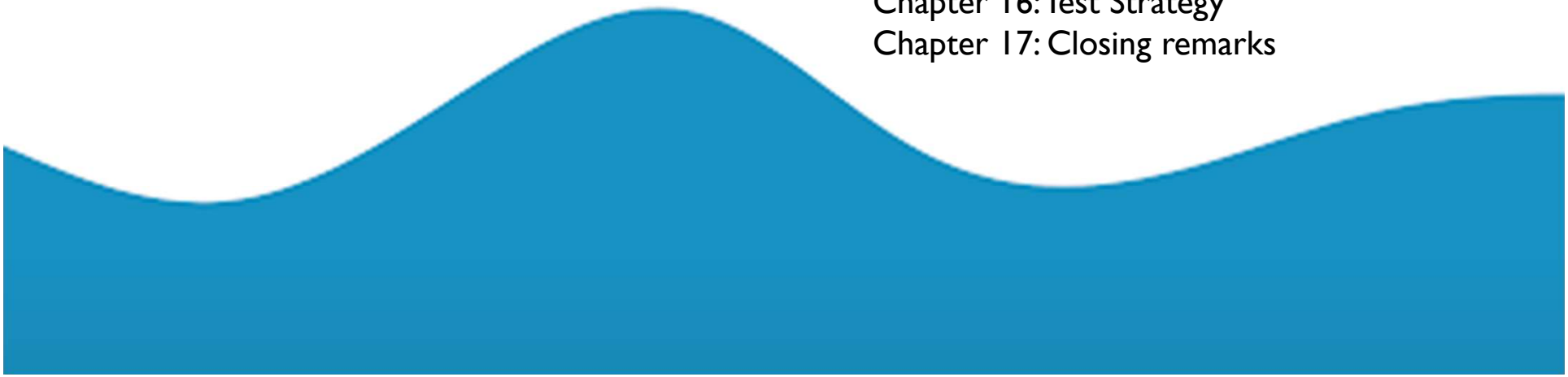
Information incl. defects and change requests

Documentation: Strategy

Documentation: Tests

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: Robot framework the very basics
Chapter 10: Documenting as skeleton test automation
Chapter 11: Robot framework browser library and CSS selectors on web pages
Chapter 12: Documenting as executable test automation
Chapter 13: Why this is not about Robot Framework
Chapter 14: Use of time
Chapter 15: Coverage
Chapter 16: Test Strategy
Chapter 17: Closing remarks



Course Outline

Section I: Options for Exploring 1 3

Section II: Control through Choices 2

Section III: Documenting (with Automation)
Extending with Function, Data, Environment and Domain 4-13

Section IV: Use of time and coverage 14-17

Test Target and Our Options for Exploring

Chapter 1

exploratorytestingacademy.com/app/

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

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?

Options for Exploring

Research the Domain

Use it *with a constraint*

Self-management Basics on Setting Yourself Constraints

Chapter 2

Charters

Charter template

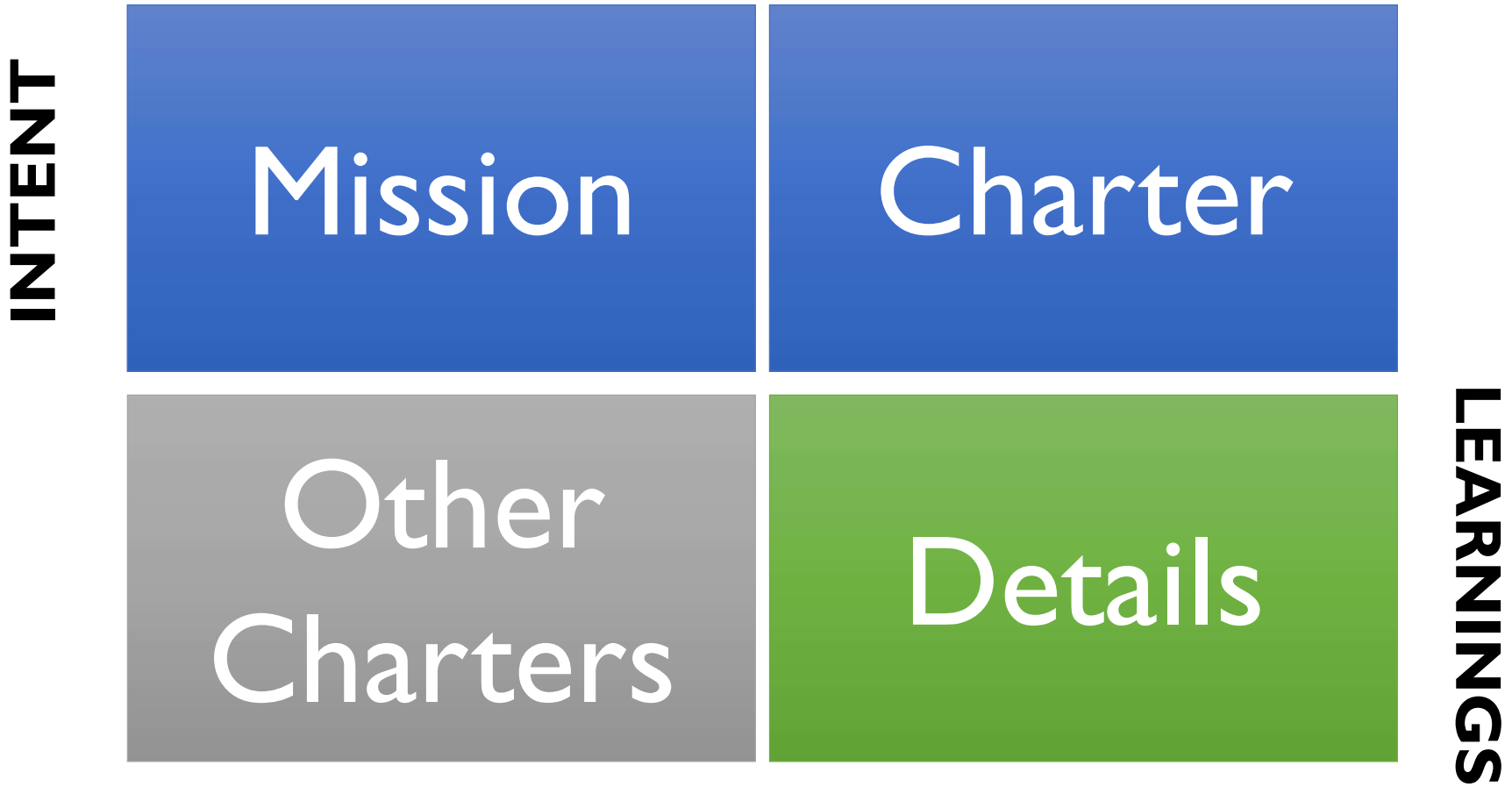
- *target*: where you're exploring
- *resources*: what you're using/how you're exploring
- *information*: what question you want to answer

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

Choose Your Own Constraint

Deliberately excluding perspectives!
Never Be Bored!

Explore with Intent



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?

The Moment of First Impression

Chapter 3

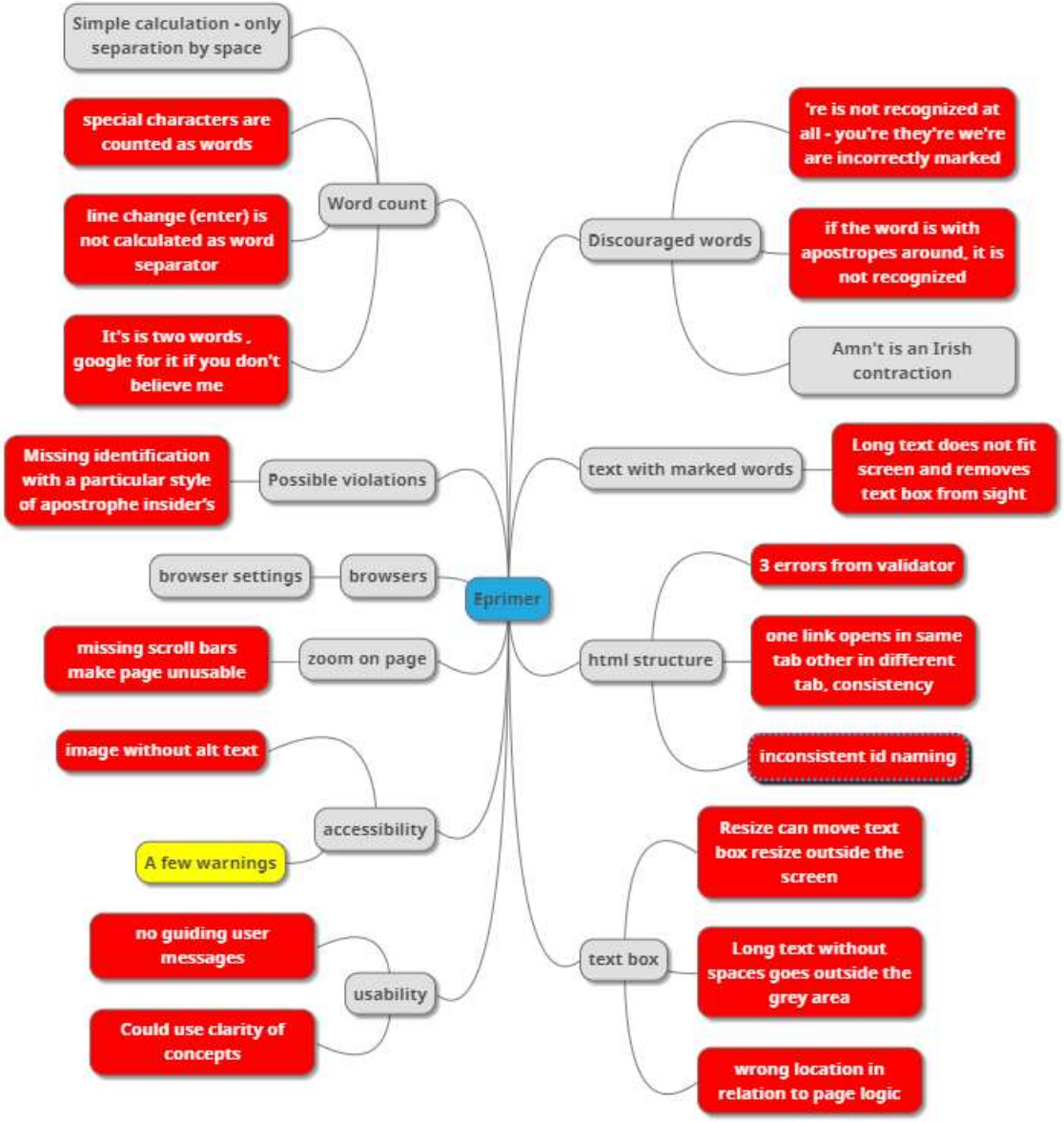
Options Expire

Capture First Impression
Borrow someone else's First Impression
Timing of feedback changes reaction to it!

Let's Test

https://www.exploratorytestingacademy.com/app/
https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html

Example: Test Results, Red is Bug



Bugs are Conversation Starters

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

Recognizing and Learning a Domain

Chapter 4

Conference
Reference
Inference

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

exploratorytestingacademy.com/app/eprime.js

```
function inEPrimeOutputFormat(aWord){
    return '<span class="ep_violation">' + aWord + "</span>";
}

function inPossibleEPrimeOutputFormat(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());

}
```

Let's Test

https://www.exploratorytestingacademy.com/app/
https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html

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!

Recognizing Functionality

Chapter 5

Naming of Function

Functions in Code
Expected Features
Visible Features

Let's Test

https://www.exploratorytestingacademy.com/app/
https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html

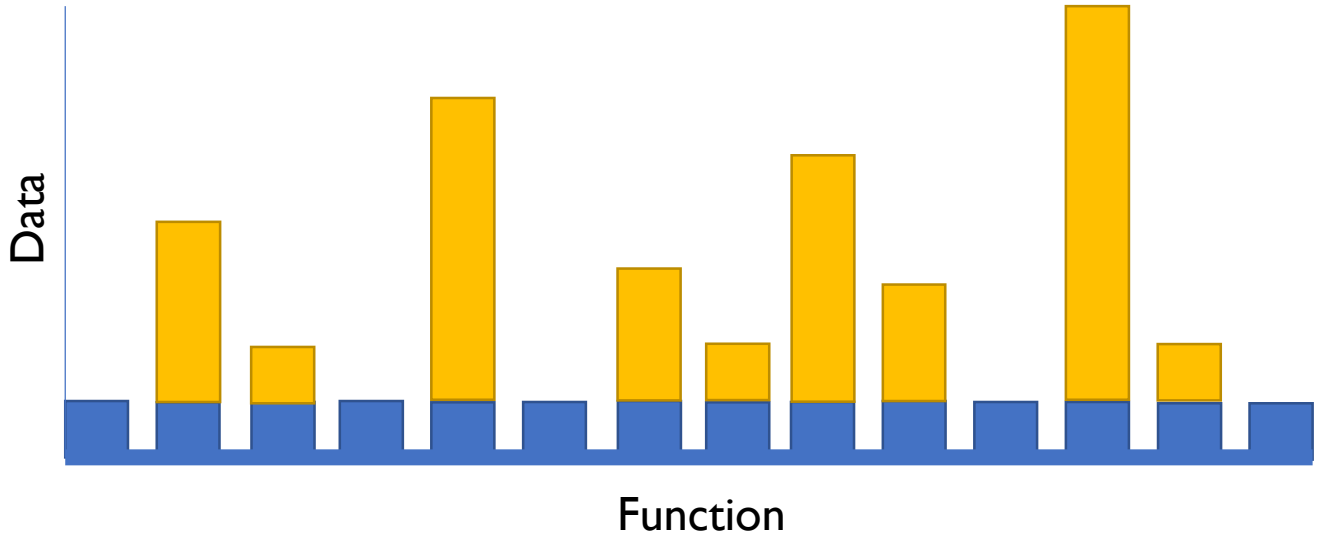
Learning of Function of E-Primer

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

Recognizing Data

Chapter 6

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

Let's Test

https://www.exploratorytestingacademy.com/app/
https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html

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

Recognizing Application and Execution Environment

Chapter 7

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

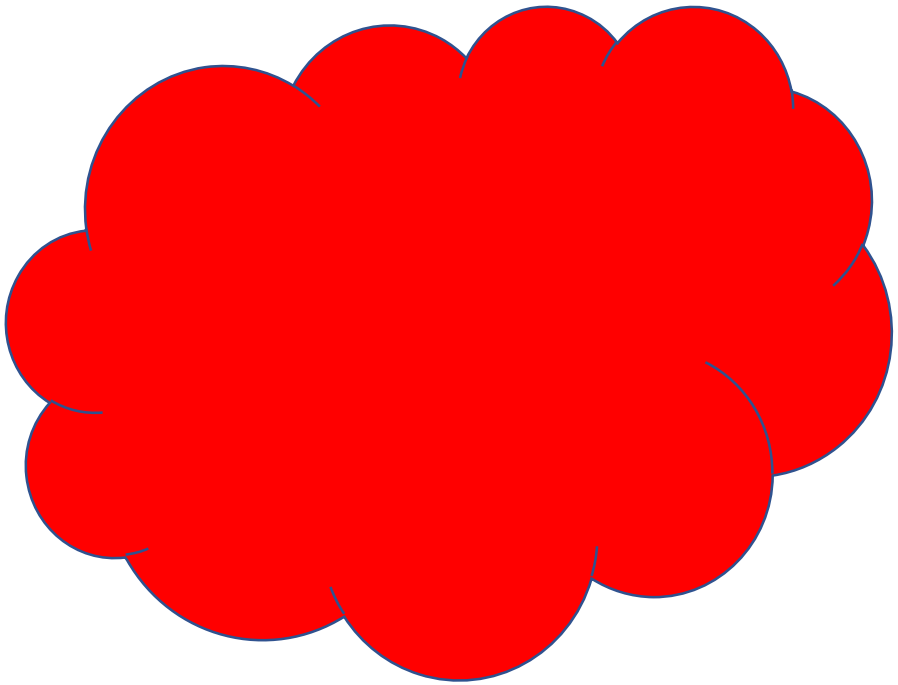
Execution Environment

Different browsers: web and mobile
Browser functionality and add-ons
HTML standard compatibility
Accessibility standard compatibility

Let's Test

https://www.exploratorytestingacademy.com/app/
https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html

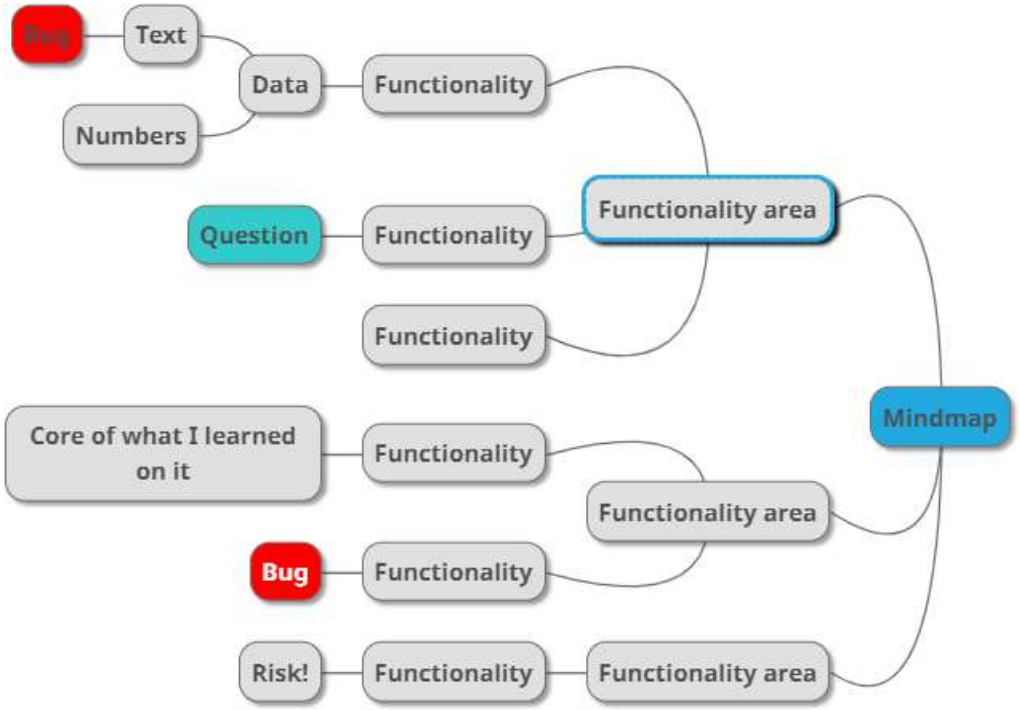
Learning of Application and Execution Environment of E-Primer



Documenting in a Mindmap

Chapter 8

Mindmap


Bug
Text
Data
Numbers
Functionality
Question
Functionality
Functionality
Functionality area
Core of what I learned on it
Functionality
Bug
Functionality
Risk!
Functionality
Functionality area
Functionality area
Mindmap

Bug Reports

Cem Kaner. Bug Reporting Heuristic.

- Replicate
- Isolate
- Maximize
- Generalize
- External
- Neutral tone

Let's Test

https://www.exploratorytestingacademy.com/app/
https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html

Mindmapping as Future Reference

Notetaking in the moment

Restructure as you learn

Documentation for the future

General purpose mindmaps

Robot Framework the Very Basics

Chapter 9

Robot Framework

Custom-made language
Built-in reporting
Ecosystem of keyword libraries

Documenting as Skeleton Test Automation

Chapter 10

Log

```
1  *** Test Cases ***
2  This is a test case name
3      Log     First thing to do
4      Log     Second thing to do
5      Log     Third thing to do
```

```
==============================================================================
Basic
==============================================================================
This is a test case name                                              | PASS |
------------------------------------------------------------------------------
Basic                                                                 | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
==============================================================================
```

Let's Test

https://www.exploratorytestingacademy.com/app/
https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html

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

Robot Framework Browser Library and css selectors on Web Page

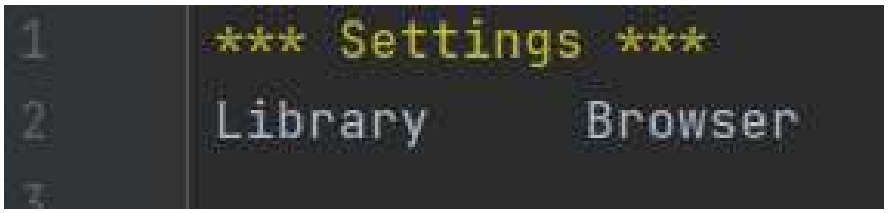
Chapter 11

Browser Library

Playwright inside

Speed – Reliability – Visibility

Automatic waits



css selectors

css=
#id
.class
tag
[attribute='value']
[part_of_attribute_value_contains*='value']

Keywords

```
1  *** Settings ***
2  Library     Browser
3
4  *** Test Cases ***
5  Open the Page Headless
6      New Page    https://www.exploratorytestingacademy.com/app/
```

https://marketsquare.github.io/robotframework-browser/Browser.html

Documenting as Executable Test Automation

Chapter 12

Let's Test

https://www.exploratorytestingacademy.com/app/
https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html

firstTest.robot

```
*** Settings ***
Library             Browser
Test Setup          Default Setup
Test Teardown       Default Teardown

*** Variables ***
${URL}                  https://www.exploratorytestingacademy.com/app/
${input text}           To be or not to be is Hamlet's dilemma
${word count}           9
${discouraged count}    3

*** Test Cases ***

Verify Word Text
    New Page   ${URL}
    Fill Text       css=#inputtext  ${input text}
    Click   css=#CheckForEPrimeButton
    Get Text    css=#eprimeoutput   ==   ${input text}
    Get Text    css=#wordCount      ==   ${word count}
    Get Text    css=#discouragedWordCount  ==   ${discouraged count}

*** Keywords ***
Default Setup
    New Browser             chromium    headless=${FALSE}

Default Teardown
    Close Browser
```

REPORT

firstTest Log

Generated
20210202 21:16:15 UTC+02:00
22 seconds ago

Test Statistics

Total Statistics	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	1	1	0	00:00:03	
All Tests	1	1	0	00:00:03	

Statistics by Tag	Total	Pass	Fail	Elapsed	Pass / Fail
No Tags					

Statistics by Suite	Total	Pass	Fail	Elapsed	Pass / Fail
firstTest	1	1	0	00:00:04	

Test Execution Log

SUITE **firstTest** 00:00:03.523

Full Name: firstTest
Source: C:\BitbucketRepos\localBrowserCoiote\eprime\firstTest.robot
Start / End / Elapsed: 20210202 21:16:11.775 / 20210202 21:16:15.298 / 00:00:03.523
Status: 1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed

TEST **Verify Word Text** 00:00:02.642

Full Name: firstTest.Verify Word Text
Start / End / Elapsed: 20210202 21:16:12.650 / 20210202 21:16:15.292 / 00:00:02.642
Status: PASS (critical)

SETUP	**Default Setup**	00:00:00.613
KEYWORD	Browser . **New Page** ${URL}	00:00:01.703
KEYWORD	Browser . **Fill Text** css=#inputtext, ${input text}	00:00:00.036
KEYWORD	Browser . **Click** css=#CheckForEPrimeButton	00:00:00.049
KEYWORD	Browser . **Get Text** css=#eprimeoutput, ==, ${input text}	00:00:00.026
KEYWORD	Browser . **Get Text** css=#wordCount, ==, ${word count}	00:00:00.015
KEYWORD	Browser . **Get Text** css=#discouragedWordCount, ==, ${discouraged count}	00:00:00.014
TEARDOWN	**Default Teardown**	00:00:00.160

test.robot

```
*** Settings ***
Library     Browser
Test Setup          Default Setup
Test Teardown       Default Teardown
Test Template       Verify Word Text

*** Variables ***
${URL}      https://www.exploratorytestingacademy.com/app/

*** Test Cases ***
Test1    nothing    1    0
Test2    to be or not to be     6    2
Test3    The cat is my only pet    6    1
Test4    The cat is Garfield    4    1
Test5    be, being, been, am, is, isn't, are, aren't, was, wasn't, were, and weren't.    13    12
Test6    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.    15    15
Test7    ${EMPTY}    0    0

*** Keywords ***
Verify Word Text
    [Arguments]     ${input text}      ${word count}      ${discouraged count}
    New Page    ${URL}
    Fill Text       css=#inputtext  ${input text}
    Click   css=#CheckForEPrimeButton
    Get Text    css=#eprimeoutput   ==   ${input text}
    Get Text    css=#wordCount      ==   ${word count}
    Get Text    css=#discouragedWordCount  ==   ${discouraged count}

Default Setup
    New Browser            chromium    headless=${FALSE}

Default Teardown
    Close Browser
```

REPORT

Source: C:\BitbucketRepos\localBrowserCoiote\eprime\test.robot
Start / End / Elapsed: 20210123 19:00:35.362 / 20210123 19:01:09.799 / 00:00:34.437
Status: 7 critical test, 6 passed, 1 failed
7 test total, 6 passed, 1 failed

+ TEST **Test1**	00:00:09.916
+ TEST **Test2**	00:00:03.287
+ TEST **Test3**	00:00:03.089
+ TEST **Test4**	00:00:03.193
+ TEST **Test5**	00:00:03.312
− TEST **Test6**	00:00:04.722

Full Name: Test.Test6
Start / End / Elapsed: 20210123 19:01:01.441 / 20210123 19:01:06.163 / 00:00:04.722
Status: FAIL (critical)
Message: Property innerText '1' (str) should be '15' (str)

+ SETUP **Default Setup**	00:00:00.961
− KEYWORD **Verify Word Text** 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., 15, 15	00:00:03.276

Start / End / Elapsed: 20210123 19:01:02.417 / 20210123 19:01:05.693 / 00:00:03.276

+ KEYWORD Browser. **New Page** ${URL}	00:00:01.622
+ KEYWORD Browser. **Fill Text** css=#inputtext, ${input text}	00:00:00.054
+ KEYWORD Browser. **Click** css=#CheckForEPrimeButton	00:00:00.053
+ KEYWORD Browser. **Get Text** css=#eprimeoutput, ==, ${input text}	00:00:00.024
+ KEYWORD Browser. **Get Text** css=#wordCount, ==, ${word count}	00:00:00.023
− KEYWORD Browser. **Get Text** css=#discouragedWordCount, ==, ${discouraged count}	00:00:01.495

Documentation: Returns text attribute of the element found by `selector`. See the `Finding elements` section for details about the selectors.
Tags: Assertion, Getter, PageContent
Start / End / Elapsed: 20210123 19:01:04.198 / 20210123 19:01:05.693 / 00:00:01.495

19:01:05.575 INFO

19:01:05.693 FAIL Property innerText '1' (str) should be '15' (str)

+ TEARDOWN **Default Teardown**	00:00:00.468
+ TEST **Test7**	00:00:03.620

Documenting as Executable Test Automation

Throwaway 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
→Runs in CI

Why This is not about Robot Framework

Chapter 13

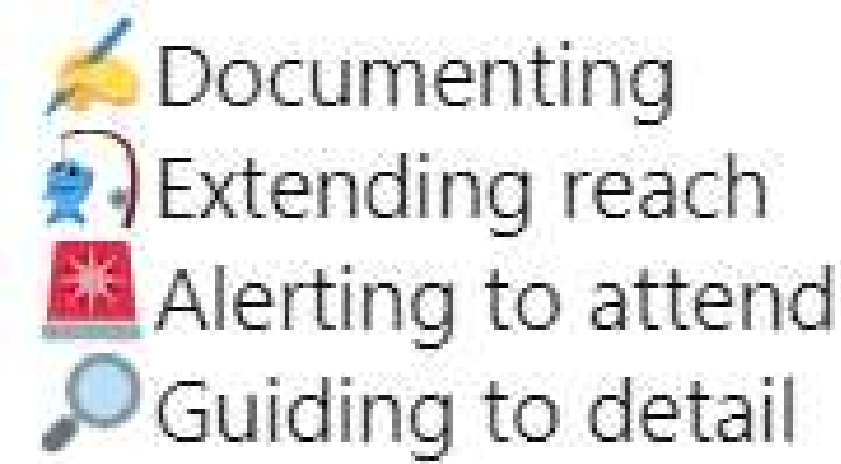
Documentation as a Constraint

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

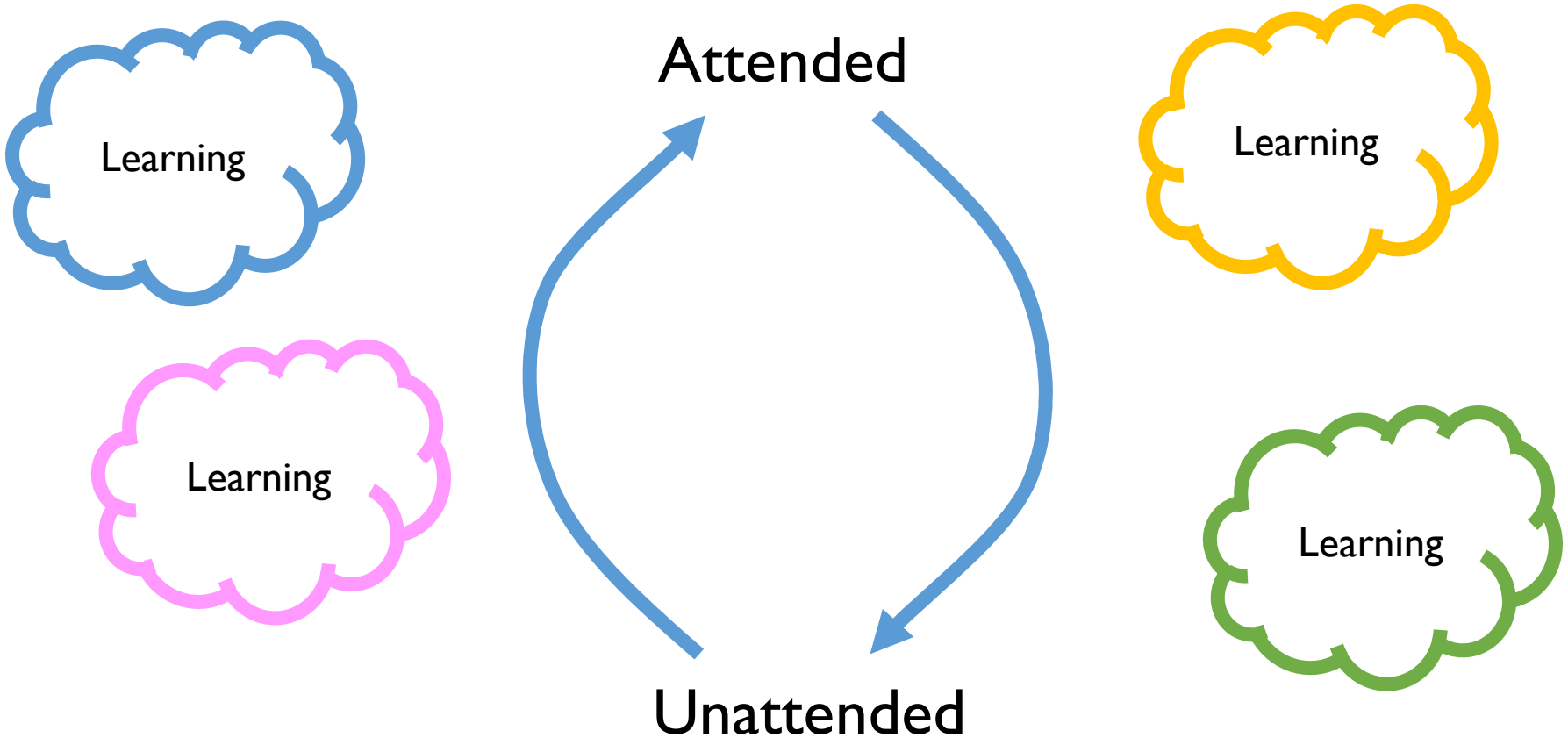
@maaretp

http://maaretp.com

Automation in Frame of Exploratory Testing



Moving Focus



Stop-and-Think: Robot Framework Browser

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

Use of Time

Chapter 14

Test, Bug, Setup

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

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: 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

Test Cases trap

Bug trap

Algorithm trap

Stop-and-Think: Time and Traps

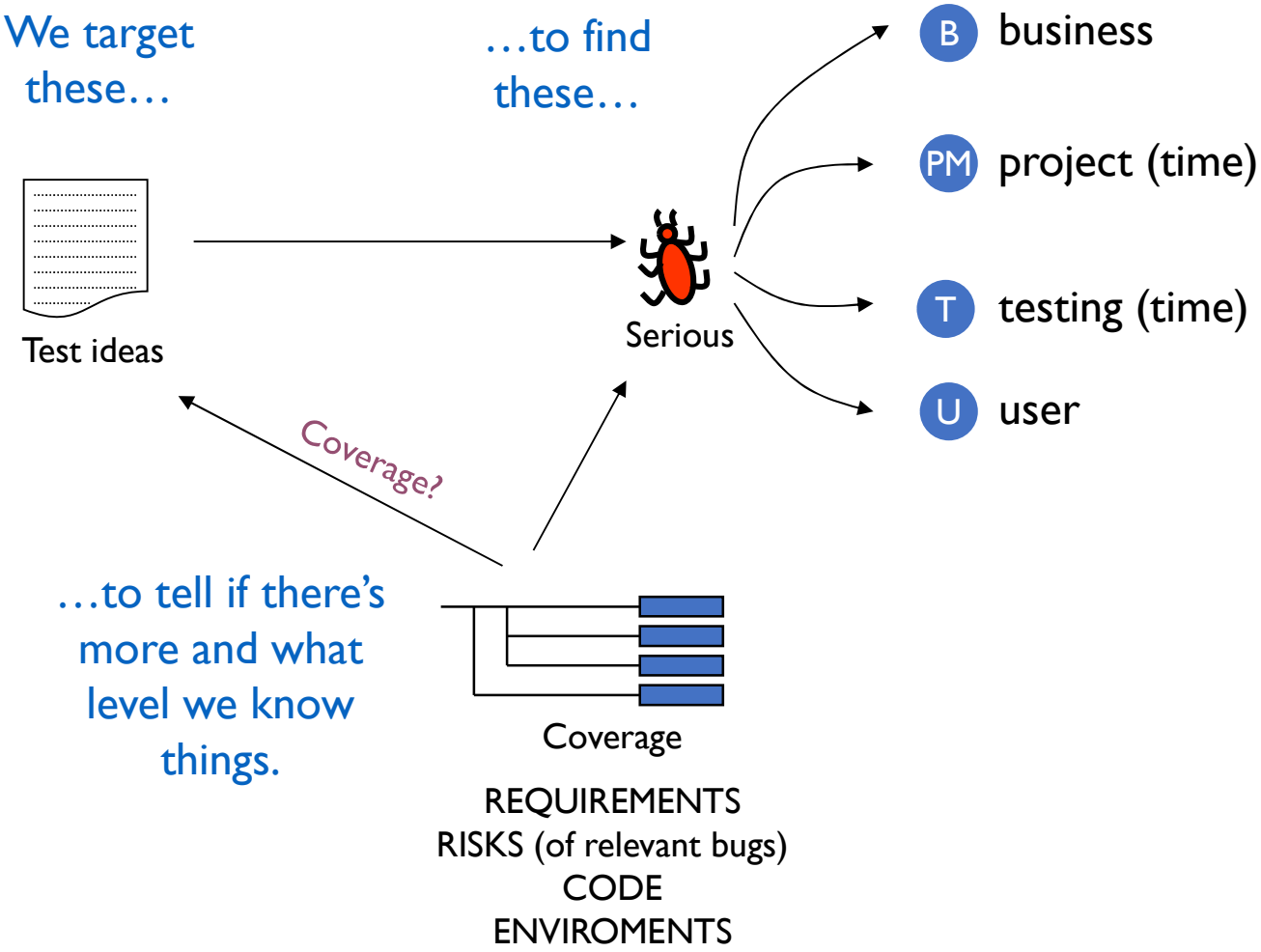
Where did your time go on testing of the application?

Coverage

Chapter 15

Setting the Stage for Testing

WHAT
WHEN
WHO
HOW
WHY



Risk Coverage

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

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?

Test Strategy

Chapter 16

Ideas that Guide Test Design

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

Let's Test

https://www.exploratorytestingacademy.com/app/
https://eviltester.github.io/TestingApp/apps/eprimer/eprimer.html

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 suggest 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?

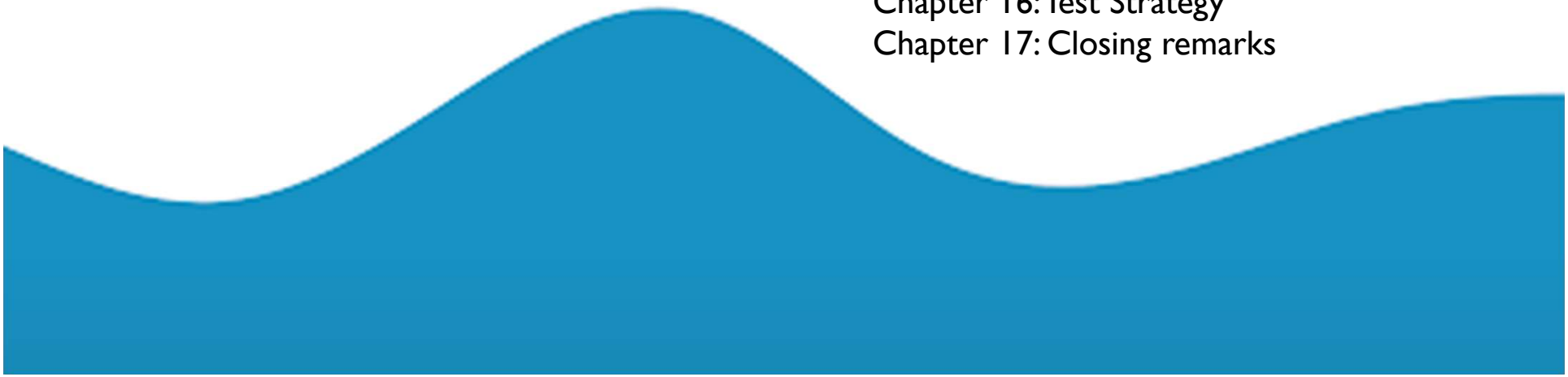
- Understand the rules of e-prime through research
- Collect data samples (short and long ones) that represent both e-prime text and text that violates rules of e-prime and run them through the program.
- Verify common forms of 'to be' are systematically recognized across the samples
- Document specification 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-validators
- 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

Closing Remarks

Chapter 17

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: Robot framework the very basics
Chapter 10: Documenting as skeleton test automation
Chapter 11: Robot framework browser library and CSS selectors on web pages
Chapter 12: Documenting as executable test automation
Chapter 13: Why this is not about Robot Framework
Chapter 14: Use of time
Chapter 15: Coverage
Chapter 16: Test Strategy
Chapter 17: Closing remarks



Maaret Pyhäjärvi

EuroSTAR
TESTING EXCELLENCE
AWARD

2020

AGILE
TESTING DAYS

MIATPP
Most Influential Agile Testing
Professional Person

2016

Email: maaret@iki.fi
Twitter: @maaretp
Web: maaretp.com
Blog: visible-quality.blogspot.fi
(please connect with me through Twitter or LinkedIn)

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