

# Testing with a Focus on User Experience

**David Duke**  
**Engineering Manager - Test**



# Learning objectives

By the end of this session:

You will understand how to interact with testers to optimise the user interface.

You will have experience of constructing, interpreting and executing test plans with a focus on accessibility and usability.



# Agenda

1. Why do we test?
2. How can we test?
3. How can we structure our findings?



**Why do we test?**



# The Two Big Questions

To discover information in order to answer two questions.

- Are we building the right thing?
- Are we building the thing right?

# Before we can think about why we test, what kinds of testing do we do?

Lots of types of testing:

- User Interface testing : manual and automation

- Unit testing

- Accessibility testing

- API testing

Split you into groups of four or five.

Come up with other types of testing you think people might do. Use post-its

One way of coming up with some of these is to think about the different aspects of the software or attributes of the software.

# Before we can think about why we test, what kinds of testing do we do?

Lots of types of testing:

- User Interface testing : manual and automation

- Unit testing

- Contract testing

- Accessibility testing

- API testing

Why do we test any of these things?



# What do I mean?

- Are we building the right thing?
- Are we building the thing right?

What do I mean by this?

What does this system do?

Is it the right thing to be building?

Does it do what it's supposed to do?



# Example

I want to travel across a town on my own faster than I can walk.

Two questions:

- Are we building the right thing?
- Are we building the thing right?

1: Wrong thing

2: Right thing but won't work properly

3: Right thing and built the right way



# Example

I want to travel across a town on my own faster than I can walk.

Two questions:

- Are we building the right thing?
- Are we building the thing right?

1: Wrong thing

2: Right thing but won't work properly

3: Right thing and built the right way



# Example

I want to travel across a town on my own faster than I can walk.

Two questions:

- Are we building the right thing?
- Are we building the thing right?

1: Wrong thing

2: Right thing but won't work properly

3: Right thing and built the right way



# Example

I want to travel across a town on my own faster than I can walk.

Two questions:

- Are we building the right thing?
- Are we building the thing right?

- 1: Wrong thing
- 2: Right thing but won't work properly
- 3: Right thing and built the right way



# Example

I want to travel across a town on my own faster than I can walk.

Two questions:

- Are we building the right thing?
- Are we building the thing right?

1: Wrong thing

2: Right thing but won't work properly

3: Right thing and built the right way



# Example

I want to travel across a town on my own faster than I can walk.

Two questions:

- Are we building the right thing?
- Are we building the thing right?

- 1: Wrong thing
- 2: Right thing but won't work properly
- 3: Right thing and built the right way



# Example

I want to travel across a town on my own faster than I can walk.

Two questions:

- Are we building the right thing?
- Are we building the thing right?

- 1: Wrong thing
- 2: Right thing but won't work properly
- 3: Right thing and built the right way



# Example

I want to travel across a town on my own faster than I can walk.

Two questions:

- Are we building the right thing?
- Are we building the thing right?

1: Wrong thing

2: Right thing but won't work properly

3: Right thing and built the right way





# Can we always answer these questions easily?

Are we building the right thing?

Are we building the thing right?

Basically you're trying to answer the question, does it work?

Functional requirements can be easy.

- Button should be green\*
- Data should be saved/ updated/ deleted
- If you enter X, do Y

Non Functional Requirements can be much harder.

\*(#006400 dark green)

# Non functional requirements (NFR)

What's a Non Functional Requirement?

Usability and Accessibility are Non Functional requirements?

Do you know any other Non Functional requirements?

# Why are NFRs difficult for testers

These can be difficult to say whether things work according to these requirements because some of them are subjective.

## Functional Requirement:

- I this button to send a specific message when you click on it.

## Non Functional requirement:

- I want my page to be performant:  
What does performant even mean, how long is okay for it to take to send a message, how many people are you expecting, do you even know?

# Break

See you in 10 mins



# How can we Test?

- Oracles
- Test cases/ Exploratory charters

# Oracles

ISTQB (International Software Testing Qualification Board):

*“Test Oracle: A source to determine expected results to compare with the actual result of the system under test.”*

Wait? Why do I care about Oracles?

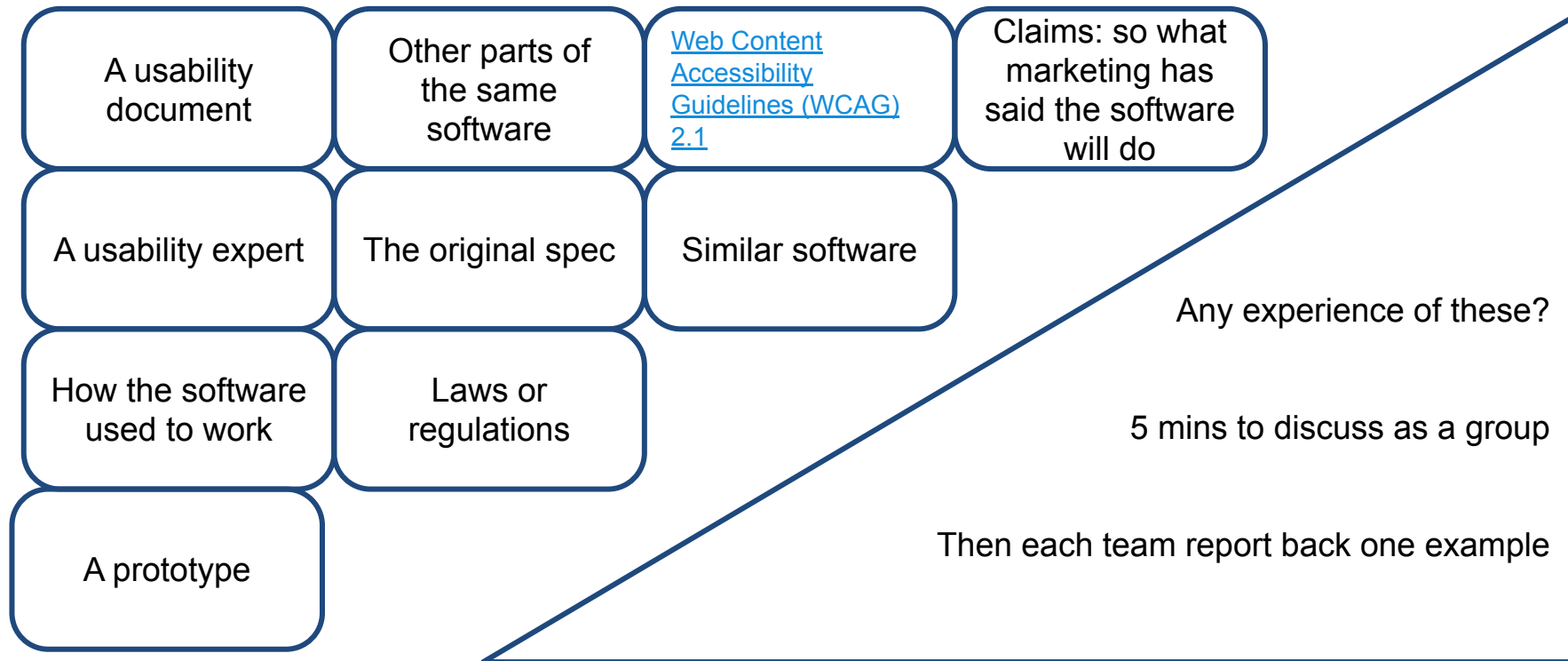
You're using them, so you should understand them.

They are the many ways you decide if the software is behaving correctly.

# Why do I need Oracles if I have a spec?

- No specification
- Specification isn't all encompassing
- Conflicts with common sense
- Self Conflicting
- Conflicts with regulations
- Confusing or vague specification

## Oracles Exercise







# **“Manual Testing”**

# How should you structure manual testing?



Test Cases

vs



Exploratory Charters

# What are test Cases?

ISTQB: A set of preconditions, inputs, actions, expected results and postconditions, developed based on test conditions.

These are things like

1. Go in to the <https://www.theguardian.com/uk> home page and click on the Sport heading,  
Expected: it should take you to the sport url:  
<https://www.theguardian.com/uk/sport>
2. Go in to the <https://www.theguardian.com/uk> home page and click on the Culture heading,  
Expected: it should take you to the culture url:  
<https://www.theguardian.com/uk/culture>
3. etc...

You have 10 minutes to test using this set of test cases.

<http://bit.ly/testcase1604>

Websites to use

<https://www.lonelyplanet.com/>

<https://www.viator.com/>

<https://www.musement.com>

<https://www.roughguides.com>

Keep a note on post-its or however you fancy

1. Problems with the test cases - wrong / confusing / unclear?
2. Things you could have tested that were outside the case?
3. Any maintenance problems with the tests that you think of?
4. Problems with running the same tests every time?

# Test Cases - discuss

The background of the slide features a blue-tinted photograph of a long, straight road covered in snow, receding into the distance towards a range of mountains under a clear sky.

1. Problems with the test cases - wrong / confusing / unclear?
2. Things you could have tested that were outside the case?
3. Any maintenance problems you think of?
4. Problems with running the same tests every time?

# Pesticide Paradox



If you always run the same test cases again and again, over a period of time these test cases aren't going to find that many new defect in the system because they're always looking at the same things and any parts of the system that aren't covered by these test cases will always be where the bugs start to fester, untamed.

# Break

See you in 10 mins

# What is Exploratory Testing?

It's not ad-hoc testing



Exploratory

vs

Ad-hoc



# What is Exploratory testing?

Exploratory Testing is a type of testing where you design and execute the tests simultaneously.

One way of doing this is using a Charter.

Explore (a place or thing)

With (some tools)

To discover (how something works)



# Sections of Charter

Goal

Oracles

Scope

Discoveries

Questions / Follow-up / Bugs /Successes

# Exploratory Charter example

You have 15 mins to run the following Exploratory charter.

<http://bit.ly/goexplore1604>

Explore getting information about different obscure holiday destinations or activities

With manual testing and any plugins you want

To discover how accessible or usable the functionality is when no results are returned

Websites to use:

<https://www.lonelyplanet.com/>

<https://www.viator.com/>

<https://www.musement.com>

<https://www.roughguides.com>



# Discussion

Have a 5 minute session in groups of 4 or 5. Talk about the following things.

What did you prefer about test cases?

What did you prefer about exploratory charters?

What's the most interesting issue you found?

What was the most challenging thing about the testing?

Can you see reasons for why you would do exploratory over test cases and the other way round?

How is testing  
NFRs different to  
functional  
requirements.

More unlikely to have easy Oracles to point at

More unlikely that what you are testing for will be present in the specification.

Less likely that the business will prioritise it.

# Break

See you in 10 mins



**How do we structure  
our findings?**



# How should findings be structured?

## Why does it matter?

*“You will understand how to interact with testers to optimise the user interface.” – Knowledge, Competencies and Behaviours Doc.*

Storytelling is key

Types of ways you may convey testing:

Information on what you've tested

Information on what you haven't tested

Information on what you've found

Bug reports

# JESSICA

Based on other things

Including PEOPLE WORKing from Michael Bolton's version of the RST course

But with more accessible language

Added focus on being sympathetic



## Justification

What makes you think that this  
is a problem?

I expected there to be cake because  
you advertised there would be cake in  
your advert.

**J**essica

Where did you notice a problem?

For informal bug reports this may feel unnecessary.

It was in the cafe here.

It's still necessary.

j**E**ssica



## Sympathetic

Try to remember the people  
who work on it are people.

No one likes bad news

Don't be accusatory

Benefit of the doubt

je**S**sica

You said you'd bring me cake and  
you haven't, you incompetent  
loser, I hate you.



## Scope

How general is the problem?

Could the problem be more general or less general?

Could it be more extreme or less extreme than what you have found?

Is it just on one browser or all of them?

Is it potentially indicative of a bigger issue?

jes**S**ica

The person next to me has cake.  
In fact everyone else has cake.  
So the problem isn't wider, it's just me.  
In fact I have a cup of tea so it's not affecting drinks maybe, just food?

## Illustration

To illustrate the problem.

- Video of issue
- An example of when it's an issue
- Or a screenshot
- Or a specific occasion where it will occur

jesslca





Clear

It's a story of what has happened.

Write it like a story so that people can understand it.

Remember that someone else needs to understand what you are talking about.

dont have it expected summat. In window saw thing said cake. Sad.

jessi**C**a





Adequate details to recreate

There need to be enough details to  
recreate what has been done to  
observe the problem.

jessic**A**

I sat down at this chair at this specific  
time

At this specific table

I was the fifth person in the cafe

I was led to believe from the poster  
outside that I would be given cake as  
soon as I sat down

I don't have cake



Recap

Justification of why you think it's an issue

Environment

Sympathetic

Scope

Illustration of the problem

Clear

Adequate details to recreate



# JESSICA but with NFRs

Is it not the same?

Well, yes and no.

Your oracle is unlikely to be your spec

It's unlikely to be black and white as to whether it matches expectations,  
and more of a grey issue

# Try it yourself

Write a bug report for a **usability** or **accessibility** issue you have found today.

Show your report to one of the mentors.

You should be able to just use a text editor to type your report.

Justification of why it's an issue

Environment

Sympathetic

Scope

Illustration of the problem

Clear

Adequate details to recreate

**Justification of why it's an issue**

**Environment**

**Sympathetic**

**Scope**

**Illustration of the problem**

**Clear**

**Adequate details to recreate**

- Do people ever find themselves raising issues or adding to existing reports?
- Are you used to considering these points when you raise issues?
- Does anyone have anything that they think should be reported when you raise a bug which isn't listed here?



**Wrap up**



You will understand how to interact with testers to optimise the user interface.



You will have experience of constructing, interpreting and executing test plans with a focus on accessibility and usability.



## More Sources

Very accessible intro to exploratory testing - Explore It! By Elisabeth Hendrickson

<https://www.developsense.com/> : Rapid Software Testing Course

Perfect Software: And Other Illusions about Testing by Gerald Weinberg

<https://www.istqb.org/> :International Software Testing Qualification Board

<https://glossary.istqb.org/search/oracle> : if ever testers are using words you haven't come across, the glossary can be useful.

My details: [david.duke@booking.com](mailto:david.duke@booking.com)

@allcapstester

