AGILE EXPLORATORY TESTING

London Testers Gathering 2016
Check in

Usage of a laptop, iPad / tablet or smartphone is recommended
If you do not have one pair up with somebody who has

Exercises will be both individual and in groups
Find yourself a group of four* people

This is a learning session
Talking and asking questions is allowed

If there is something that could distract you from participation let me know
<table>
<thead>
<tr>
<th>1 Introduction</th>
<th>2 From exploration to testing</th>
<th>3 Bringing structure to chaos</th>
<th>4 Managing Exploratory Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>• What is exploration</td>
<td>• What is testing?</td>
<td>• What are heuristics?</td>
<td>• ET Model</td>
</tr>
<tr>
<td>• Exploration skills</td>
<td>• Just doing it</td>
<td>• Coverage</td>
<td>• SBTM</td>
</tr>
<tr>
<td>• Exploration challenges</td>
<td>• What are oracles?</td>
<td>• Mnemonics</td>
<td>• What about agile?</td>
</tr>
<tr>
<td></td>
<td>• Oracle problem</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Introduction
1 – What is exploration?

Exercise:

Write ten things about where you are sitting right now that you hadn’t noticed when you sat down. Use your senses. Do it quickly, without censoring.

5 minutes
1 – What is exploration?

**Exploration** is the act of searching for the purpose of discovery of information or resources.
1 – What is exploration?

**Exploration** is the act of searching for the purpose of discovery of information or resources.

**Key elements:**

- The act of searching: Activity
- Purpose of discovery: Goal or direction
- Information: Something worth knowing
- Resources: Something that helps to advance
1 – Exploring Software

Exploring software is a technical activity:

There are many possible answers to questions you might have and the best answer is not immediately or easily available.

It is a situation that is complex enough to require the assignment of a specialist…

(engineer, developer, architect, software tester, etc.)

...to assess the question and organize resources as necessary to find an answer
1 – Exploration Attributes

Successful technical explorations share these attributes:

• They are self-directed, as opposed to dictated from outside

• They are cyclic, not linear

• They are mission and problem-driven, as opposed to procedure-driven. Learning is part of the mission

• We accept that the process is not fully predictable

• We accept the risk that some of our efforts may not pay off
1 – Modeling Types

In exploration you aim to create a **sense making model** where the data precedes the framework.

This is opposite from a **categorisation model** where the framework precedes the data.

**Why does this difference matter?**
Making Sense of Sensemaking 2: A Macrocognitive Model
Gary Klein, Brian Moon, and Robert R. Hoffman
1 – Categorising

**ESFP**
Pragmatists who love excitement and excel in a crisis.
1. Detective
2. Banker
3. Investor
4. Entertainment agent
5. Sports coach

Straightforward and honest people who prefer action to conversation.
1. Civil engineer
2. Economist
3. Pilot
4. Data communications analyst
5. Emergency room physician

Lively and playful people who value common sense.
1. Child welfare counselor
2. Primary care physician
3. Actor
4. Interior designer
5. Environmental scientist

Warm and sensitive types who like to help people in tangible ways.
1. Fashion designer
2. Physical therapist
3. Massage therapist
4. Landscape architect
5. Storekeeper

**ENTJ**
Natural leaders who are logical, analytical, and good strategic planners.
1. Executive
2. Lawyer
3. Market research analyst
4. Management/Business consultant
5. Venture capitalist

Creative perfectionists who prefer to do things their own way.
1. Investment banker
2. Personal financial adviser
3. Software developer
4. Economist
5. Executive

People-lovers who are energetic, articulate, and diplomatic.
1. Advertising executive
2. Public relations specialist
3. Corporate coach/Trainer
4. Sales manager
5. Employment specialist/HR professional

Thoughtful, creative people driven by firm principles and personal integrity.
1. Therapist/Mental health counselor
2. Social worker
3. HR diversity manager
4. Organizational development consultant
5. Customer relations manager

Myers Briggs Type Indicator
1 – Exploration, not black and white

Jon Bach; Agile2010, Telling your exploratory story
From exploration to testing
2 – What is testing?

“Testing is the process of executing a program with the intent of finding errors.”
Glenford J. Meyers; The art of software testing [1979]

“The purpose of testing a program is to find problems in it”
Cem Kaner, Jack Falk, Hung Quoc Nguyen; Testing Computer Software [1988]

“A tester is given a false statement (‘the system works’) and has the job of selecting, from an infinite number of possibilities, an input that contradicts the statement. … [You want to find] the right counterexample with a minimum of wasted effort.”
2 – What is testing?

“Testing is the process of evaluating a product by learning about it through exploration and experimentation, which includes to some degree: questioning, study, modeling, observation, inference, etc.”


“Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. The information itself is compared against specifications, business requirements, competitive products, past versions of this product, user expectations, industry standards, applicable laws and other criteria. So that the information becomes meaningful for those who receive it.”
2 – Let’s explore

Http://getdoneapp.com/login

Requires a Google+ account

http://arborosa.com/games/Horse_Lunge_Game_With_Dressage_by_eponagirl.swf

10 minutes
10 minutes
2 – Let’s explore (mobile alternative)

Download MOOVIT from app store or play store
2 – How to judge if a bug is actually a bug?
An oracle is a heuristic principle or mechanism by which you recognize a potential problem.

Oracles help to tell whether a test passed or failed.

Oracles help to evaluate test results.
2 – Oracles from the inside out

experience

your

feeling & mental models

conference

stakeholders’ feelings & mental models

observable consistencies

reference

shared artifacts (specs, tools, etc.)

michael Bolton – emotions in testing 2014
3 – Oracles applied
Color game

http://arborosa.com/colorGame/colorGame.html
Wikipedia
“Exploratory testing is an approach to software testing that is concisely described as simultaneous learning, test design and test execution.”

“Exploratory testing is simultaneous learning, test design and test execution”

Michael Bolton (2010)
“Exploratory testing is parallel learning, test design, and test execution”
2 – Exploratory Testing

Cem Kaner (2006)
“Exploratory Testing is a style of software testing that emphasizes the personal freedom and responsibility of the individual tester to continually optimize the value of their work by treating test-related learning, test design, test execution, and test result interpretation as mutually supportive activities that run parallel throughout the project”

Rob Sabourin (2006)
continuous test design as testing continues; continuous testing as design continues; continuous test planning as testing continues”

Elisabeth Hendrickson (2011)
“Exploratory testing is simultaneously learning about the system while designing and executing tests, using feedback from the last to inform the next”
Exploratory Testing is a way of learning about the system, investigating it through concurrent design, execution, experimentation and reporting of tests enabling you to provide answers for both currently known and, as yet, unknown questions.

Jean-Paul Varwijk (2016)
Bringing structure to chaos
3 – Heuristics

**Billy Vaughn Koen** (1985)

A heuristic is anything that provides a plausible aid or direction in the solution of a problem but is in the final analysis unjustified, incapable of justification, and fallible. It is used to guide, to discover and to reveal.

A heuristic does not guarantee a solution

It may contradict other heuristics and still be useful

It reduces the search time in solving a problem

Its acceptance depends on the immediate context instead of on an absolute standard
‘A heuristic is a method for finding the solution to a problem. It’s essentially a plausible guess, or a mechanism that helps generate plausible guesses.”

James Bach
3 – Oracles and Heuristics

An oracle is a principle or mechanism by which we recognize a problem.

Oracles are heuristic, but not all heuristics are oracles.

An oracle is a *heuristic* principle or mechanism by which we recognize a problem.

Oracles are fallible and context-dependent. An oracle can only point you to a possible problem.

We can’t see a problem unless we have an oracle for that problem: that is, a principle or mechanism for recognizing that problem.
A coverage outline is an artifact that provides an inventory of the dimensions or elements of a product by which it could be tested.
Structure (what the product is):
What files does it have? Do I know anything about how it was built? In what parts can I split it up? Is it one program or many? What physical material comes with it? Can I test it module by module?
Function (what the product does):
What are its functions? What kind of error handling does it do? What kind of user interface does it have? Does it do anything that is not visible to the user? How does it interface with the operating system?
3 – (sf) Data (ipot)

**Data (what it processes):**

What kinds of input does it process? What does its output look like?
What kinds of modes or states can it be in? Does it come packaged with preset data?
Is any of its input sensitive to timing or sequencing?
Interface (how does it use connections):
Does it have import/export functions? Does it store intermediates? Is an api provided?
Does one exists that isn't provided?
Does it interact in the ways that a user will expect that kind of interface to interact.
3 – (sfdi) Platform (ot)

Platform (what it depends upon):
What operating systems does it run on? Does the environment have to be configured in any special way? Does it depend on third-party components?
Operations (how it will be used):
Who will use it? Where and how will they use it? What will they use it for? Are there certain things that users are more likely to do? Is there user data we could get to help make the tests more realistic?
3 – (sfdipo) Time

Time (how time has influence):
Does the application run for or at specific times? What happens on calendar events?
What if runs too slow? Too fast? Does it process simultaneously or sequentially? Is there a cache?
Input Interface

Platform

Output Interface

http://arborosa.com/parkingCalc/ParkCalc_JPV.html

15 minutes
10 minutes
Managing exploratory testing
4 – Exploratory Testing Model
4 – Concurrent Testing

- Testing is executed concurrent with analysis, design and development throughout a sprint
- Breaking down work in small items, e.g. user stories which can be analyzed, designed, developed and tested in parallel
- Improve awareness and responsibility of all team members for the software quality
- Reinforce the concept of effective feature teams working in parallel which creates focus and collaboration
- Everyone is involved in testing
A Test Idea is a brief statement about a test that could be performed.

Test ideas ensure that important ideas are not forgotten and can be detailed later in test cases or test charters.

Heuristics, requirements documents, quality attributes, bug lists, experience or mnemonics and checklists are a good source of test ideas.
4 – Risk catalog

Specify types of problems to look for

Construct tests (on the fly) to find each one

It is a pragmatic variation on the concept of identifying product risks

Prioritise them based on potential severity
An exploratory test session often begins with a charter

A charter states the mission and perhaps some of the tactics to be used

A charter may be chosen by the tester himself, or assigned by the test lead or test manager

A charter is a 1-to-3 sentence mission for a testing session
4 – Template charter

**Target**
Where are you exploring?
Which feature, requirement, or part (or module) of the system?

**Resources**
What resources will you use?
A tool, data set, technique, configuration, or perhaps an interdependent feature.

**Information**
What kind of information are you hoping to find?
About security, performance, standards, consistency, -ility, or some other aspect of the system?
4 – Charters and test cases

*Test charters* help you see the system from many perspectives at once.

*Test cases* worth repeating = subset of Testing performed
4 – Test case types

**Pre-scripted**
The test case is based on some kind of reference (documented) and its steps, actions and expected results are determined in advance

**Post-scripted**
While investigating the system a sequence of activities and/or results is noticed that is worth to be repeated. Either in its current form or with possible variations
4 – Test case types

Chartered
The investigation of the system aimed to cover a particular area or attribute of the system and its execution is recorded for the purpose of conveying information and as evidence to support test results.

Recorded
The investigation of the system is recorded for the purpose of conveying information and as evidence to support test results or (new) test ideas.
4 – Charter preparation

• Preparation  -  Domain knowledge?
  Tool knowledge?

• Set-up  -  Environment
  Software (version)
  Tooling

• Handling distraction  -  Private space
  No mail, no phone, no apps, etc.

• Understand the goal  -  What’s the charters objective?

• Don’t forget the reporting
A typical session last between 60 to 90 minutes
But not all of the time is spent on actual testing!

A division for session activities and the time spent is:

- **Test design and execution**
- **Bug investigation and reporting**
- **Session setup**
4 – Session sheet (example)

Charter
Target areas covered

Time
Start Time + Session duration

Tester Name(s)

Charter/opportunity time
Expressed as a time/TBS breakdown (% of on-charter time)

Data Files
Any files the tester used or created to support their testing

Test Log
Notes made during testing

Bugs or issues
A problem that threatens the value of the product or of the testing process
4 – Debriefing

Apart from walking through the session sheet, you could use an agenda summarized by the acronym PROOF

- **Past**: What happened during the session?
- **Results**: What was achieved during the session?
- **Obstacles**: What got in the way of good testing?
- **Outlook**: What still needs to be done?
- **Feelings**: How does the tester feel about all this?
4 – Metrics

- Number of sessions completed (absolute / percentage)
- Number of problems found
- Test areas covered (absolute / percentage)
- Percentage of session time spent on Test design and execution
- Percentage of session time spent on Bug investigation and reporting
- Percentage of session time spent on Session setup
Take home exercise

Now apply what you have learned

Write a charter that you will use at work

Add which oracles you are planning to use
Questions?!