EXPLORATORY TESTING

Purpose : Our team will become more effective in finding important bugs sooner by increasing the amount and focus of exploratory testing that we do.

Exploratory Testing Defined

- Exploratory testing is simultaneous learning, test design, and test execution.

- James Bach

Exploratory Vs. Scripted

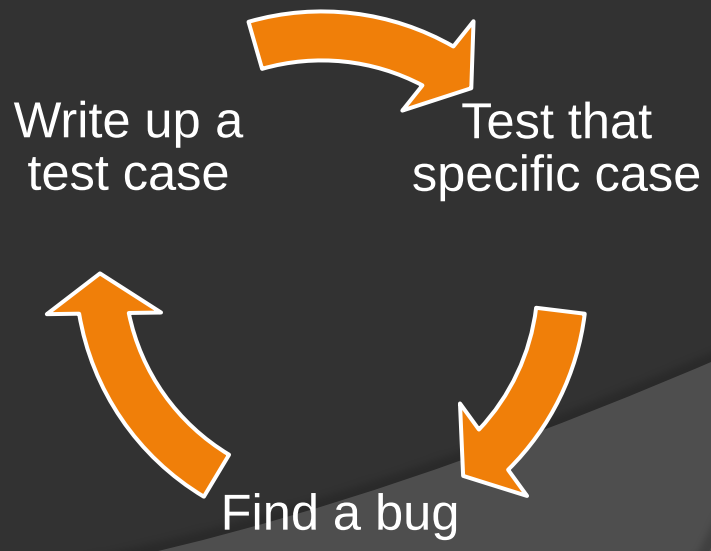
- Scripted testing is running a test following a defined set of steps written in advance.
- Scripted could be automated – but in our case it's generally manual.

Elizabeth Hendrickson

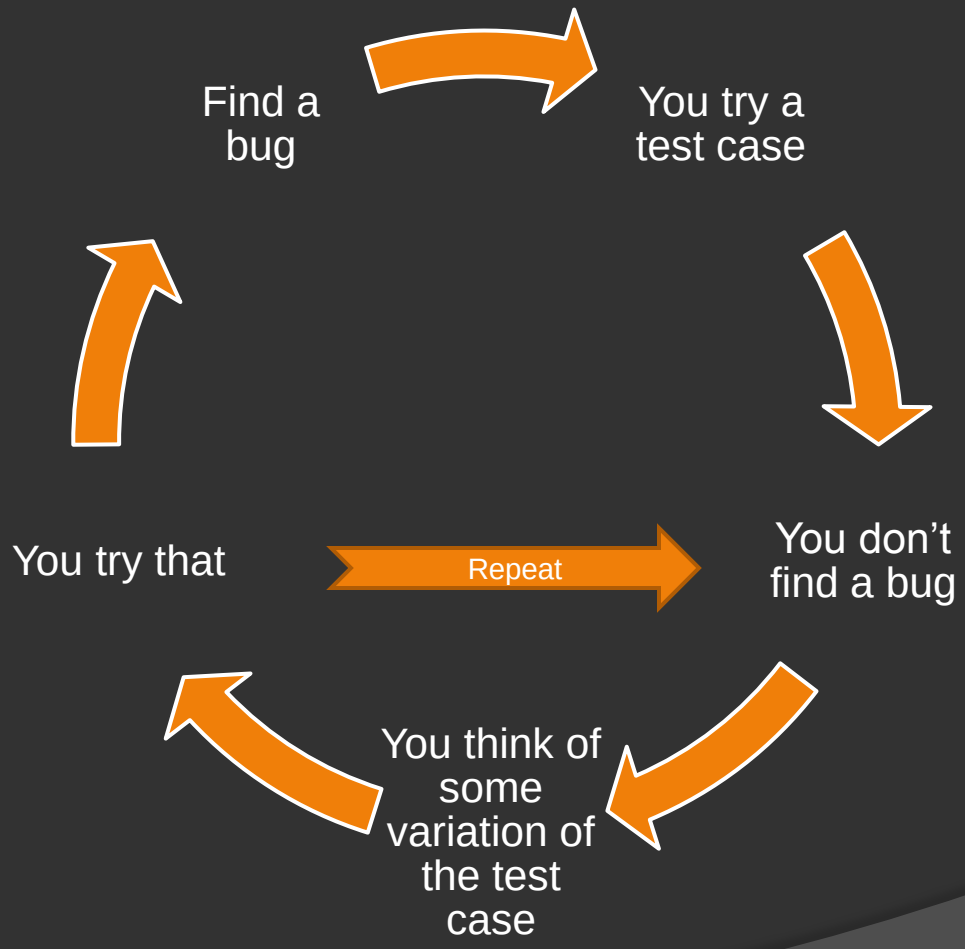
"Exploratory testing is a style of testing in which you explore the software while simultaneously designing and executing tests, using feedback from the last test to inform the next. Exploratory testing helps us find surprises, implications of interactions that no one ever considered, and misunderstandings about what the software is supposed to do."

How do you find Bugs?

- How often does this happen to you?
  - You write up a test case
  - You try out that specific test case
  - You find a bug.



How do you find bugs?



- How many of those unfruitful test cases do you document?
- How many of them do you think you could have even dreamed of before you got to the page and were using it?

Exploratory Testing Defined

- Exploratory Testing is a way of thinking about testing. Any technique can be applied in an exploratory way (i.e. load, regression, functional, usability)
- It's something we all do already – to differing degrees.

Exploratory Testing Defined

- Exploratory Testing is sometimes also called ‘Ad hoc’ testing. But we need to avoid the negative connotations associated with that term.
- Exploratory testing is NOT sloppy, careless work.

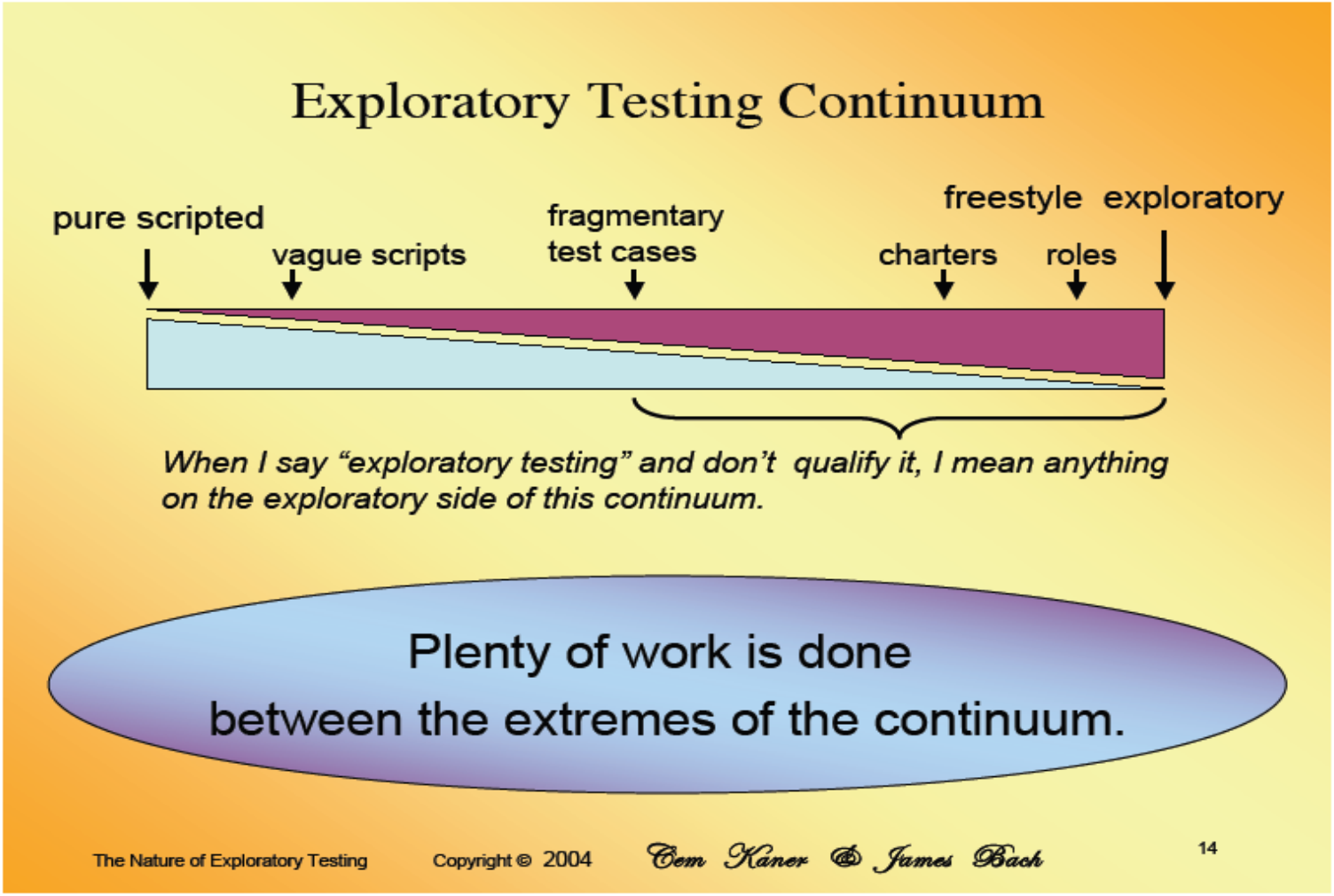
Testing is a creative process. We are not trained monkeys pushing buttons and pulling levers by rote.

Exploratory Testing is NOT -

‘PLAYING AROUND’

Exploratory testing is, or rather should be, a disciplined, intellectual activity.

From Cem Kaner and James Bach:



James Bach

"Exploratory testing can be described as a martial art of the mind. It's how you deal with a product that jumps out from the bushes and challenges you to a duel of testing."

Ninjas!!

1) You are handed off software with little more than a handshake and a vague list of bugs/features.
2) You are told that some application (Common, anyone?) which effects one of yours will be going out and you need to “make sure nothing broke”.
3) The PM (or dev) tells you that the app *must* be in production by a certain date. You don’t have time to test everything. You’ll just have to ‘do your best’.
4) You are asked to jump in to help out on a project and get up to speed quickly so you can assist the test effort.
5) You feel like you need to test parts of the app that are “unchanged” – but you can’t really justify why.
6) You are bitten by a bug that was not represented anywhere in your test cases.
7) You read the steps to duplicate a bug, or notes on how it was fixed and still don’t know exactly what is expected.

Scenarios

1) You are given an app – or maybe just a piece of the app. You have no documentation yet, no FDD, no flowcharts and you've only been involved in a scattering of meetings.

WHAT DO YOU DO?

Scenarios

2) You are given an app, complete with an FDD, flow charts and beautiful use cases.

WHAT DO YOU DO?

Scenarios

3) You are asked to regression test Common.

WHAT DO YOU DO?

How is this different from ‘Playing’?

- You are making choices and tracking the results in some fashion
- You have a general plan of attack but allow yourself to deviate from it for short periods of time (Bach)
- You have a mission.

Where Exploratory Testing Fits

(from James Bach)

- You need to provide rapid feedback on a new product or feature.
- You need to learn the product quickly.
- You have already tested using scripts, and seek to diversify the testing.
- You want to find the single most important bug in the shortest amount of time.
- You want to investigate the status of a particular risk, in order to evaluate the need for scripted tests in that area.

Where Exploratory Testing Fits

(from James Bach)

- Interpreting vague test instructions.
- Product analysis and test planning.
- Writing new test scripts.
- Regression testing based on old bug reports.
- Testing based on reading the user manual (or FDD) and checking each assertion.

When do you use Scripted?

- There are times when you must use a purely scripted approach
  - When any part of the test may be controversial.
  - When it will be subject to audit.

Session Based Testing – from Wikipedia

- Definition : A software test method that combines accountability and exploratory testing to provide rapid defect discovery, creative on-the-fly test design, management control and metrics reporting. The method can also be used in conjunction with Scenario testing. Session-based testing was developed in 2000 by Jonathan and James Bach.

Charter

A goal or agenda for a test session.

- Charters are created by the test team prior to the start of testing, but may be added or changed at any time.
- Often charters are created from a specification, test plan, or by examining results from previous test sessions.

Session

- An uninterrupted period of time spent testing, ideally lasting one to two hours
- Each session is focused on a charter, but testers can also explore new opportunities or issues during this time
- The tester creates and executes test cases based on ideas, heuristics or whatever frameworks to guide them and records their progress.
  - This might be through the use of written notes, video capture tools or by whatever method as deemed appropriate by the tester.

Make sure you tag everything with the SMRF or Bug number you are working on.

Exploratory Charters

- Explore and analyze the product elements of Study Charges (STAGES). Produce a test coverage outline.
- Define use cases for T&E. These should reflect realistic scenarios for use. Try each one. Do *not* specify what to click…etc.
- Run each report listed in WebCollect and export the results to Excel.
- Attempt to cause a data truncation error by overfilling every input field (not search fields) on the Create Study Page
- Log into WebCollect and act like an Accountant (or Cashier) user.

What do you document?

- **Session Report**
- The session report records the test session. Usually this includes:
- Charter.
- Area tested.
- Detailed notes on how testing was conducted.
- A list of any bugs found.
- A list of issues (open questions, product or project concerns)
- Any files the tester used or created to support their testing
- Percentage of the session spent on the charter vs. investigating new opportunities.
- Percentage of the session spent on:
  - Testing - creating and executing tests.
  - Bug investigation / reporting.
  - Session setup or other non-testing activities.
- Session Start time and duration.

Make sure you tag everything with the SMRF or Bug number you are working on.

Payment Gateway Exploratory Log:

Exploratory Test of bug 7294

Some scheduled payments are being routed incorrectly to deposit file

Find out what the data look like in the tables and write queries

Hospitality vs. Other Web Collect

Look at xml in the logs

From the Payment Gateway logs
Param_Amount=4500&13

HIDDEN NAME=PARAM_AUTHORIZED_AMOUNT VALUE="4500"

Param_Avs_Indicator=1&14

4500"><INPUT TYPE=HIDDEN NAME=PARAM_MSDI VALUE="1"

?? Find out how to interpret the &13

Try several payments, watch logs and review phppyrs after completion of each

Payment for 10.00, created 3/19, scheduled for 3/22

[INFO][PaymentGatewayAdminBean] [03/19/2010 13:16:07.611] [PAYGATEWAY] [CNANDRAS] [0] [Committed new payment to database with receipt number 20105552]
[INFO][TerminalReadCommand] [03/19/2010 13:16:38.673] [PAYGATEWAY] [CNANDRAS] [0] [Acquire terminal id for facilityId <132> department <729> type <CR> terminalId <99988827> serviceTime <2010-03-19 13:16:38.47>]
[INFO][LoggerFactory] [03/19/2010 13:16:55.548] [PAYGATEWAY] [UNKNOWN] [0] [LoggerFactory is already initialized!]
[DEBUG][PaymentGatewayAdminBean] [03/19/2010 13:16:56.736] [PAYGATEWAY] [CNANDRAS] [0] [Wallet found from payment source]
Param_Card_Account_Nbr=xxxxxxxxxxxx1111&10=Set_Context_Param Param_Card_Exp_Date=0212&11=Set_Context_Param Param_CVV2_Data=xxx&12=Set_Context_Param Param_Amount=10&13=Set_Context_Param Param_Avs_Indicator=1&14=Set_Context_Param Param_Street=test&15=Set_Context_Param Param_Zip=84111&16=Send_Context]
[INFO][NovaViaConex] [03/19/2010 13:17:53.392] [PAYGATEWAY] [CNANDRAS] [0] [Verification response (receipt=20105552): <!doctype html public "-//w3c//dtd html 4.0 transitional//en"><html><head><meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"><title>SCRIPT RESPONSE</title></head><body><INPUT TYPE=HIDDEN NAME=PARAM_RESPONSE_CODE VALUE="ND"><INPUT TYPE=HIDDEN TYPE=HIDDEN NAME=PARAM_REFERENCE_NBR VALUE="0319191753"></body></html>]
[TRACE][NovaViaConex] [03/19/2010 13:17:53.392] [PAYGATEWAY] [CNANDRAS] [0] [Created verification result]
[
PAYMENT WAS DECLINED - actually payment was only for .10 not 10 dollars
- no record written to phppyrs

Used Exploratory log to create test cases :

Test Case	Description	Testing Steps	Validation Critieria	Da
Bugs 7294 and 7295				
	Try several payments, watch logs and review phppyrs after completion of each			
		Payment for 10.00, created 3/19, scheduled for 3/22	PAYMENT WAS DECLINED - actually payment was only for .10 not 10 dollars - no record written to phppyrs	
		redid payment for 10.00 - approved	approved PHPPYRS.PRORGA = 1000 In the XML - Param_Amount=10&13=Set_Context_Param Param_Avs_Indicator=1&14	
		Payment for 8.91 created 3/19, scheduled for 3/26	Param Param_Avs_Indicator=2 Set_Context_Param Param_Amount=000 PHPPYRS.PRORGA=0	
	Hospitality	Payment Taken on 3/25 scheduled for 3/26	Param_Avs_Indicator=1 Param_Amount=1222200&13	

Heuristics Cheat Sheets

- One way to generate ideas for an exploratory charter is to use a test heuristic.
- This helps our testing become more reliable and repeatable
- Our ‘common things to test’ page is one example.

Bibliography

- Bach, James and Kaner, Cem. “The Nature of Exploratory Testing” 2004 <http://www.testingeducation.org/a/nature.pdf>
- Bach, James. “Exploratory Testing Explained” 2003 <http://www.satisfice.com/articles/et-article.pdf>
- Hendrickson, Elizabeth. “Test Heuristics Cheat Sheet” 2006 <http://testobsessed.com/wordpress/wp-content/uploads/2007/02/testheuristicscheatsheetv1.pdf>
- Kohl, Jonathan. “Demystifying Exploratory Testing” 2010 <http://www.stickyminds.com/BetterSoftware/magazine.asp?fn=cifea>
- Wikipedia. “Session Based Testing” 2010 <http://en.wikipedia.org/wiki/Session-based_testing>