EXPLORATORY TESTING IN PRACTICE
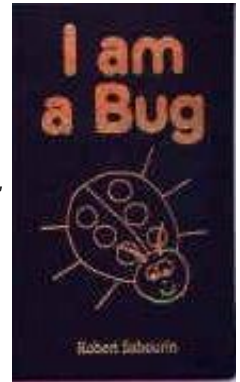
Instructor Introduction
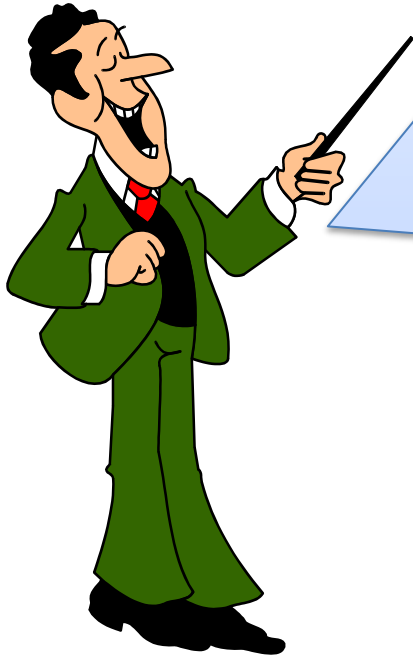
Presented By

Robert Sabourin

robsab@gmail.com

Exploratory Testing in Practice

- Robert Sabourin , *Software Evangelist*
- President
- AmiBug.Com Inc.
- Montreal, Quebec, Canada
- robsab@gmail.com

Table of Contents — Agenda

1. Example of Exploration

2. Chartering the Course

3. Framework for Exploration

1

EXAMPLE OF EXPLORATION

Bug Confirmation Testing

- Confirm that the "*fixed*" bug is really *fixed*

Exploratory Testing in Practise

Bug Confirmation Testing

- Was bug fixed?
- Did bug move?
- Did developers accidentally inject a new bug?

Bug Confirmation Testing

Was bug really fixed?

1. *Use repro steps from bug report*
2. *Confirm bug was in <u>old build</u>*
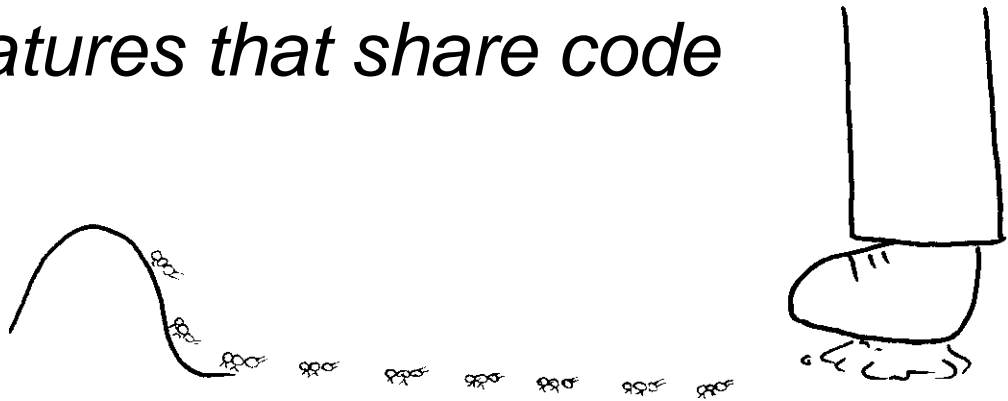3. *Show bug is not in <u>new build</u>*

Bug Confirmation Testing

Explore: Did the bug move?

1. *Exercise related capabilities*

2. *Vary steps to reproduce*
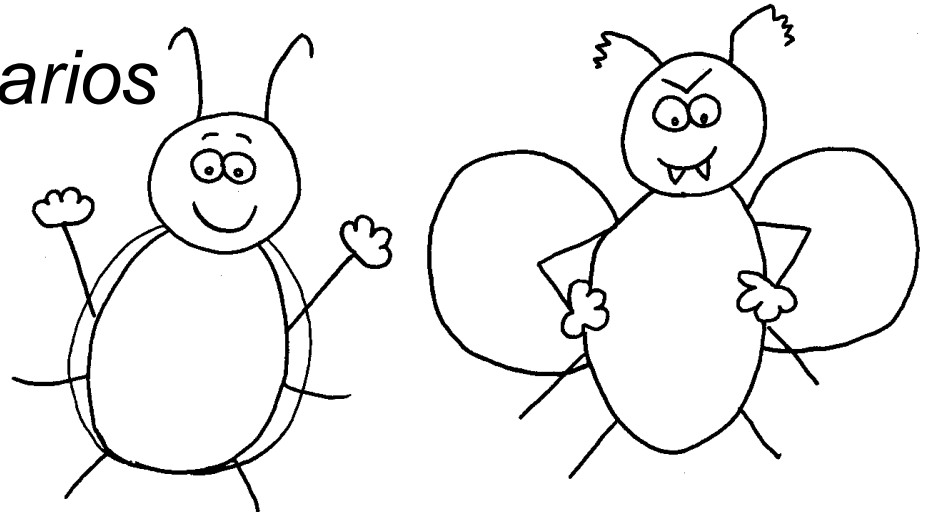
3. *Investigate features that share code*

Exploratory Testing in Practise

Bug Confirmation Testing

Explore: Did developers inject a new bug?

1. *Exercise application capabilities*

2. *Vary data*

3. *Walk through scenarios*

Exploratory Testing in Practise

Bug Confirmation Testing

Jason's Approach?

1. *Speak with developer*
2. *Confirm bug was in old build (repro)*
3. *Confirm bug is not in new build (repro)*
4. *Identify a usage scenario which would have exposed the bug*
5. *Walk through the scenario many times with variation of steps and user data*
6. *Stop when perturbation is absurd*
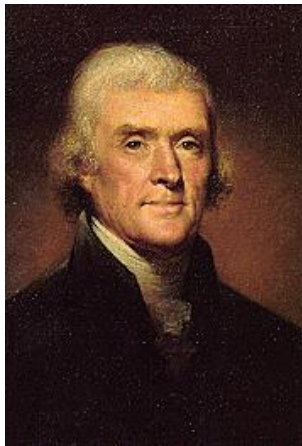
2
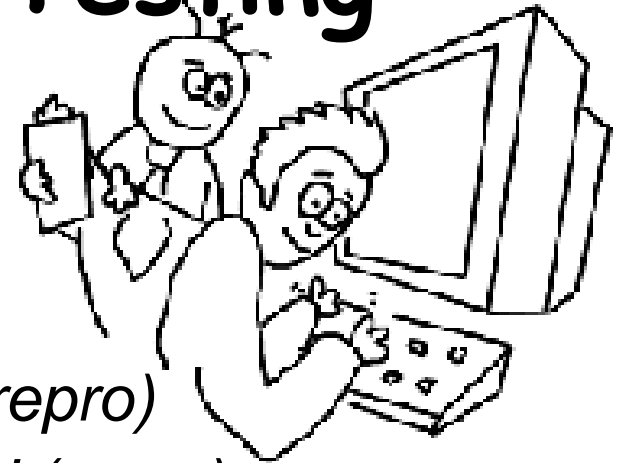
CHARTERING THE COURSE

Mandate to Explore

William Clark

Meriwether Lewis

The object of your mission is to explore the Missouri river, & such principle streams of it, as, by its course and communication with the waters of the Pacific ocean...may offer the most direct & practicable water communication across this continent for the purposes of commerce.

- Thomas Jefferson's letter to Meriwether Lewis, June 1803

Charter Statement

- Statement of mission

- Ties to purpose

- Focuses work

- Confirms understanding

- Delineates scope

Charter Types

- Capabilities
- Across Story Relationships
- Failure Modes
- Software Breaking
- Quality Factors
- Usage Scenarios

- End to End
- Creative Ideas
- States
- Sequences
- Data
- Environments
- White Box

Reconnaissance

- Investigative approaches
  - We become truffle snorting pigs and try to find useful information in all evidence we discover
 - We can even get good ideas from <u>out of date</u> sources

Charter Types

Reconnaissance Example

Project	**Determine if Freemind is a suitable tool to support Note Taking during session based exploratory testing.**
Application	**Freemind**
Version	**0.8.0**
Source	**SourceForge.Net**
OS	**Windows 10**
Customer	**AmiBug.Com, Inc.**
Session Size	**90 minute**
Charters	
FM000A-001	**Identify capabilities and areas of potential instability of Freemind.**
	Exercise each capability identified with data typical to the intended use of the application.

3

FRAMEWORK FOR EXPLORATION
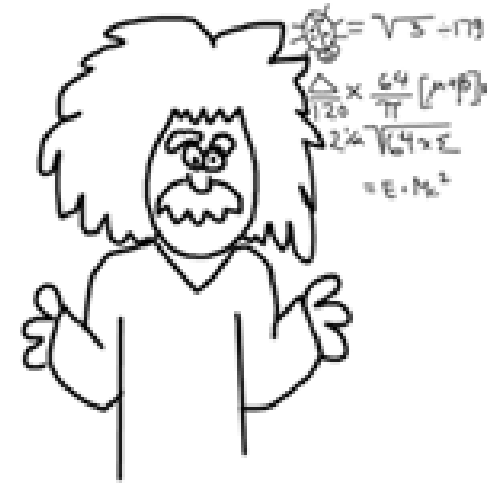
Framework for Exploration
TEST DESIGN

Decision Making

Bug Workflow

- Priority
- Severity
- Good enough?

Test Design

Variable identification

Domain analysis

Boundary testing

Test Design

Usage scenarios

Control flow

Decision tables

Test Design

State models
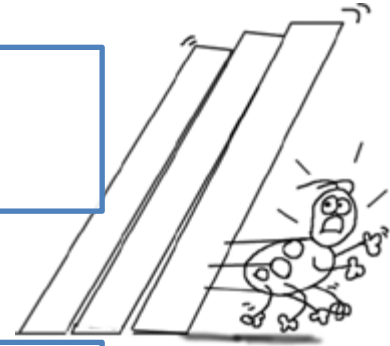
Pareto analysis

Pairwise tests

Test Design

Failure modes

Constrained pairs

Permutations

Framework for Exploration
ASSESSING CORRECTNESS

Assessing Correctness

Test Oracles

Test Oracles

Lifelines

Assessing Correctness

Test Oracles

 Call a friend

 Subject matter expert

 Test domain expert

Assessing Correctness

Test Oracles

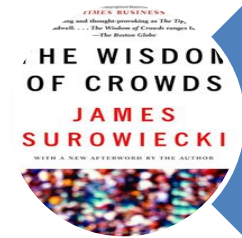|	Eliminate incorrect …
	Eliminate impossible …
	Rules of thumb

Assessing Correctness

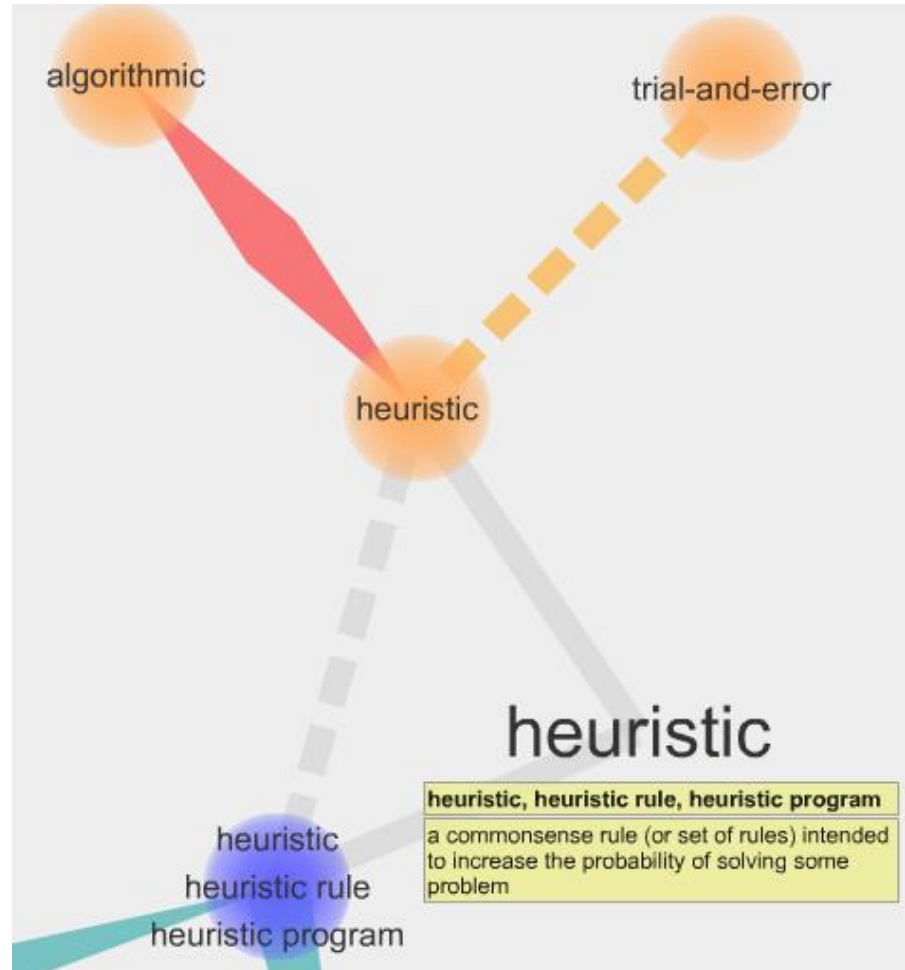## Test Oracles

 Ask the audience

 Wisdom of crowds variety

 Consult peers

Framework for Exploration
TEST HEURISTICS

Heuristics in Exploration

Heuristics in Exploration

Guide exploration

Rules of thumb

Fallible but useful

Heuristics in Exploration

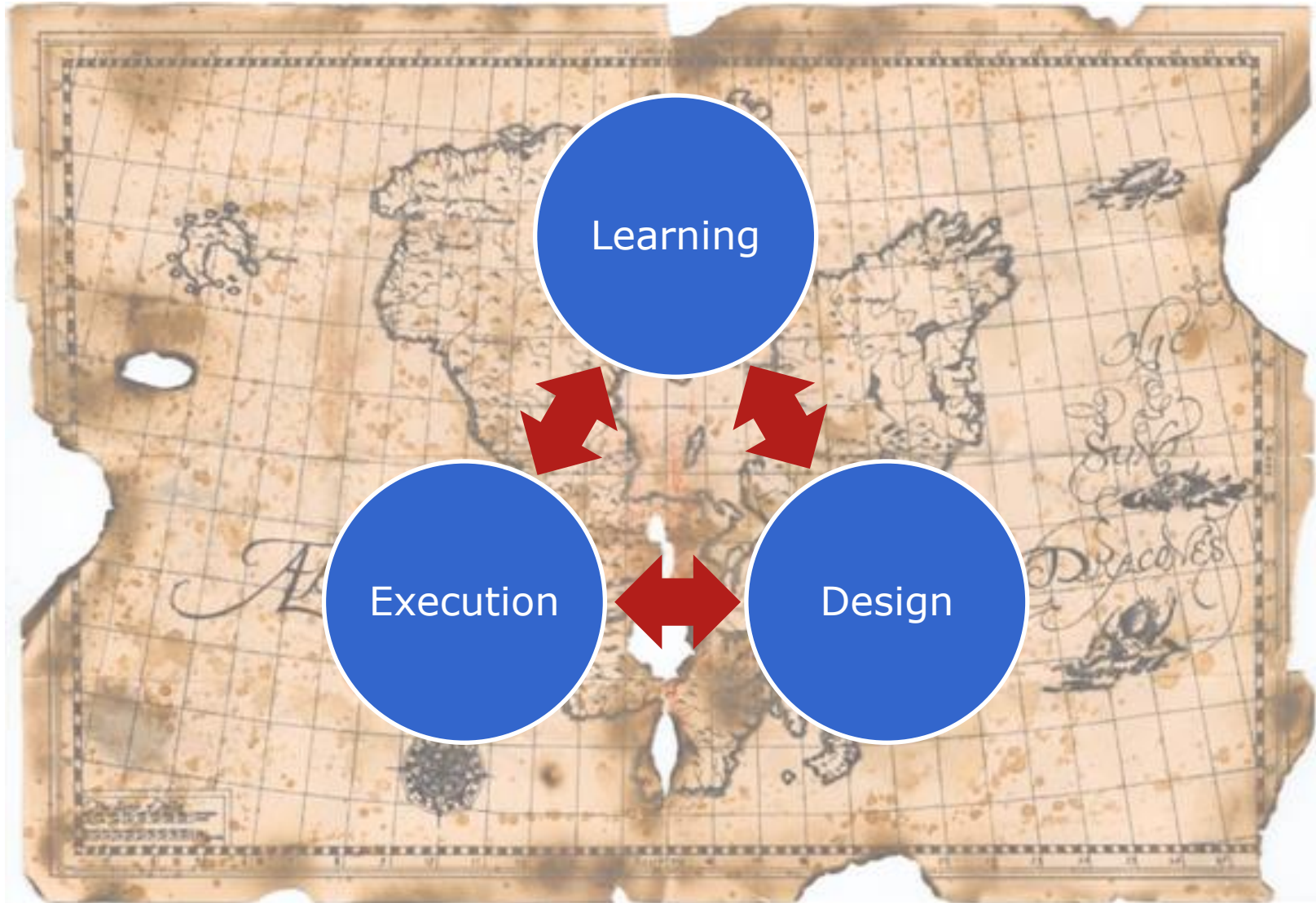|	
CRUD	• Create Read Update Delete
Goldilocks	• Too Big, Too Small, Just Right
Take AIM	• Application Logic, Input Method, Memory
Baby Steps	• Construct Progressively
Action/Effect	• Conditions trigger actions? • Actions caused by conditions?
Triangulate	• Confirm Screen, Database, Report

Framework for Exploration
EXPLORING AND LEARNING
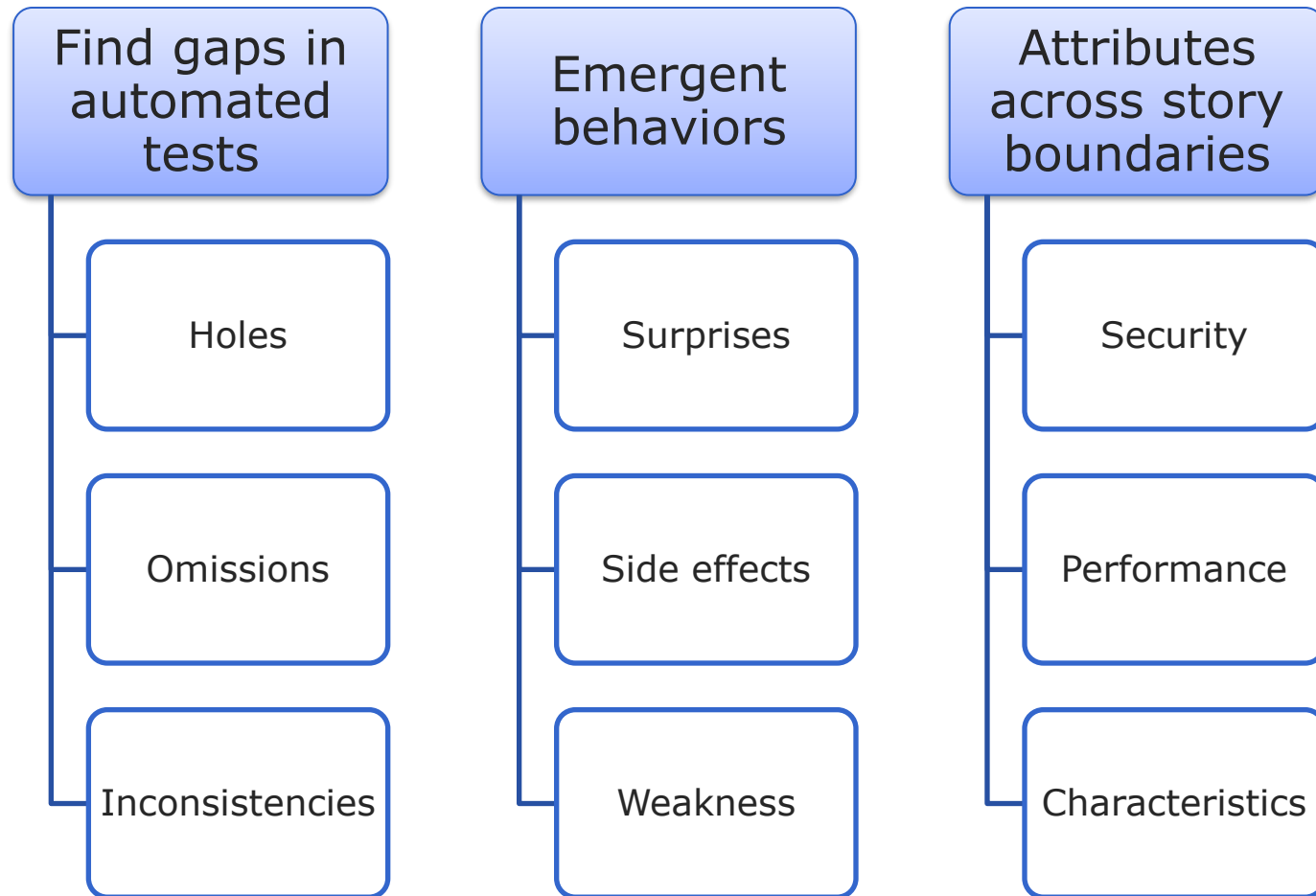
Exploratory Testing

Exploratory Testing

- *"Exploratory software testing is a style of software testing that emphasizes the personal freedom and responsibility of the individual tester to continually optimize the value of her work by treating test-related learning, test design, test execution, and test result interpretation as mutually supportive activities that run in parallel throughout the project."*

— Cem Kaner's Blog :
On the craft and community of software testing

Exploratory Testing: A New Perspective

Find gaps in automated tests
- Holes
- Omissions
- Inconsistencies

Emergent behaviors
- Surprises
- Side effects
- Weakness

Attributes across story boundaries
- Security
- Performance
- Characteristics

Framework for Exploration
EXPLORATION NOTES

Exploration Notes

Take notes about test decisions

Map out where you have been and what you discovered

Exploration Notes

- Tabular
- Chronological
- Schematic
- Point form
- Concise

Exploration Notes

- ## Test cases
  - Not known in advance
  - Defined and executed "on the fly" while you learn about the product
  - Document
    - What we tested
    - Test Data
    - Test Findings
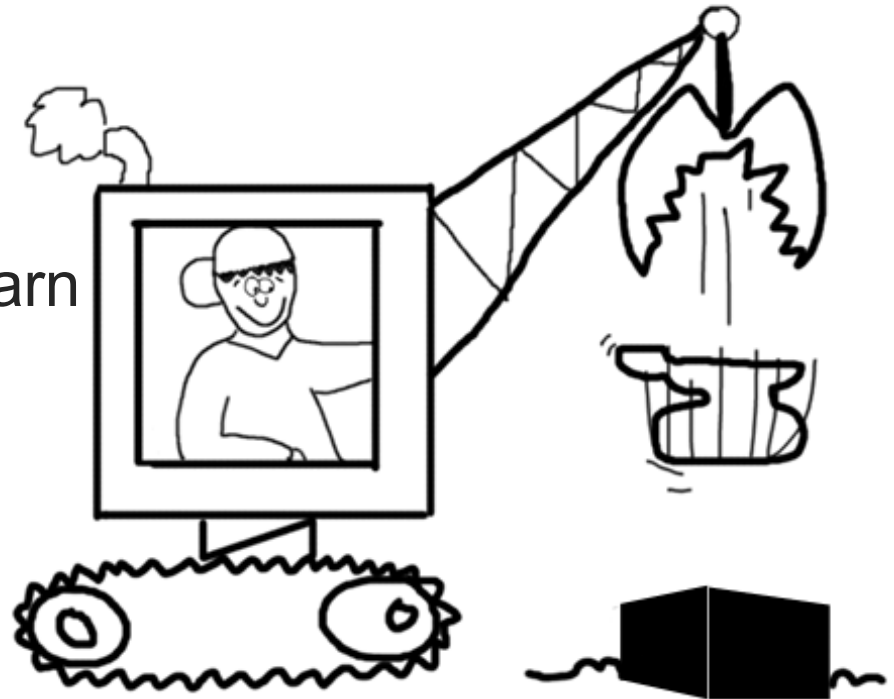
Exploration Notes

- Testers need to "hone up" their skills in making maps!
  - Consistent style
  - Clear
  - Terse
  - Peer review

Exploration Notes

- ## Note templates may be helpful
  - Uniform way to capture details
  - Great for teams to share findings
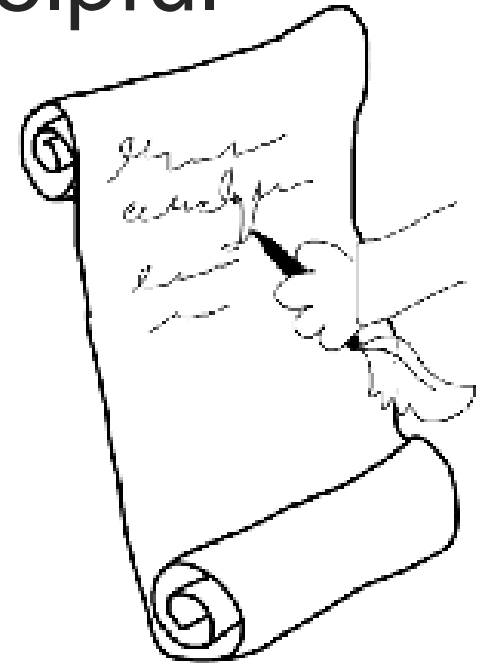
Exploration Notes

- ***Noteworthy during testing …***
  - Function, options or sub-functions being explored
  - Comments, notes, images or attachments
  - Hints, reminders and observations which may be useful to future testers
  - Date, Platform, Build or Configuration under test
  - Name of person running test
  - Oracles, "strategy to assess correctness"
  - Other relevant details

Exploration Notes

- ***How time was spent in session***
  - Set up
  - On charter testing
  - Off charter testing
  - Interruptions
  - Bug investigation
  - Bug logging
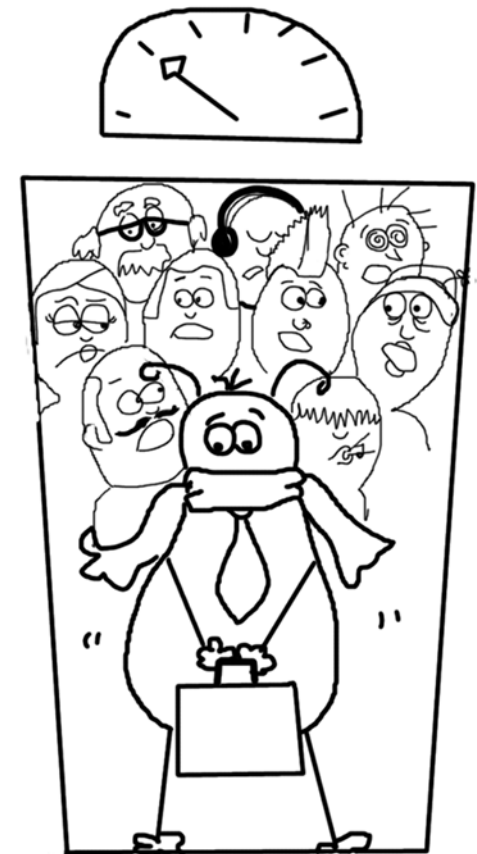
Exploration Notes

- ## **_Do_**
  - Keep terse, to the point but clear
  - Stick to template
  - Use timestamps
  - Professional tone
  - Cross reference bug reports
- ## **_Do Not_**
  - Write a novel
  - Use casual tone
  - Point fingers
  - Rewrite a bug report
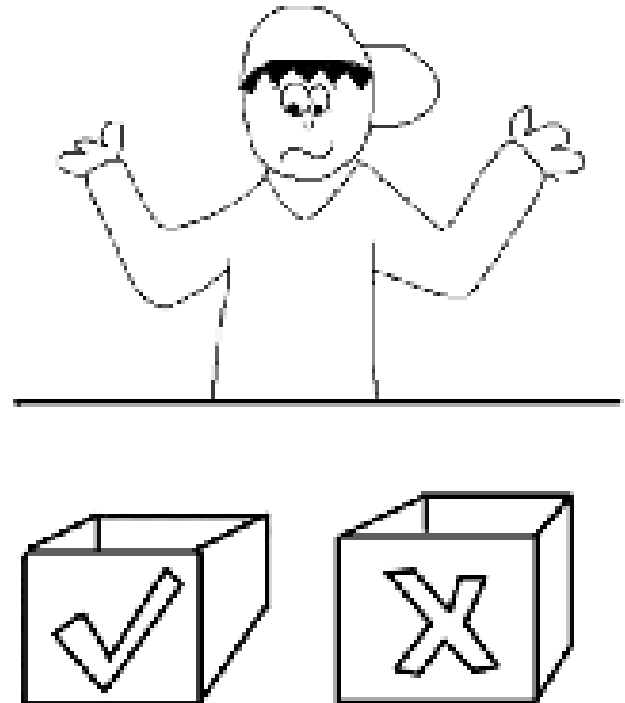
Exploration Notes
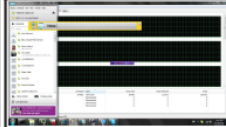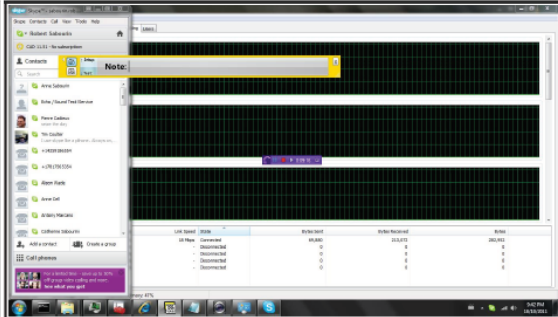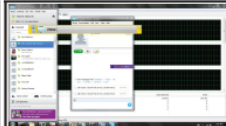
Session Report | Powered by [Rapid Reporter](#)

☑ Show autogenerated rows
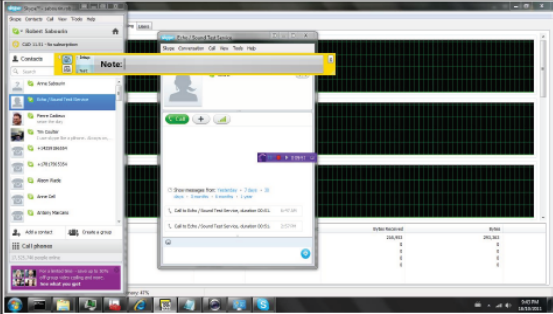
Time	Reporter	Type	Content	Screenshot	RTF Note
18/10/2011 9:36:36 PM	Robert Sabourin	**(Rapid Reporter version)**	1.11.03.20		
18/10/2011 9:36:36 PM	**Robert Sabourin**	**Session Reporter**	**Robert Sabourin**		
18/10/2011 9:36:36 PM	**Robert Sabourin**	**Session Charter**	**Explore amount of data billed vs used on TMoble 4G Hotspot**		
18/10/2011 9:36:48 PM	Robert Sabourin	**Setup**	WIndow 7; Dell Inspiron		

Exploration Notes

18/10/2011 9:41:11 PM	Robert Sabourin	**Note**	Test 1 Skype with Test Call		
18/10/2011 9:42:32 PM	Robert Sabourin	**Note**	Launch skype		
18/10/2011 9:42:42 PM	*Robert Sabourin*	*autogenerated*	*screenshot saved*		
18/10/2011 9:43:03 PM	Robert Sabourin	**Note**	Run Test Call with Sound Test Service		
18/10/2011 9:43:15 PM	*Robert Sabourin*	*autogenerated*	*screenshot saved*		

Exploration Notes

18/10/2011 9:43:20 PM	Robert Sabourin	**Note**	Make call	
18/10/2011 9:44:30 PM	Robert Sabourin	**Note**	Call complete	

Framework for Exploration
TESTING SESSIONS

Timeboxed Sessions

Typical
- About two hours
- Uninterrupted
- Focused

Short
- 15–30 minutes
- Punctual
- Confirmations

Long
- Several hours or days
- Experiments
- Non-functional charters

At End of Session

Press pause button
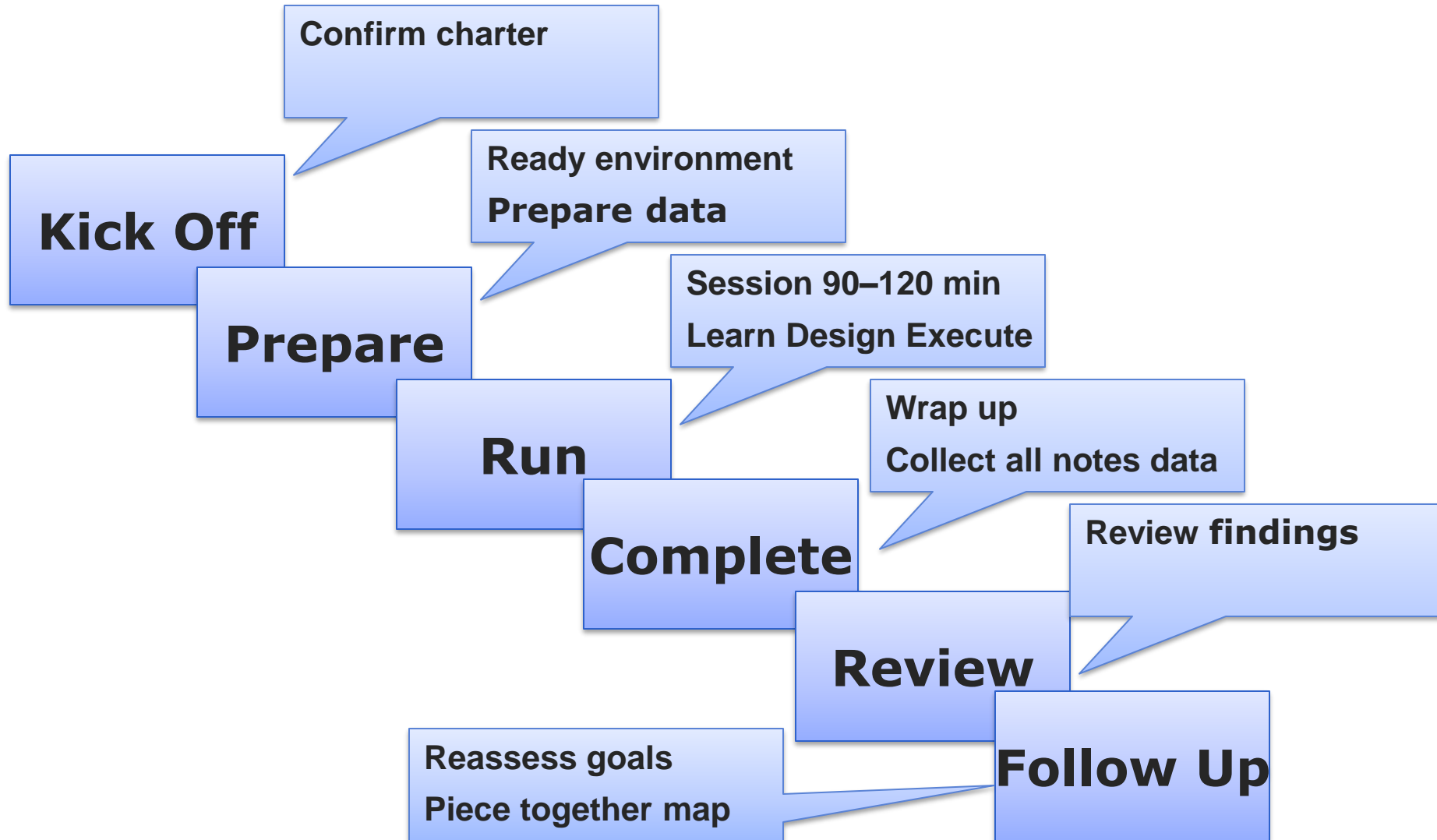
- After timebox we stop
- Can resume later

Review findings

- What works?
- What does not work?
- What did we learn?
- What do we want to learn more about?

Did we learn enough?

- Will we benefit from another session on the same charter?
- Should we try a new charter?

Session-based Exploratory Testing

Kick Off

Confirm charter

Prepare

Ready environment
Prepare data

Run

Session 90–120 min
Learn Design Execute

Complete

Wrap up
Collect all notes data

Review

Review **findings**

Follow Up

Reassess goals
Piece together map

Framework for Exploration
PAIRED EXPLORATION

Paired Exploration

Other testers
- Improve observation
- Feed off other's experiences
- "XP" ish

Developers
- White box perspective
- What is it programmed to do?
- What is it supposed to do?

Paired Exploration

Subject matter experts
- Assess correctness
- Drive from business perspective
- Identify missing functionality

Test domain experts
- Usability gurus
- Performance engineers
- Security experts

Paired Exploration

Administrators

- How will system be administered?
- Support concerns?
- Conflict with other applications?

End Users

- User experience
- Human factors
- Inconsistency

Paired Exploration

Novice User
- New to software
- Learnability
- Questions illuminate UX

Expert User
- Seasoned user
- Efficiency
- Missing capabilities