

# Exploring Exploratory Testing

Andy Tinkham, CSQA, CSTE  
Florida Institute of Technology

[andy@tinkham.org](mailto:andy@tinkham.org)

Copyright © 2003, Andy Tinkham & Cem Kaner

This research was partially supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

# About Me

- Master's student at Florida Tech
- Pursuing degree in Software Engineering, with focus on Software Testing
- Thesis topic is exploratory testing and how to more effectively train testers in exploratory techniques



Copyright © 2003, Andy Tinkham & Cem Kaner

This research was partially supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

# Learning Objectives

- Realize what good exploratory testing entails
- See why testers should plan to use exploratory techniques in their day to day testing activities
- Discover how to use risks to define specific exploratory tests
- Learn simple heuristics to help in risk identification

# What is Exploratory Testing?

## ● LAWST VII Attributes

- Interactive
- Concurrence of cognition and execution
- Creativity
- Drive towards fast results
- De-emphasize archived testing materials



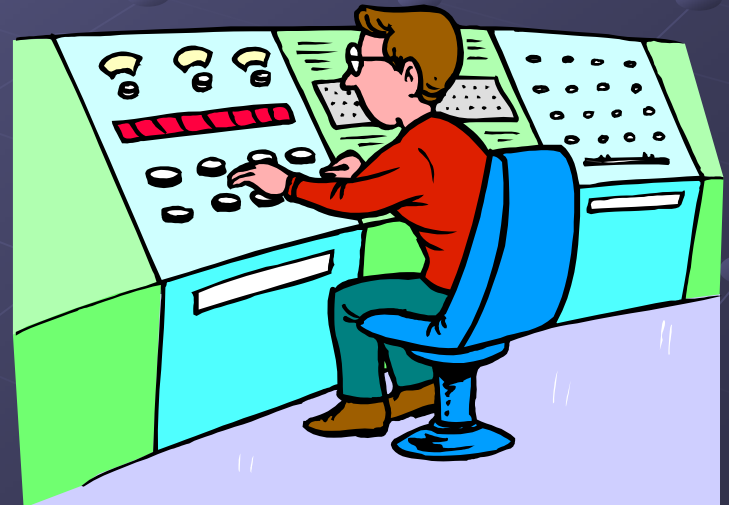
Copyright © 2003, Andy Tinkham & Cem Kaner

This research was partially supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

# What is Exploratory Testing?

## ● Current working definition (from Bach)

- “Any testing to the extent that the tester actively controls the design of the tests as those tests are performed and uses information gained while testing to design new and better tests”

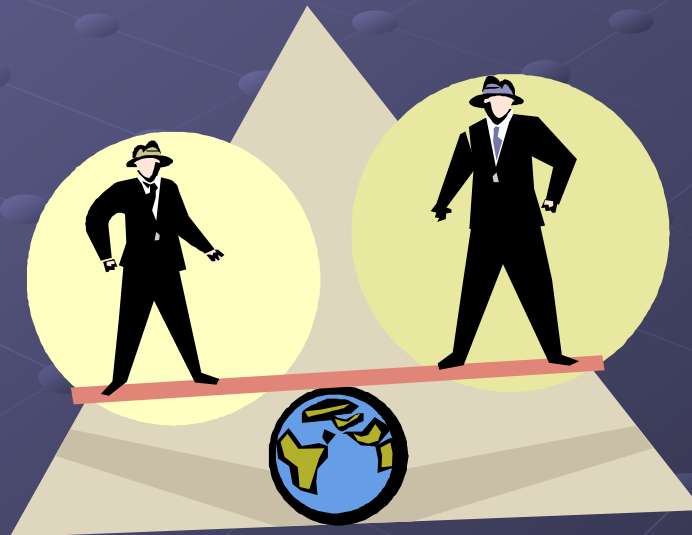


Copyright © 2003, Andy Tinkham & Cem Kaner

This research was partially supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

# Continuum of testing

- Purely scripted on one side
- Purely exploratory on the other
- Most testing falls somewhere in the middle



Copyright © 2003, Andy Tinkham & Cem Kaner

This research was partially supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).



# Good exploratory testing skills

- Able to craft tests that systematically explore the product
- Ability to analyze product
- Evaluate risk
- Use tools
- Think critically
- More careful observation
- Distinguish observation from inference
- Able to review and explain logic
- Look for errors in their own thinking
- Produce more and better ideas
- Group brainstorming
- Maintain deep inventory
- Ability to pose useful questions
- Describe what you see
- Recognize and manage bias
- Form and test conjectures

Copyright © 2003, Andy Tinkham & Cem Kaner

This research was partially supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

# What do Exploratory Testers Know?

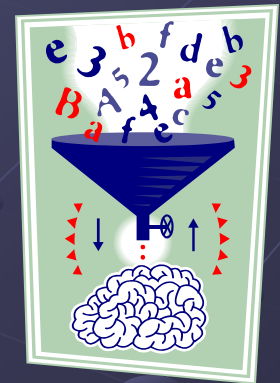
- All exploratory testing based on knowledge
- Some of this knowledge is gained during the testing
- Other knowledge is already in the tester's possession





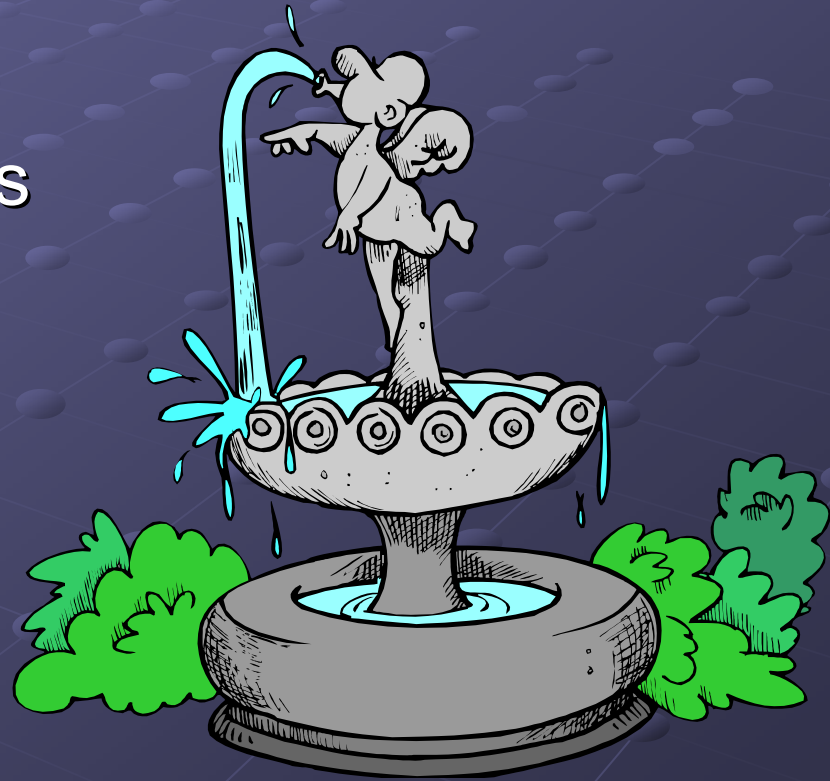
# Examples of Tester Knowledge

- This application (or this type of application)
- The platform the application will run on (or this type of platform)
- Risks associated with this platform or application type
- How programs fail in general
- Specific test techniques and tools



# Sources of Tester Knowledge

- Experiential
- Training
- Published Bug Taxonomies
- Heuristic Models



Copyright © 2003, Andy Tinkham & Cem Kaner

This research was partially supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

# Other Differences Across Exploratory Testers

## ● “Styles” of exploration

- Subject matter experts → come up with scenarios that rely heavily on use of product
- Technique experts → know certain techniques and look for situations to use them in



Copyright © 2003, Andy Tinkham & Cem Kaner

This research was partially supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

# Questioning Strategies

- Testing as questioning the application
  - Each question answered correctly increases our confidence in the application
- Testing becomes problem of choosing appropriate questions to get the best information we can
- Need ways to determine what questions to ask



# Context-free Questions

● Questions designed to focus thinking on a problem by examining it from multiple angles

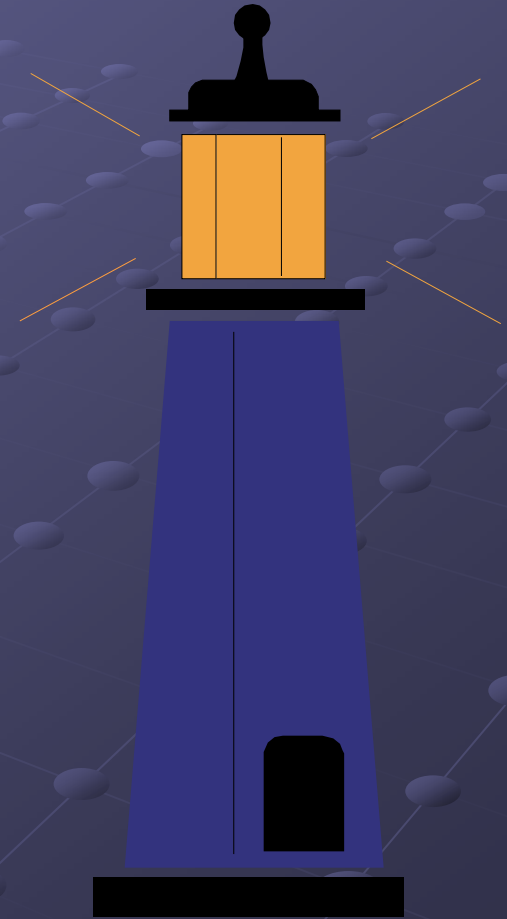
- Clear understanding of current problem
- Related problems/experience
- How to approach finding the best solution

● Examples

- CIA Phoenix Checklist
- Gauss & Weinberg, *Exploring Requirements*

# Heuristics

- Guides used to help people make decisions
- Not applicable in all situations
- Not correct in all situations



Copyright © 2003, Andy Tinkham & Cem Kaner

This research was partially supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).



# Examples of Heuristics

## ● Bach's lists

- *Lessons Learned in Software Testing* (Test Plan Evaluation Heuristics)
- "General Functionality and Stability Test Procedure"

## ● General testing lore

## ● Risk lists

# Types of Heuristics

- Describe what testers actually do or *should* do
  - “Important problems fast”
- Point towards specific types of failures
  - “Consistent with comparable products”
- Illustrate specific bugs
  - “Cache servers cause problems with web-based shopping cart applications”

# Examples of Heuristic Use

## ● Live at presentation



Copyright © 2003, Andy Tinkham & Cem Kaner

This research was partially supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

# Training Explorers

- Just beginning research into this
- Similarities to training people in improvisational theater



Copyright © 2003, Andy Tinkham & Cem Kaner

This research was partially supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

# Summary

- Good exploratory testing involves determining the right questions to ask an application. Achieving this requires a number of skills and knowledge
- Testing activities are already exploratory. Awareness of exploratory techniques can help testers better perform their traditionally non-exploratory duties

# Summary II: The Continuance

- Risks serve as heuristics for testers to highlight additional tests that a tester of which she might not otherwise think
- Heuristics can point to things that a tester should do, general types of bugs the tester should look for, or specific bugs that might exist.
- Heuristics are fallible guides which may or may not be applicable (or even work) in a given situation



# Questions?



Copyright © 2003, Andy Tinkham & Cem Kaner

This research was partially supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

The words "THANK YOU" are rendered in a bold, 3D, light blue font. The letters are positioned on a black rectangular plane that is tilted at an angle. The background of the entire slide is a dark blue grid with small, raised circular nodes at the intersections.

[andy@tinkham.org](mailto:andy@tinkham.org)

Copyright © 2003, Andy Tinkham & Cem Kaner

This research was partially supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).