



Exploratory Testing

Lecture Objectives:

- 1) Define Exploratory testing. ✓
- 2) Define a testing charter. ✓
- 3) Define timebox. ✓
- 4) Critique the advantages and disadvantages of exploratory testing. ✓
- 5) Compare and Contrast Scripted testing with exploratory testing, highlighting the differences of the approaches. ✓

A real world example

- A hex to binary converter...
 - This program converts a hex string into a binary number.

What is exploratory testing?

"Exploratory testing involves simultaneously learning, planning, running tests, and reporting / troubleshooting results."

Dr. Cem Kaner (2001)

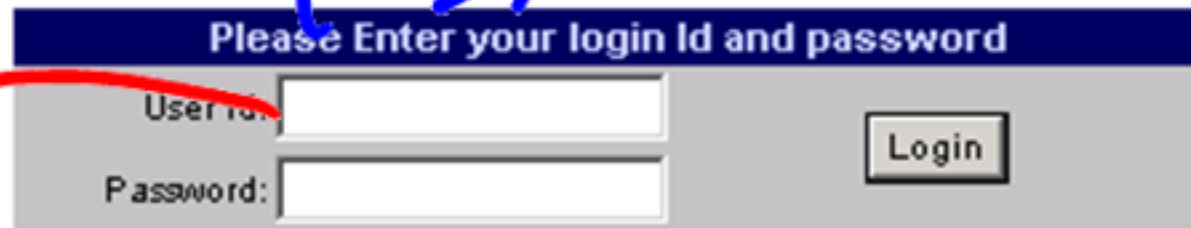
"Exploratory testing is an interactive process of concurrent product exploration, test design and test execution."

" To the extent that the next test we do is influenced by the result of the last test we did, we are doing exploratory testing."

James Bach, Satisfice (2001)

What is Scripted Testing?

- Small (but realistic) example:



A screenshot of a login form with a dark blue header containing the text "Please Enter your login Id and password". Below the header are two input fields: "User Id:" and "Password:". To the right of the "Password:" field is a "Login" button. Above the "User Id:" field, there is a handwritten blue symbol resembling a stylized 'S' or '8'. Above the "Password:" field, there is a handwritten blue symbol resembling a star or asterisk.

- How to script and test this login?
(Functional tests only – not security!)

→ Smith ;' *//

Sample test scripts

- Sample test script 1:
 - Launch the Login screen
 - Enter User-id: “xyz”
 - Enter Password: “zyx”
 - Press <Enter>
 - Expected result: login ok
- Sample test script 2:
 - Launch the Login screen
 - Enter User-id: “xyz”
 - Enter Password: “zyx”
 - Click the “Login” button
 - Expected result: login ok
- Sample test script 3:
 - Launch the Login screen
 - Enter User-id: “”
 - Enter Password: “zyx”
 - Press <Enter>
 - Expected: login rejected
- Sample test script 4:
 - Launch the Login screen
 - Enter User-id: “”
 - Enter Password: “zyx”
 - Click the “Login” button
 - Expected: login rejected

Sample Generic Scripts

- Sample generic test script 1:
 - Launch the Login screen
 - Enter valid User-id
 - Enter valid Password
 - Press <Enter> or click button
 - Expected result: login ok
- Sample generic test script 2:
 - Launch the Login screen
 - Enter invalid User-id
 - Enter valid Password
 - Press <Enter> or click button
 - Expected result: login rejected

Sample test “Pattern”

- Input fields:
 - Valid data
 - Invalid data
 - Length $>$ max
 - Length = max +1
 - Length = max
 - Length = max -1
 - Combinations of above
 - ...
- Actions:
 - Keyboard
 - Buttons
 - ...
- Operations:
 - Add, Modify, Inquiry, Delete
 - What to test for each...
 - ...

What is the goal of exploratory testing?

- Probe for weaknesses within a software system
- Learn about the construction of a software system

When to use Exploratory Testing? (1)

- A common goal of exploration is to *probe* for weak *areas* of the program.
- Test team's resource consumption per week:
 - 25% of the group's time developing new tests
 - 50% executing old tests (including bug regression)
 - 25% on exploratory testing

What do we want to do with exploratory testing?

- Create a mental model of the proper functioning of the system
- Design one or more tests that would disprove that conjecture
- Execute those tests and observe the outcomes
- Evaluate the outcomes against the conjectures
- Repeat the process until the conjecture is proved or disapproved

How do we do this?

- Timebox

Easier to schedule
short bursts.

- Uninterrupted block of time devoted to testing

Typically 60-120 minutes

*Long enough for solid testing
but not too long to allow
mind to wander.*

- One time box is devoted to a single test charter

A Charter

- Defines the mission for testing session
 - Different sessions may have different charters

what to test

what type of defects to look for

what tactics

what risks

etc.

Possible charters

- Thoroughly investigate a specific systems functionality
- Understand the performance characteristics of the software
- Ensure that all fields are properly validated
- Force all error conditions to verify each error message
- Check the design against user interface standards

Advantages

Advantages

- Beneficial when we can not determine the next test to run in advance ✓
- Can provide rapid fire feedback in short notice ✓
- Useful to refine the scope, size, and variations of existing defects
- Useful when scripted testing is no longer finding defects

Disadvantages

Disadvantages

- No ability to prevent defects ✓
 - Scripted testing starts earlier and may catch defects sooner
- If tests can be defined in advance, no need to explore
- No traceability
 - Very tough to prove that specific testing occurred
- Difficult to repeat on multiple releases
 - May lead to missed defects

What's the right answer?

- Both approaches have a place.
 - Neither is complete right
 - Neither is complete wrong

