



CUI Abbottabad

Department of Computer Science

SOFTWARE TESTING

Lecture 18

Exploratory Testing

EXPLORATORY TESTING

▶ **Inventors:**

Cem Kaner , James Bach (1990 s)

▶ **Definition:**

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

▶ **Elements / Variants**

- Charter: defines mission (and sometimes tactics to use)

Example: Check UI against Windows interface standards

- Session based test management:

Defects + Notes + Interviews of the testers

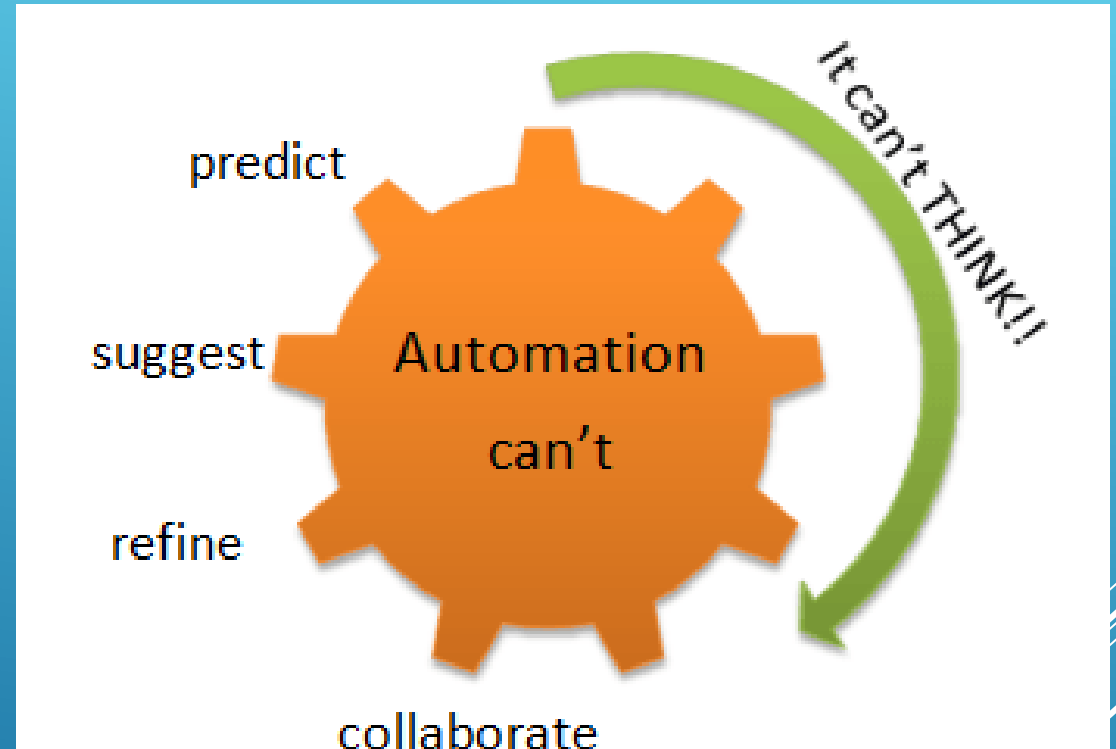
EXPLORATORY TESTING

- ▶ It is a type of software testing where Test cases are not created in advance, but testers check system on the fly. They may note down ideas about what to test before test execution. The focus of exploratory testing is more on testing as a "thinking" activity.
- ▶ Exploratory Testing is widely used in Agile models and is all about discovery, investigation, and learning. It emphasizes personal freedom and responsibility of the individual tester.
- ▶ Exploratory testing is a hands-on approach in which testers are involved in **minimum planning** and maximum test execution.

EXPLORATORY TESTING

- ▶ The test design and test execution activities are performed in parallel typically without formally documenting the test conditions, test cases or test scripts.
- ▶ This does not mean that other, more formal testing techniques will not be used.
 - ▶ For example, the tester may decide to use boundary value analysis but will think through and test the most important boundary values without necessarily writing them down. Some notes will be written during the exploratory-testing session, so that a report can be produced afterwards.
- ▶ Scripted Test Execution is usually a non-thinking activity where testers execute the test steps and compare the actual results with expected results. Such test execution activity can be automated does not require many cognitive skills.

Though the current trend in software testing is to push for automation, exploratory testing is a new way of thinking. Automation has its limits.



EXPLORATORY TESTING

- Is not random testing but it is ad-hoc testing with a purpose of find bugs
- Is structured and rigorous
- Is cognitively (thinking) structured as compared to the procedural structure of scripted testing. This structure comes from Charter, time boxing etc.
- Is highly teachable and manageable
- It is not a technique but it is an approach. What actions you perform next is governed by what you are doing currently

HOW TO DO EXPLORATORY TESTING

One type of exploratory testing --also called session-based test management (SBTM Cycle) –has the following 5 stages:

1. Create a Bug Taxonomy (classification)

- I. Categorize common types of faults found in the past projects
- II. Analyze the root cause analysis of the problems or faults
- III. Find the risks and develop ideas to test the application.

HOW TO DO EXPLORATORY TESTING

2. Test Charter

- Test Charter should suggest
 - what to test
 - how it can be tested
 - What needs to be looked
- Test ideas are the starting point of exploration testing
- Test charter helps determine how the end user could use the system

HOW TO DO EXPLORATORY TESTING

3. Time Box

- This method includes a pair of testers working together not less than 90 minutes
- There should not be any interruption in those 90 minutes session
- Timebox can be extended or reduced by 45 minutes
- This session encourages testers to react on the response from the system and prepare for the correct outcome

4. Review Results:

- Evaluation of the defects
- Learning from the testing
- Analysis of coverage areas (function, code)

5. Debriefing:

- Compilation of the output results
- Compare the results with the charter
- Check whether any additional testing is needed

WHEN TO DO EXPLORATORY TESTING

- In the **early stages of SDLC** – software development life cycle when the code undergoes rapid changes, exploratory testing can be highly effective.
- The developers can use this technique to perform unit tests while the testers can acquire familiarity with the application using this testing approach.
- The experience gained from exploratory testing can be **valuable in preparing test scripts** and doing additional testing in the later stages of the software development life cycle.
- In the **agile development environment**, there are short scrum cycles and little time available for developing formal test design and scripts. Exploratory testing is well suited for agile environment as it keeps up with the short scrum cycles.
- While performing exploratory testing, test plans are developed on the fly which saves lot of time for the tester. At the end of each scrum cycle, critical exploratory tests can be captured for subsequent scrums.

CATEGORIES OF EXPLORATORY TESTING

1. Freestyle exploratory testing

In freestyle exploratory testing, application is tested in adhoc way and not many guidelines or procedures are set for testing. But exploratory testing can be useful in the following situations:

- The tester needs to get familiarized with the application rapidly
- The tester has to validate other tester's work
- A defect is required to be examined by the tester

CATEGORIES OF EXPLORATORY TESTING

2. Scenario based exploratory testing

In Scenario based exploratory testing, testing is performed based on scenarios. Scenarios could be provided by the customer or could be prepared by the test team. After preliminary testing has been performed, testers may innovate testing based on gained knowledge and skill.

CATEGORIES OF EXPLORATORY TESTING

3. Strategy based exploratory testing

In strategy based exploratory testing, common testing techniques like Decision Table based testing, Cause-Effect graphing, and Error Guessing are joined with the exploratory testing. The suitable tester for this type of testing would be someone who has good knowledge about the application.

ADVANTAGES

- ▶ Less preparation is needed and important bugs are found quickly
- ▶ Useful when requirement documents are not available or only partially available
- ▶ Involves Investigation process which helps find more bugs than normal testing
- ▶ Helps to expand the imagination of testers by executing more and more test cases which finally improves productivity as well
- ▶ This testing drills down into details of the application and covers (potentially) all the requirements
- ▶ This testing covers all the types of testing and it covers various scenarios and cases
- ▶ Encourages creativity and intuition
- ▶ Generation of new ideas during test execution

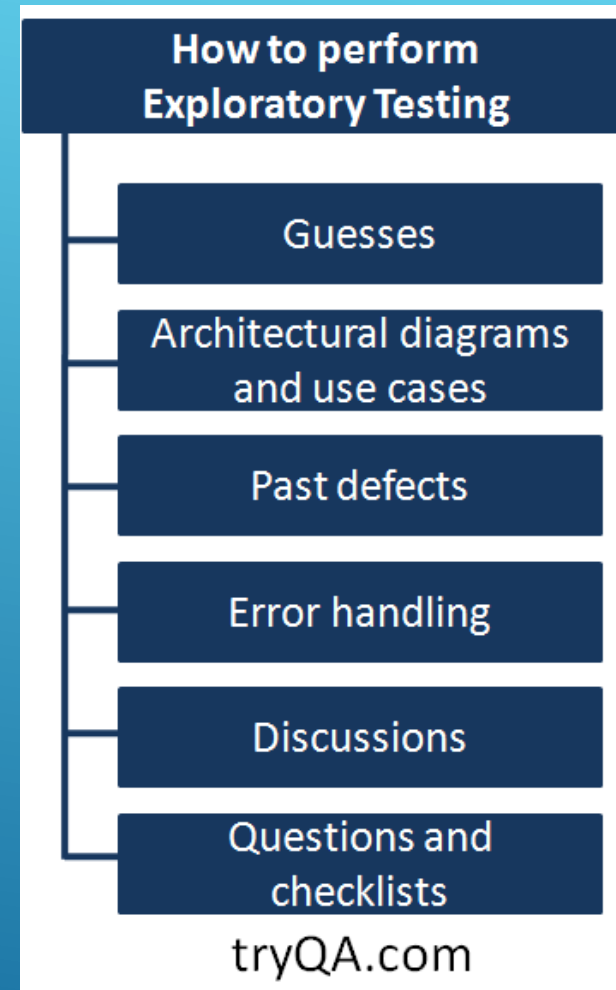
DISADVANTAGES

- ▶ As the tests are invented and performed randomly, they can't be reviewed in advance, and it can be difficult to show which tests have to be run.
- ▶ Testing is dependent on tester's knowledge, skill and experience.
- ▶ Gaining familiarity with an application takes time, so there is a possibility of defects being missed if the tester has less knowledge of the website or application
- ▶ Limited by domain knowledge of the tester
- ▶ Not suitable for Long execution time (e.g., scientific programs)

HOW TO DO EXPLORATORY TESTING

Let us assume you have been asked to **perform exploratory testing on a Hospital management system.**

As shown in the figure below, there are several **ways to perform exploratory testing.**



1. GUESSES

- ▶ Guesses are used to find part of the program that is likely to have more errors. Previous experience of working on a similar product / software / technology helps in guessing.
- ▶ In case of the Hospital management system, you could guess that the payment module can have higher errors since it has to integrate with the payment gateway, transactions could time out and lead to errors if not handled correctly.

2.ARCHITECTURAL DIAGRAMS AND USE CASES

- ▶ Architectural diagrams describe the interactions and relationships between different components and modules. Use cases give an understanding of the product's usage from the end user's perspective.
- ▶ Exploration technique may use these diagrams and use cases to test the product.
- ▶ Hospital Administration System – You remember that there was a use case where more than one person could share the same phone number and the application should accept it, you decide to test that scenario.

3.PAST DEFECTS

- ▶ Studying the defects reported in the previous releases helps in understanding those features of the software that are expected to have maximum defects.
- ▶ In the past, reporting module of the hospital management system used to consume a lot of memory and have several errors so you decide to test it as well.

4. ERROR HANDLING

- ▶ Error handling is that part of the code which takes appropriate actions in case of any failure. Exploratory testing could be done using different scenarios to test graceful error handling.
- ▶ In the Hospital Management System, the reporting module would consume a lot of memory and crash sometimes. You decide to test it to ensure that reports that have been queued for generation are generated eventually even if there are any issues encountered in between.

5. Discussions

- ▶ Exploratory testing could also be planned based on the understanding of the software during project discussions and meetings.

6. Questions and checklists

- ▶ Questions like “what, when, how, who and why” can provide clues for exploratory testing of the software.

EXPLORATORY TESTING IN AGILE

- ▶ In Agile methodology there are short sprints usually of about one month and the deadlines are extremely tight. Exploratory testing is very useful in agile because of tight deadlines and focus on speedy results. Once the requirements are understood by the tester, he can perform testing based on his/her skill and intuition.
- ▶ When the tester becomes more familiar with the application's features and behaviour, he can create additional test cases for further testing and could find more bugs.
- ▶ As exploratory testing in agile is free of rigid processes and documentation requirements, the tester does not have to maintain documents for everything but it would be good if a short report on items tested, bugs found etc. is maintained for the purpose of future reference.

EXPLORATORY TESTING - TIPS

During exploratory testing, do the following:

- Mission of testing should be very clear
- Keep notes on what needs to be tested, why it needs to be tested and the assessment of the product quality
- Tracking of questions and issues raised during exploratory testing
- Better to pair up the testers for effective testing
- The more we test, more likely to execute right test cases for the required scenarios

EXPLORATORY TESTING - TIPS

It is very important to document and monitor the following:

- Test Coverage- Whether we have taken notes on the coverage of test cases and improve the quality of the software
- Risks- Which risks needs to be covered and which are all important ones?
- Test Execution Log- Recordings on the test execution
- Issues / Queries- Take notes on the question and issues on the system