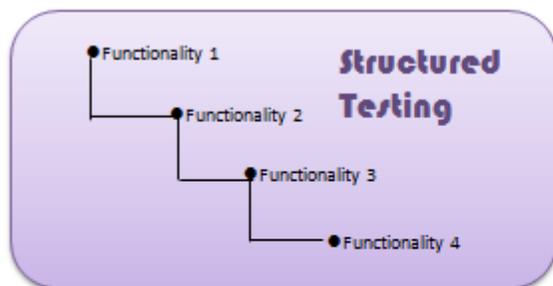


Let's explore - Exploratory Testing!

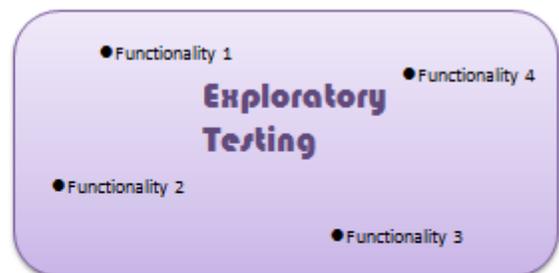
Under scripted testing, you design test cases first and later proceed with test execution. On the contrary, exploratory testing is simultaneous process of test design and test execution all done at the same time.

Scripted Test Execution is usually a non-thinking activity where testers executes the test steps and compare the actual results with expected results. Such test execution activity can be automated does not require much cognitive skills.

Exploratory testing, on the other hand, is all about discovery, investigation and learning. This emphasizes on personal freedom and responsibility of the individual tester. 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.

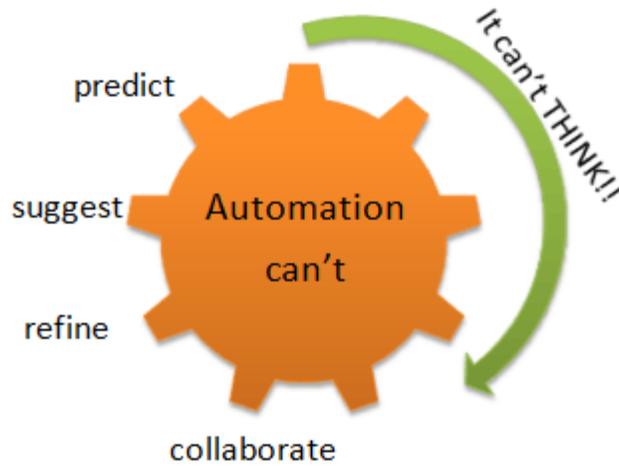


**Functionalities are checked
in a structured manner**



**Functionalities are checked
in a ad-hoc manner**

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



Here are some major differences between Scripted and exploratory testing:

Scripted Testing	Exploratory Testing
Directed from requirements	Directed from requirements and exploring during testing
Determination of test cases well in advance	Determination of test cases during testing
Confirmation of testing with the requirements	Investigation of system or application
Emphasizes on prediction and decision making	Emphasizes on adaptability and learning
Involves confirmed testing	Involves Investigation
Is about Controlling tests	Is about Improvement of test design
Like making a speech - you read from a draft	Like making a conversion - its spontaneous

Exploratory testing -

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

Exploratory Test Preparation:

Exploratory test preparation goes through following 5 stages detailed below and it is also called session based [test management](#) (SBTM Cycle):

1. Create a Bug Taxonomy (classification)
 - Categorize common types of faults found in the past projects
 - Analyze the root cause analysis of the problems or faults
 - Find the risks and develop ideas to test the application.
2. Test Charter
 - Test Charter should suggest
 1. what to test
 2. how it can be tested
 3. 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
3. Time Box
 - This method includes pair of testers working together not less than 90 minutes
 - There should not be any interrupted time in those 90 minutes session
 - Time box 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

5. Debriefing:

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

During **exploratory execution**, following needs to be done:

- 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

It is very important to take 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

Smarter exploratory testing finds more errors in less time.

Pros and Cons of Exploratory Testing:



Pro	Con
<p>This testing is useful when requirement documents are not available or partially available It involves Investigation process which helps find more bugs than normal testing- Uncover bugs which are normally ignored by other testing techniques Helps to expand the imagination of testers by executing more and more test cases which finally improves productivity as well This testing drill down to smallest part of application and covers all the requirements</p>	<ul style="list-style-type: none"> • This testing purely depends on the tester skills • Limited by domain knowledge of the tester • Not suitable for Long execution time

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	
--	--

Challenges of Exploratory Testing:

There are many challenges of exploratory testing and those are explained below:



- Learning to use the application or software system is a challenge
- Replication of failure is difficult
- Determining whether tools need to be used can be challenging
- Determine the best test cases to execute can be difficult
- Reporting of the test results is challenge as report doesn't have planned scripts or cases to compare with the actual result or outcome
- Documentation of all events during execution is difficult to record
- Don't know when to stop the testing as exploratory testing has definite test cases to execute.

When use exploratory testing?

Exploratory testing can be used extensively when

- The testing team has experienced testers
- Early iteration is required
- There is a critical application
- New testers entered into the team

Conclusion:

Exploratory testing is performed to overcome the limitations of scripted testing. It helps in improving test case suite. It empathizes on learning and adaptability.