

# Exploratory Testing

## What is Exploratory Testing?

The term 'Exploratory' means 'travel through' or 'inquire in to' or discuss a subject into detail. Exploratory testing in software testing means the same. Where in a tester is exploring the software or going through its functionality. What is important here is testers applying their own experience, knowledge, or thinking about the application with user perception.

## Where it is Applied

This is the most adopted testing in Agile development, as in the Agile methodology testing team gets release very frequently, and with every release, it is expected from a tester to give its feedback quickly.

An experienced tester quickly goes through the release and explore all the areas, note down the functional flaws, and uncover the bugs which are not possible in a traditional approach where tester go through pre-formatted test cases.

## Advantages of Exploratory Testing

- It is a quick and time-saving method of testing; it helps to take real time decisions.
- It helps the tester to understand the application when there is no enough document present.
- It is a more practical approach where a tester thinks with a user perspective.
- It is a human approach testing, which cannot be automated where real user intelligence and experience is used, it does not work on a pass or fail, it actually thinks about User-friendliness, ease of use or usability aspect of the application.

# Disadvantages of Exploratory Testing.

- Sometimes it is difficult to reproduce the defect because the testing is unstructured actual steps are missed and cannot be written in the bug reproduction steps.
- It should not be done by a fresher or who has no prior experience in the domain because he won't be able to think with the user perspective and will always log unrealistic bugs which will kill the time of the entire team.
- Automation is not possible like few things are totally dependent on end user's own perspective, intelligence, or mindset...and these humans depended things cannot be automated ever.

## A practical example of Exploratory testing

Let's take an example of exploratory testing,

A Web development company [ecommerce] has to release its last build in one and a half months, it has a team of 3 developers and a tester who is a fresher. No much documents are present only a few user stories are in the system and the team has to complete these user stories on time. The test cycle is about to begin in one day and they have to finish testing in 7 days.

An experienced test engineer gets hired in the team since he has domain experience, he won't spend much time on document and stories reading, he will directly go to the application and start looking at the core functionalities of the application.

He can adopt the following ways of doing exploratory testing.

- **Freestyle or Adhoc Testing:** he can randomly click on any link, or button or any operations to see the response of the application, he can note down the bug if application breaks somewhere.
- **Scenario-based testing:** based on his experience he can think of multiple scenarios like search a product and add it into the cart and proceed to checkout, he can observe the behavior if he is able to execute each scenario successfully or not.

- **User's roles-based testing:** he can explore the application by using different user roles of the application, and for every step, he can check if the particular user is allowed to perform certain actions or not.
- **Structured Exploratory testing:** he can write down high-level test steps, for example, Login, search, add to cart, checkout, and payment. Now he can explore each area in detail. And write down the bugs he discovered while testing.

-----END-----

Muntasir Abdullah Mizan  
muntasir.abdullah01@gmail.com