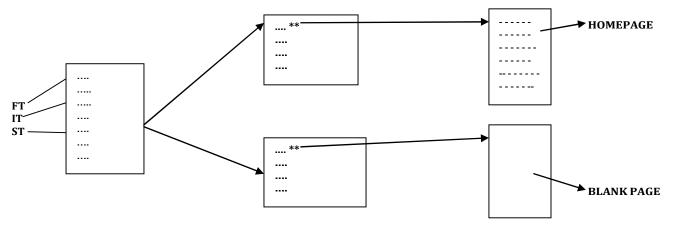
<u>AD - HOC Testing</u> (also called <u>Monkey Testing</u> / <u>Gorilla Testing</u>) Testing the application randomly is called Ad-hoc testing.

Why we do Ad-hoc testing?

1) End-users use the application randomly and he may see a defect, but professional TE uses the application systematically so he may not find the same defect. In order to avoid this scenario, TE should go and then test the application randomly (i.e, behave like and end-user and test).

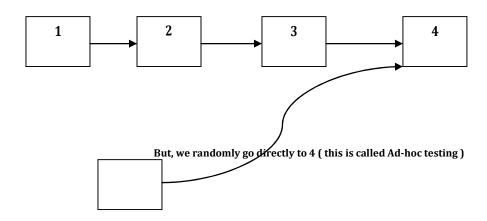
For ex,



In the above figure, after we have tested the application for FT, IT and ST – if we click on some feature instead of going to homepage (or) sometimes datapage, if it goes to blank page then it will be a bug. In order to avoid these kind of scenarios, we do Ad-hoc testing.

- 2) Development team looks at the requirements and build the product. Testing Team also look at the requirements and do the testing. By this method, Testing Team may not catch many bugs. They think everything works fine. In order to avoid this, we do random testing behaving like end-users.
- 3) Ad-hoc is a testing where we don't follow the requirements (we just randomly check the application). Since we don't follow requirements, we don't write test cases.

Requirement says to test like below (1 -> 2 -> 3 -> 4)



Examples of Ad-Hoc testing for Gmail:

- 1) Login to Gmail using valid username and password. Logout from Gmail. Click on Back button. It should not go back to Inbox page. If it does, then it is a javascript error and it is a bug. It should go back to Login page and say session expired.
- 2) Login to Gmail homepage using valid username and password. Once we are in Inbox page, copy the URL of the inbox which is in the address bar of the homepage and paste it in Notepad file. Logout from Gmail. Now, open browser page and paste the URL of the inbox in the address bar. It should not go to the inbox, instead it must go to the welcome page of Gmail.
- 3) Login into Gmail. Go to Settings and Change Password. Set the old password only as the new password and see what happens.

Let us consider an example of Fraud Management System of Online Banking Application.

BLOCK ACCOUNT		
	_	
	ACCOUNT NUMBER	
	DI OCK	CANCEL
	BLOCK	CANCEL
BLOCK ACCOUNT		
Confirmation Message should be displayed		
ACCOUNT IS BLOCKED		

When we click on the **Block Account link**, we are transferred to the Block Account page where we find several features in that. Enter the data and click on Block Account, then that account has to be blocked. Now, we will see how we can do **Ad-hoc testing on this application.**

- 1) Login as Bank Manager and enter the Account Number and click Block and see whether it is blocked or not.
- 2) Before blocking the Account, Go and Delete the person whose account is to be blocked and again Login and check whether it is blocked or not. As we click the Block it should throw a message saying customer not available as an error message. Here, we randomly check the application and nothing is mentioned in the requirements. Thus here we do Ad-hoc testing.
- 3) Suppose some User B transfers money to A whose account is blocked. In this case also, we should get a message saying Account(of A) is blocked [i.e, by the time B transfers money, manager blocks A"s account]. Actually the requirement does not say check for money transfer from other account and do testing. But this testing is done by TE not against the requirement. Even sometimes without throwing message that Account

is blocked the money gets transferred. In this case also, the TE checks for it and thus it becomes Ad-hoc Testing.

NOTE :-

- Ad-hoc testing is basically negative testing because we are testing against requirements (out of requirements).
- Here, the objective is to somehow break the product.

When to do Ad-Hoc testing?

- Whenever we are free, we do Ad-hoc testing. i.e, developers develop the application and give it to testing team. Testing team is given 15days for doing FT. In that he spends 12 days doing FT and another 3days he does Ad-hoc testing. We must always do Ad-hoc testing in the last because we always 1st concentrate on customer satisfaction
- After testing as per requirements, then we start with ad-hoc testing
- When a good scenario comes, we can stop FT, IT, ST and try that scenario for Ad-hoc testing. But we should not spend more time doing Ad-hoc testing and immediately resume with formal testing.
- If there are more such scenarios, then we record it and do it at the last when we have time.