

MANUAL SOFTWARE TESTING

Gamita Patel

Who Should Test?



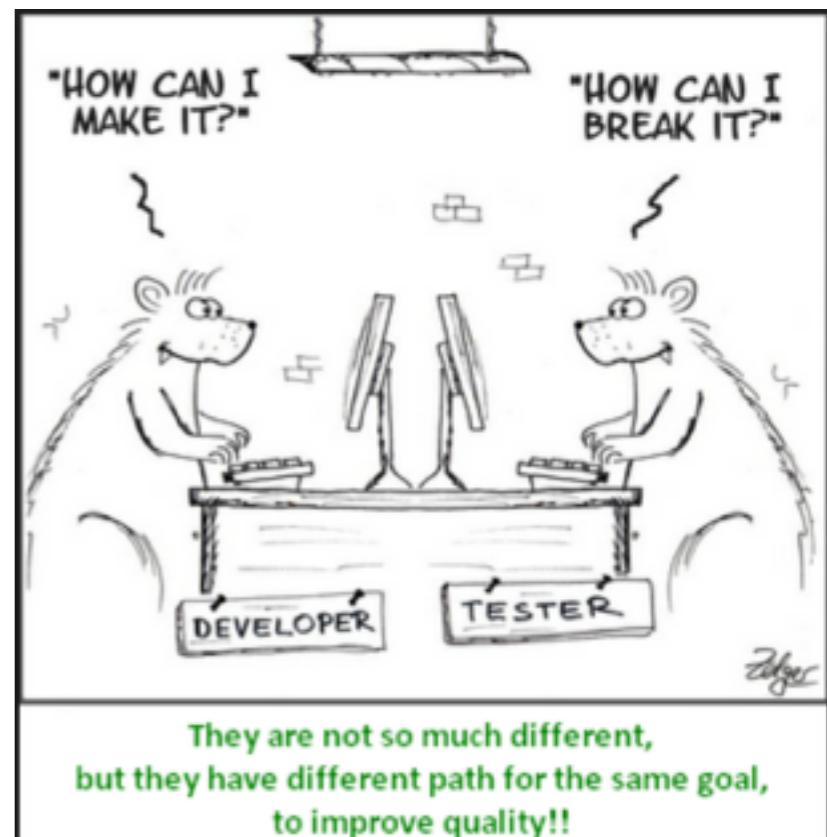
- Developer
 - Understands the system
 - But, will test gently
 - And, is driven by deadlines
- Independent tester
 - Must learn system
 - But, will attempt to break it
 - And, is driven by “quality”

1. Don't trust a developer



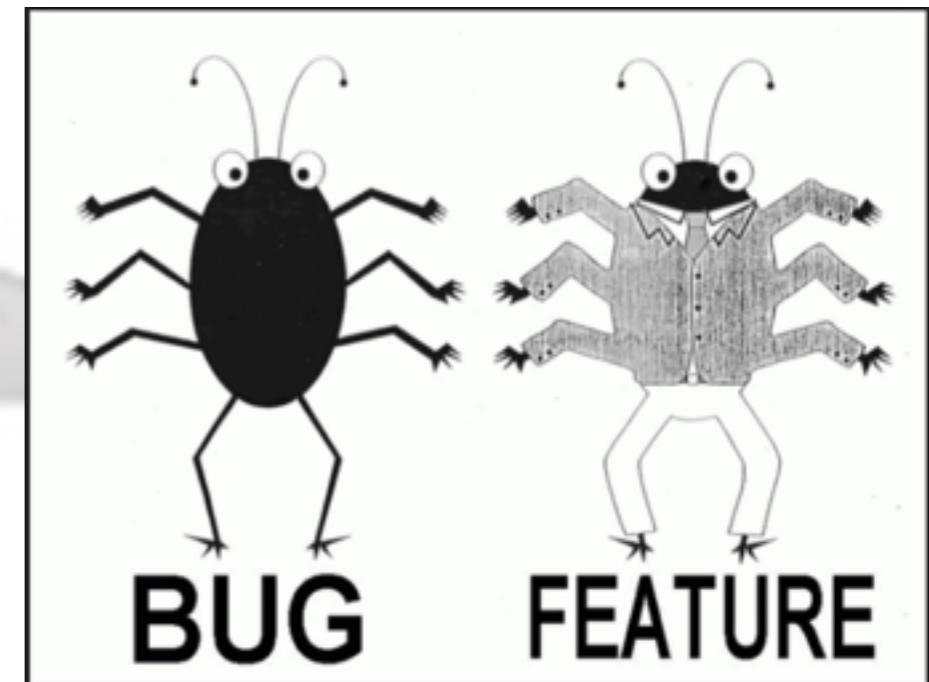
It is your responsibility to **doubt**. A sceptical approach to testing significantly improves the quality of your work.

Quickly take a butcher's, it's a minor change.
For sure nothing has broken



WHAT IS BUG?

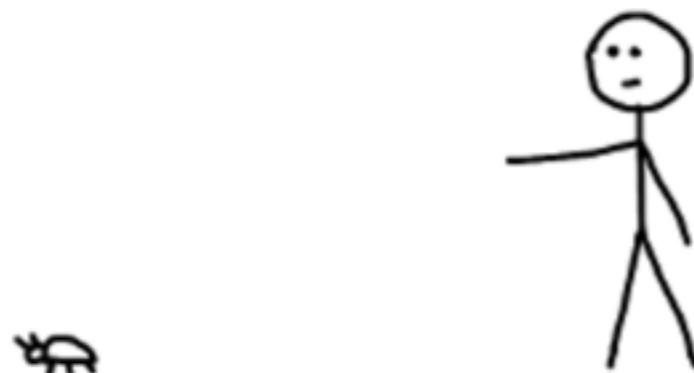
- Bug can be defined as abnormal behaviour of the software
- A programmer while designing and building the software can make mistakes or error/flaws, which are called defects.
- In Software testing, when the expected and actual behaviour is not matching, an incident needs to be raised. An incident may be a Bug.
- A bug can be absence of expected feature
- Why?
 - User error (Insufficient knowledge of that application)
 - Code defect(Developer error)
 - Test Environment error



WHAT TESTER NEEDS TO DO WHEN GENUINE BUG IS FOUND ?

How to raise bugs in 3 simple steps

1 SPOT



2 APPROACH



3 RAISE



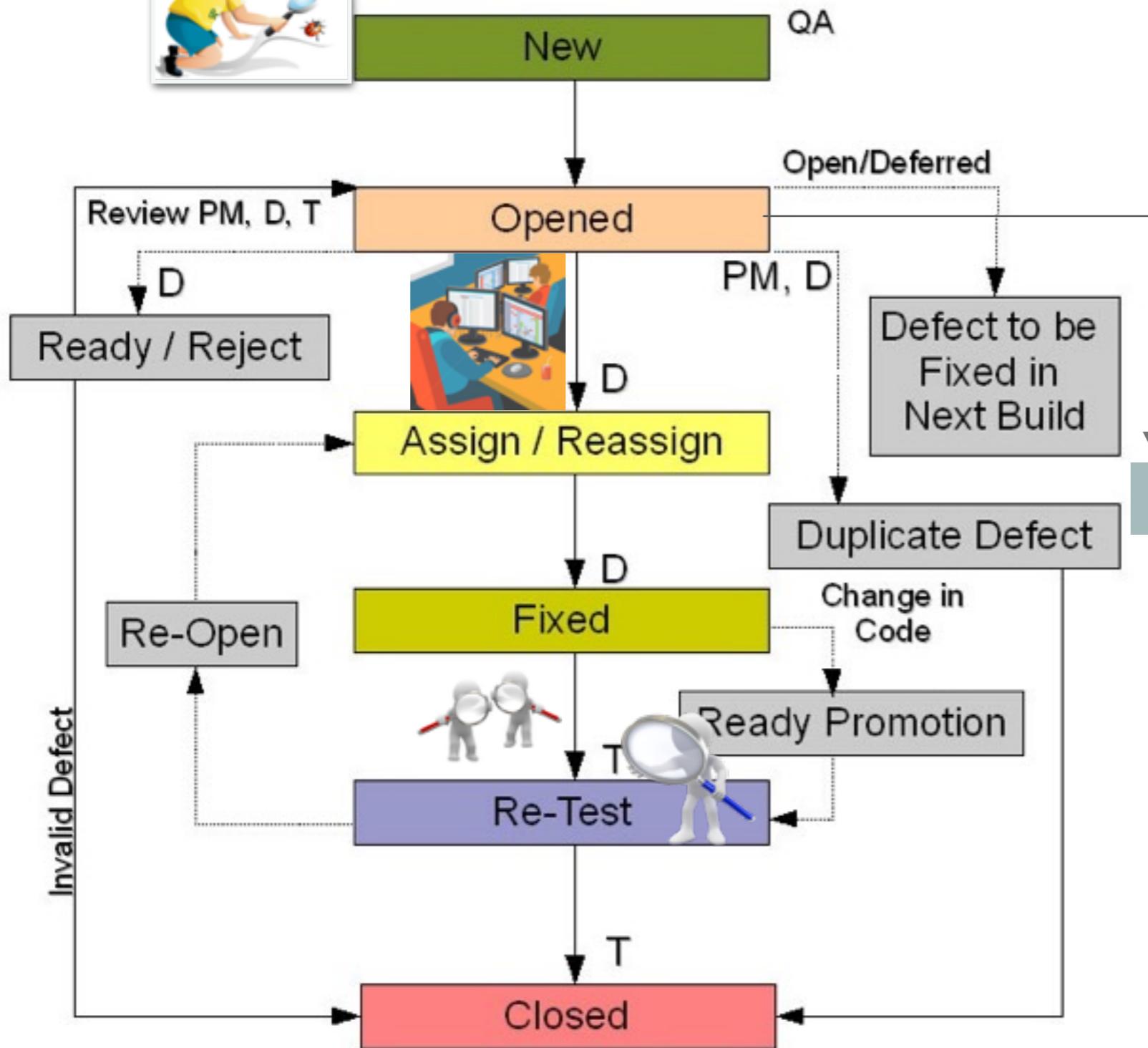
A badly reported bug is much
worse than a badly drawn one...



... At least I could claim



BUG Life Cycle



Defect/Bug life cycle is a cycle, which a defect goes through during its lifetime. It starts when defect is found and ends when a defect is closed, after ensuring it's not reproduced.

Can't Reproduce

Developer - Tester Conversation....

I am not able to replicate this issue. this is working fine on my machine. So close this bug!!

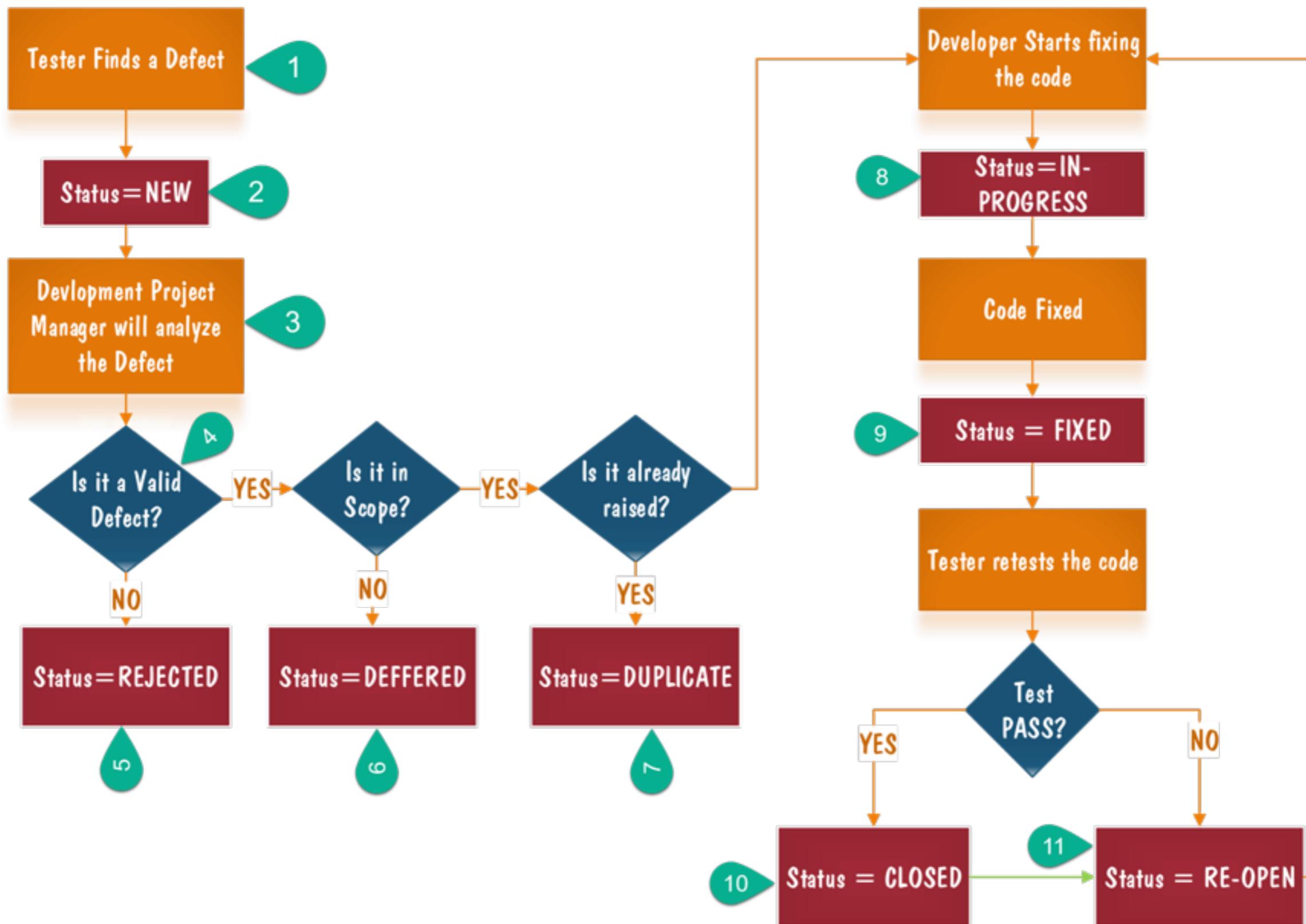


Developer

I don't care if it is working fine on your machine. We are not going to deliver your machine to the client.

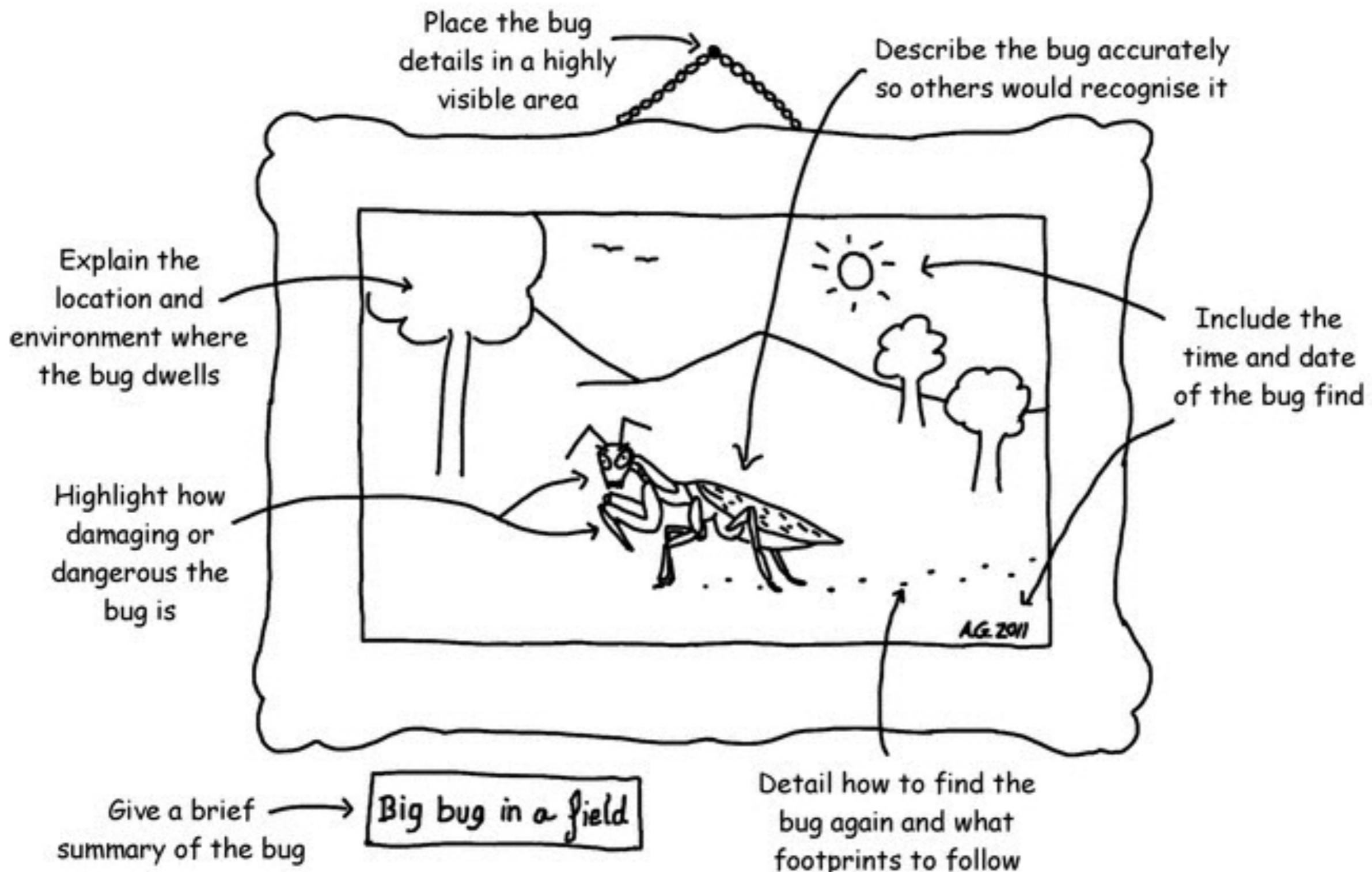


Tester



BUG REPORTING

The Art of Bug Reporting



BUGS HAVE FEELINGS TOO

IF YOU FIND A BUG:
REPORT IT

BUGS DON'T LIKE
TO BE FORGOTTEN



IF YOU FIND A BUG:
GET TO KNOW THEM

BUGS LIKE TO BE
UNDERSTOOD



This ladybird
has 3 spots

IF YOU FIND A BUG:
TAKE A PHOTO

BUGS LIKE TO KEEP MEMORIES
OF THE OCCASION



IF YOU FIND A BUG:
GET TO KNOW THEIR MATES

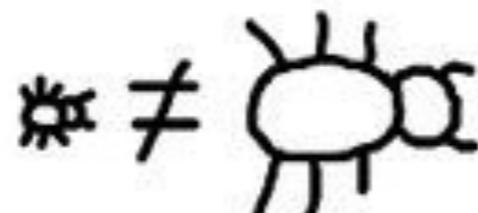
BUGS ARE SOCIALITES



IF YOU FIND A BUG:
REPORT IT QUICK
OTHERWISE BUGS SETTLE IN AND
MAKE A HOME FOR THEMSELVES



IF YOU FIND A BUG:
BE HONEST
BUGS DON'T LIKE
GOSSIP



IF YOU FIND A BUG:
NOTE HOW YOU
MEET THEM
BUGS ARE ROMANTICS



IF YOU FIND A BUG:
DON'T IGNORE IT
BUGS CAN BITE IF
NOT APPRECIATED



BUG/DEFECT REPORT TEMPLATE

- Defect ID – Every bug or defect has its unique identification number
- Defect Description – This includes the abstract of the issue.
- Product Version – This includes the product version of the application in which the defect is found.
- Detail Steps – This includes the detailed steps of the issue with the screenshots attached so that developers can recreate it.
- Date Raised – This includes the Date when the bug is reported
- Reported By – This includes the details of the tester who reported the bug like Name and ID
- Status – This field includes the Status of the defect like New, Assigned, Open, Retest, Verification, Closed, Failed, Deferred, etc.
- Fixed by – This field includes the details of the developer who fixed it like Name and ID
- Date Closed – This includes the Date when the bug is closed
- **Severity** – Based on the severity (Critical, Major or Minor) it tells us about impact of the defect or bug in the software application
- **Priority** – Based on the Priority set (High/Medium/Low) the order of fixing the defect can be made. (Know more about Severity and Priority)

Defect Severity vs. Priority

Priority

How quickly the bug needs to be fixed

Severity

What is the impact of the bug on the application/system

P1	• Critical	S1	• Critical
P2	• High	S2	• Major
P3	• Medium	S3	• Moderate
P4	• Low	S4	• Low

Defect Priority Levels

Defect Severity Levels

P1

- Critical

P2

- High

P3

- Medium

P4

- Low

Defect Priority Levels

S1

- Critical

S2

- Major

S3

- Moderate

S4

- Low

Defect Severity Levels

© www.SoftwareTestingHelp.com

Severity

Critical

Non - Critical

Fix 1st

High
Very Important functionality that does not work

Fix 2nd

Logo or Name of the Company is wrong

Feature that is rarely used is not working as expected

Fix 3rd

Caption on an image is in wrong colour

Fix 4th

Priority

High
Low

Priority

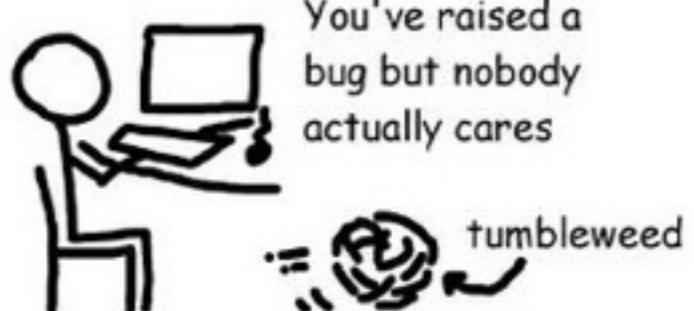
- Priority refers to the **Project** and **how urgent** it is to solve the Bug.
- Priority is set based on **changing project factors** eg. Status of the bug, its importance from customer side.
- Priority is a **dynamic field**, should be revised and updated as the project progresses.

Severity

- Severity refers to the **Bug** and **how it affects** the User's interaction with the Application
- Severity is **objectively** set based on the direct and indirect impact of the bug and its probability of occurrence.
- Severity is usually a **static** field.
(the only reason to modify it would be if we learn something new about the bug.)

DON'T KNOW WHAT SEVERITY RATING TO GIVE A BUG?
THEN DO NOT FEAR, THE CARTOON TESTER IS HERE

SEVERITY STATUS: MINOR



SEVERITY STATUS: SIGNIFICANT



SEVERITY STATUS: MAJOR

You've managed
to get the developer
off their seat

OK.. can you
re-do those
steps again...



SEVERITY STATUS: CRITICAL

You have 3
developers
interested!

And that only
happens when you
skip step D... Nice!



SEVERITY STATUS: CATASTROPHIC

Everyone is
crowding round
your desk!

Holy mackerel!! Who
wrote this code?!



THE LEVEL OF BUG INTEREST IS EQUIVALLY PROPORTIONAL TO THE SEVERITY

BUG REPORTING TOOLS

Atlassian



SEVEN PRINCIPLES OF SOFTWARE TESTING

1- Testing shows presence of defects

- No defects found ≠ No defects present

2- Exhaustive testing is not possible

- All scenarios of Software can't be tested
- Adopt smarter ways of testing

3- Early testing

- Save Time
- Save Money
- Make Customer Happy 😊



4- Defect clustering

- Defects are present in cluster

5- Pesticide Paradox

- Test cases should be update periodically
- Review of Test cases is important activity

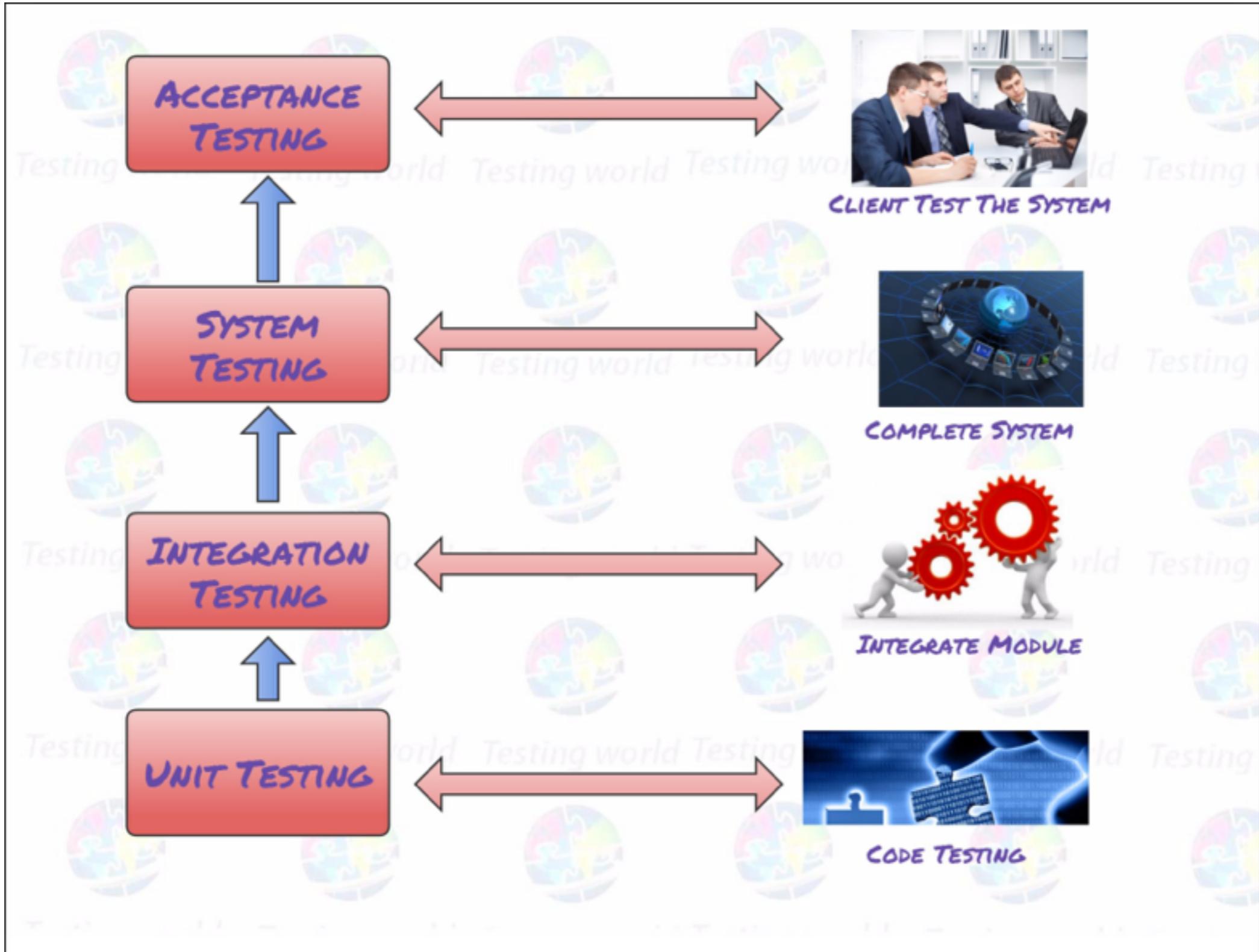
7- Absence of errors fallacy

- Test only stable software's

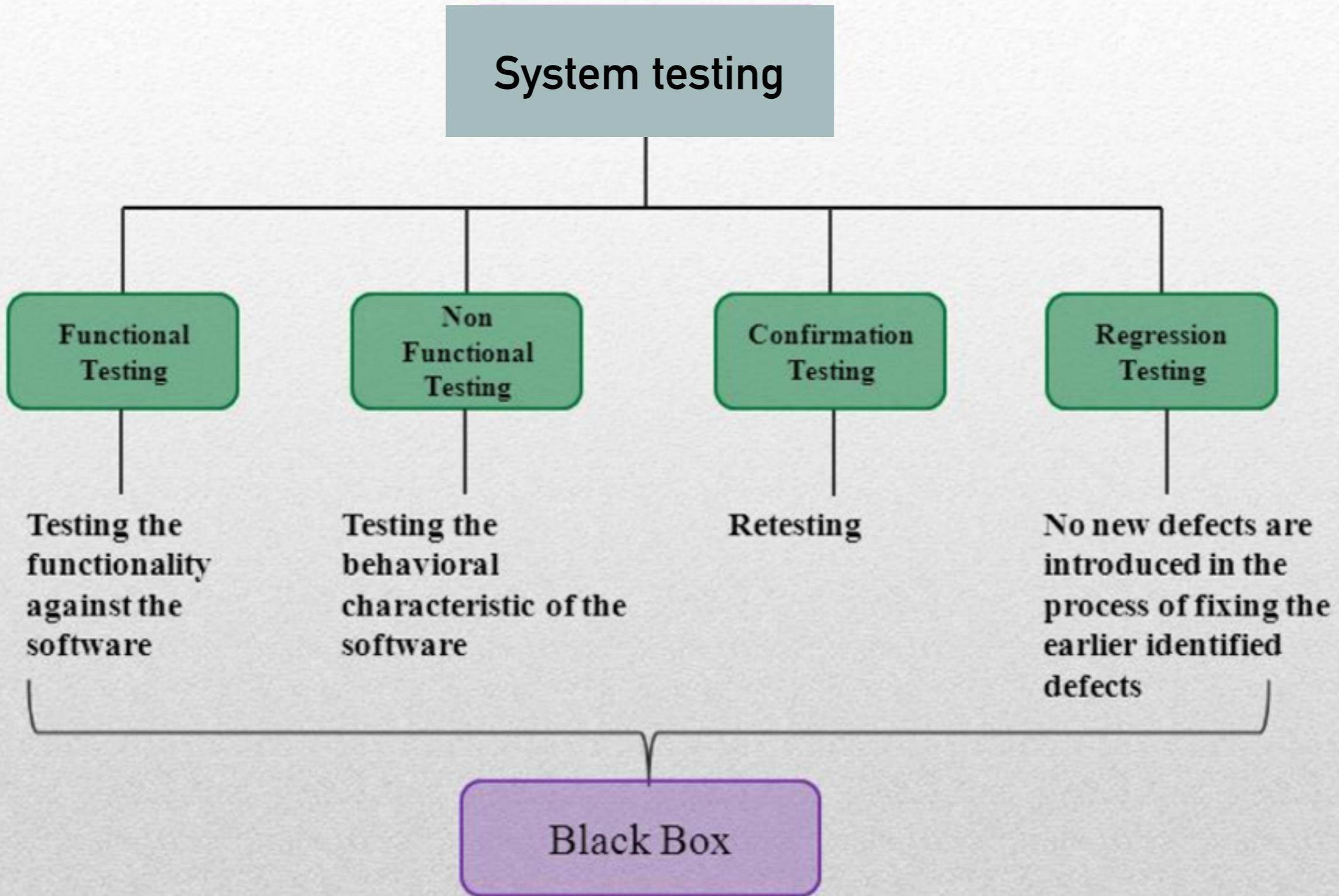
6- Testing is context dependent

- Testing method / strategy is different for different software's

TESTING LEVELS



Test Types



NON FUNCTIONAL TESTING

- Compatibility testing: Testing across
 - Cross Browser
 - Cross Operating system
 - Cross Device
- Performance Testing: How my application behaves in case of load??
 - Load: By applying consecutive load (increase and decrease) for short interval
 - Stress: By applying non-consecutive load for short interval
 - Soak: Check load for longer period of time. Check when system can break
- Security Testing: Penetration testing - esp. in financial, banking sector
- Database testing
- GUI Testing: (Graphical User Interface Testing)

USER ACCEPTANCE TESTING (UAT)

- User acceptance testing (UAT) is the last phase of the software testing process. During UAT, actual software users test the software to make sure it can handle required tasks in real-world scenarios, according to specifications.

