

Manual Testing

Class 3

Agenda

What is a Bug

Bug Life Cycle

What is a Bug?

Wikipedia Definition - A software bug is an error, flaw, failure or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways



What is a Bug?

- It is an unexpected result you find during testing when you are comparing expected result with actual result.
- Bug can also be a tester finding software difficult to understand, hard to use, or slow from an end-user point of view.
- A software bug is an error, flaw, failure or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways.

Bug Examples

Software Doesn't do something that specification says it should do.

Example - User was not able to login with valid credentials.

Software does something that specification says it should NOT do.

Example - User was able to login with invalid credentials.

Bug Examples

Software Does something that specification does not mention.

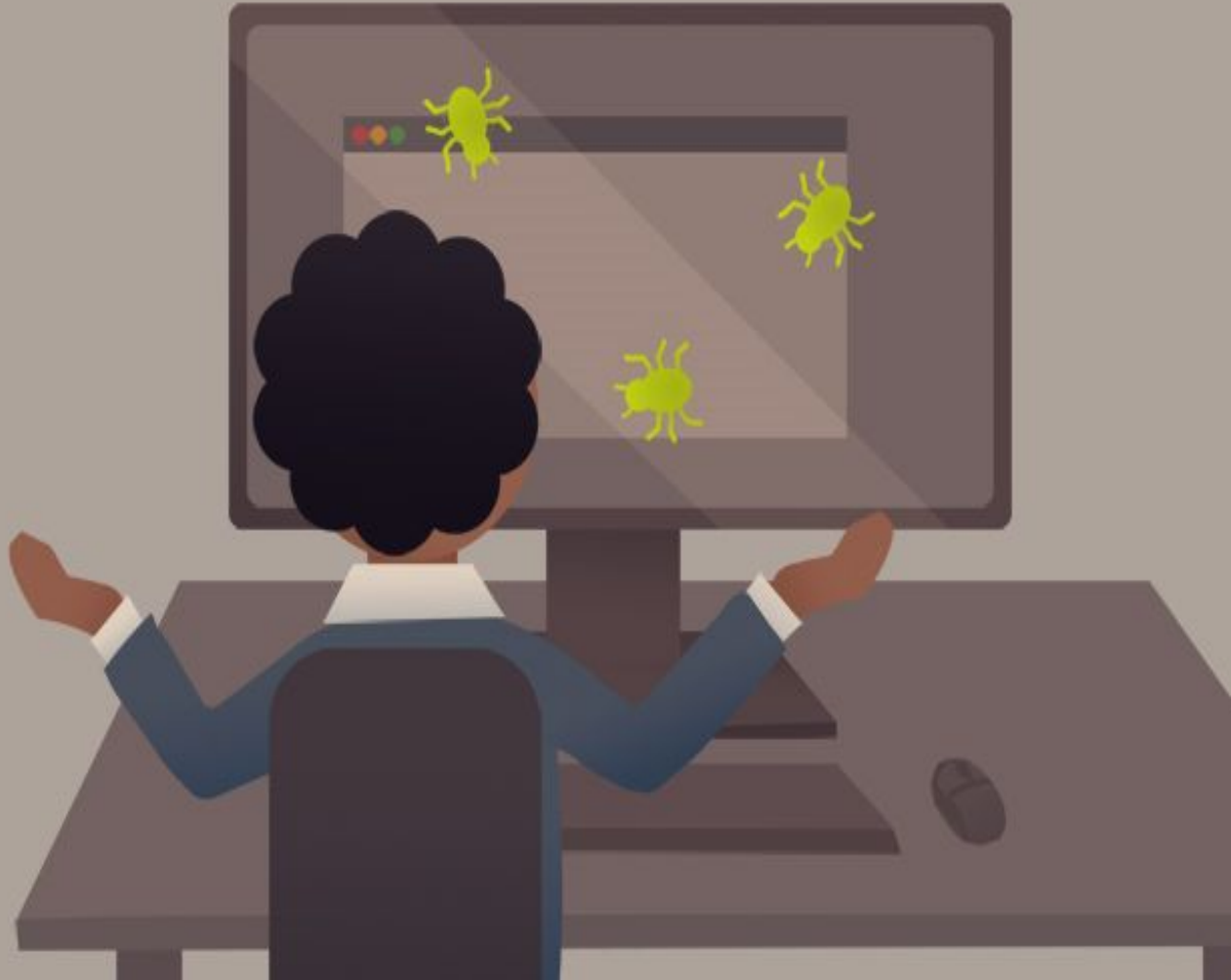
Example - Software allowed user to add 2 spouses to the policy.

Software Doesn't do something that specification doesn't mention but it should.

Example - Login button is in-between Username and Password field.

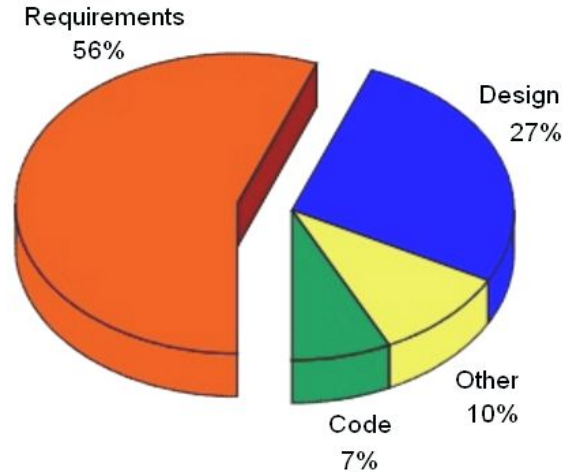
<https://www.youtube.com/watch?v=UlhGhj1tj-A>

Why Bugs Occur?

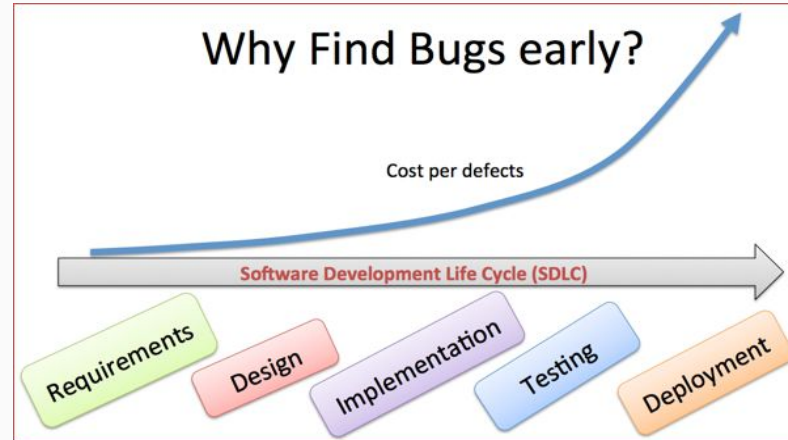
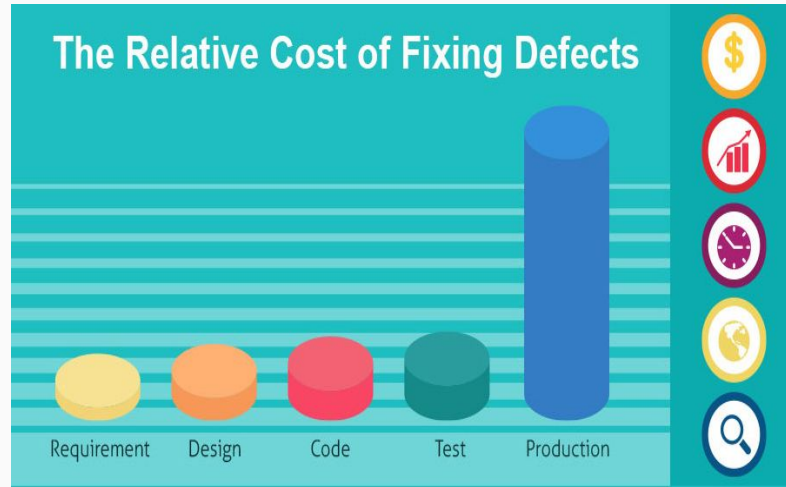


Why Bugs Occur?

It has been shown that some **56%** of all software defects emerge during the **requirement phase**, **27%** in the **design phase**, and only **7%** during the **development phase**.



Why Bugs Occur?



What should we do after a bug is found?

In Software testing, when the expected and actual behavior is not matching, an incident needs to be raised/reported.

Before reporting a bug:

- 1) ALWAYS: Try to **reproduce** the BUG
- 2) Check requirements (requirements update or functionality removed)
- 3) Once bug is reproducible you need to report it

How Reporting Works? What should be included in Reporting? Time

Priority and Severity of the Bug

- Severity is defined as the level of impact that bug has on the functionality of application/system.
- Severity indicates the seriousness of the defect on the product functionality
- Priority is defined as the order in which a defect should be fixed. Higher the priority the sooner the defect should be resolved.
- Priority indicates how soon the bug should be fixed

Priority categorization

- **Show Stopper:** Not able to test application further.
- **Major:** Major functionality not working but able to test application.
- **Minor:** The defect that does not result in the termination and does not damage the usability of the system
- **Cosmetic:** Issues in location of the object or the look and feel issue.

Severity categorization

- **Low:** The defect is an irritant but repair can be done once the more serious bugs have been fixed
- **Medium:** During the normal course of the development activities defect should be resolved. It can wait until a new version is created
- **High:** The defect must be resolved as soon as possible as it affects the system severely and cannot be used until it is fixed

Examples:

- **High Priority & High Severity:** User is unable to login to their Bank of America account
- **High Priority & Low Severity:** Bank of America logo on the front page misspelled as 'Bank of Amerika'
- **High Severity & Low Priority:** On Syntax Technologies website under 'Campus' section not able to scroll to view pictures
- **Low Priority and Low Severity:** Any cosmetic issues (Image is displaying on top right instead of center right) or spelling issues (when user trying to make purchase there is always acknowledge message)

Bug Life Cycle

Bug life cycle is the specific set of states that a Bug goes through from discovery to defect fixation. It varies from organization to organization, from project to project and also depends upon the tools used.

Bug Life Cycle:

- **New:** When a new defect is logged and posted for the first time. It is assigned a status NEW.
- **Rejected:** If the developer feels that the bug is not genuine, developer rejects the bug. Then the state of the bug is changed to “rejected”.
- **Assigned:** Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to developer team.

Bug Life Cycle

- **Open:** The developer starts analyzing and works on the defect fix
- **Fixed:** When developer makes necessary code change and verifies the change, he/she can make bug status as "Fixed."
- **Retest:** Tester do the retesting to check whether bug got fixed or not.
- **Reopened:** If the bug still exists even after the bug is fixed , the tester changes the status to "reopened". The bug goes through the life cycle once again.
- **Closed:** Once the bug is fixed, it is tested by the tester. If the tester feels that the bug no longer exists in the software, tester changes the status of the bug to "closed". This state means that the bug is fixed, tested and approved.

Bug Life Cycle

