EXPERIMENT: 2

(2K17/SE/79 PARV GUPTA)

AIM: Defect detection activities like reviews and testing help in identifying the defects in the artifacts (deliverables). These defects must be classified into various buckets before carrying out the root cause analysis. Following are some the defect categories: Logical, User interface, Maintainability, and Standards.
In the context of the above defect categories, classify the following statements under the defect categories.

THEORY:

RCA (Root Cause Analysis) is a mechanism of analyzing the Defects, to identify its cause. The RCA team brainstorms, reads and digs the defect to identify whether the defect was due to "testing miss", "development miss" or was a "requirement or design miss".

OUTPUT:

a. Divide by Zero Error is not guarded

Logical Defect:

Logical defects are mistakes done regarding the implementation of the code. They are related to the core of the software and happen when the programmer does not take care of the corner cases or doesn't understand the problem clearly or thinks in a wrong way. Not handling corner cases can lead to low-quality software causing crashes and other kinds of defects.

Poor Test Cases:
If this defect wasn't unearthed during the testing phase then it can be due to weak test cases. Test cases should be developed for both valid and invalid input conditions. Such test cases also help to evaluate the robustness of the software.

b. Usage of 3.14 in the statement Circle_Area = 3.14 * Radius * Radius;

Logical Defect:
Logical defects are mistakes done regarding the implementation of the code. Using 3.14 can lead to loss of precision for some applications. When the programmer doesn't understand the problem clearly or thinks in a wrong way then such types of defects happen. Also while implementing the code if the programmer doesn't take care of the corner cases then logical defects happen. It is basically related to the core of the software.

c. 3500 lines of code in a single function

Maintainability:
Managing large monolithic codebases is tough. And modifying such codebases is tougher. Decomposing a system into subsystems reduces the complexity developers have to deal with by simplifying the parts and increasing their coherence. A pointer is declared but not initialized. It is used in the program for storing a value.

d. A program designed to handle 1000 simultaneous users, crashed when 1001 the user logged in.

Logical Defect, Coding errors:

Logical defects are mistakes done regarding the implementation of the code. Also while implementing the code if the programmer doesn't take care of the corner cases then logical defects happen. Such errors are basically related to the core of the software.

e. A "while" loop never exits

Insufficient Testing:

The above program isn't robust. It is a "testing miss". Before deployment, load testing and stress testing of the software is performed. A load test, by definition, measures the performance of a system under an expected load. A stress test overloads a system in order to find the breaking point.

Incorrect design:

The system should have been designed to be robust and reliable. Why on adding only a single user the system crashed? Incorrect design makes software vulnerable to failures.

Performance Defect:

Performance defects are the defects when the system or the software application is unable to meet the desired and the expected results. It also includes the response of the system with the varying load on the system.

f. User interface displays "MALFUNCTION 54" when something goes wrong in the back-end

Logical Defect:

Logical defects are related to the core of the software. Logical defects are mistakes done regarding the implementation of the code. These happen due to negligence on part of the

developers.

g. No documentation (comments) for the source code

User Interface defect:

Interface defects means the defects in the interaction of the software and the users. The system may suffer different kinds of the interface testing in the forms of the complicated interface, unclear interface or the platform based interface.

No proper UI testing:

One of the main aspects of GUI testing is checking whether correct error messages are being displayed. UI Testing is done to uncover such User Interface Bugs. UI testing has become integral to the Software development process, where we analyze an application from the user's point of view.

Since a bug-free interface is a big step forward to a successful product release, listed below are the main aspects of GUI testing checklist:

- Screen with its control of buttons, icons, images, menus is the basic element testers check.
- The correct display of error messages.
- Text check-up: readable font, color, proper text alignment.
- Images are checked for clarity, proper alignment.
- GUI elements are tested for compliance with different screen resolutions.

h. Hungarian Notation not followed while coding, even though the coding guidelines mandate to use Hungarian Notation

Logical Defect:
When the programmer doesn't understand the problem clearly or thinks in a wrong way then such types of defects happen. Logical defects are mistakes done regarding the implementation of the code. These happen due to negligence on part of the developers.

Poor Standards:
It is considered to be a good coding standard to name the functions according to what they perform.(Hungarian Notation). Coding guidelines were not followed. Not compliant with the requirement document. All coding guidelines should be followed for successful software development.

i. Pressing of "Tab" key moves the cursor in different fields of a web form randomly.

Interface Defects:
Interface defects means the defects in the interaction of the software and the users. The system may suffer different kinds of the interface testing in the forms of the complicated interface, unclear interface or the platform based interface.

No proper UI testing:
One of the main aspects of GUI testing is checking whether correct error messages are being displayed. UI Testing is done to uncover such User Interface Bugs.