Software Errors and Bugs

As defined in Wikipedia "*An **error** is a deviation from accuracy or correctness*" and "*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* ".

So, the following can be inferred:

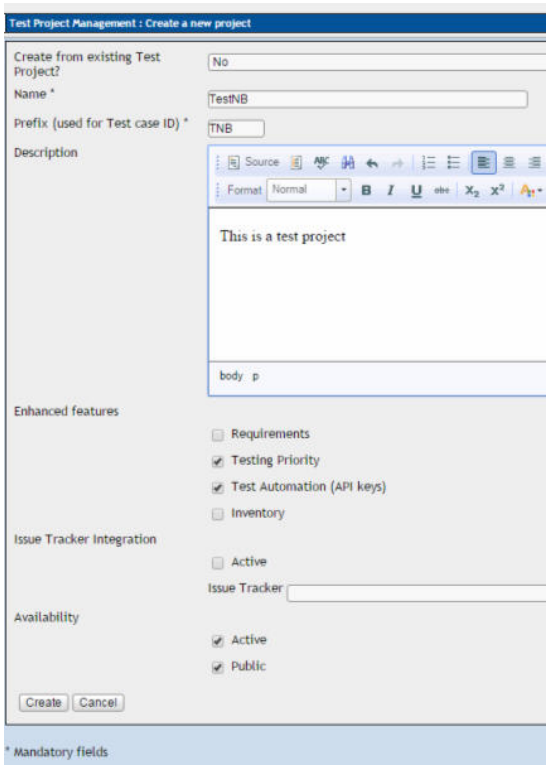
- Error is a variance of the actual result from the expected result.
- Errors are a category of software bugs.
- Errors can be introduced as result of incomplete or inaccurate requirements or due to human data entry problems.

Common Categories of Software Errors:

#1) Functionality Errors:

Functionality is a way the software is intended to behave. **Software has a functionality error if something that you expect it to do is hard, awkward, confusing, or impossible.**

Check this screenshot:



Expected Functionality for Cancel button is that the 'Create new project' window should close and none of the changes should be saved (i.e. no new project must be created). If the Cancel button is not clickable then it is a functionality error.

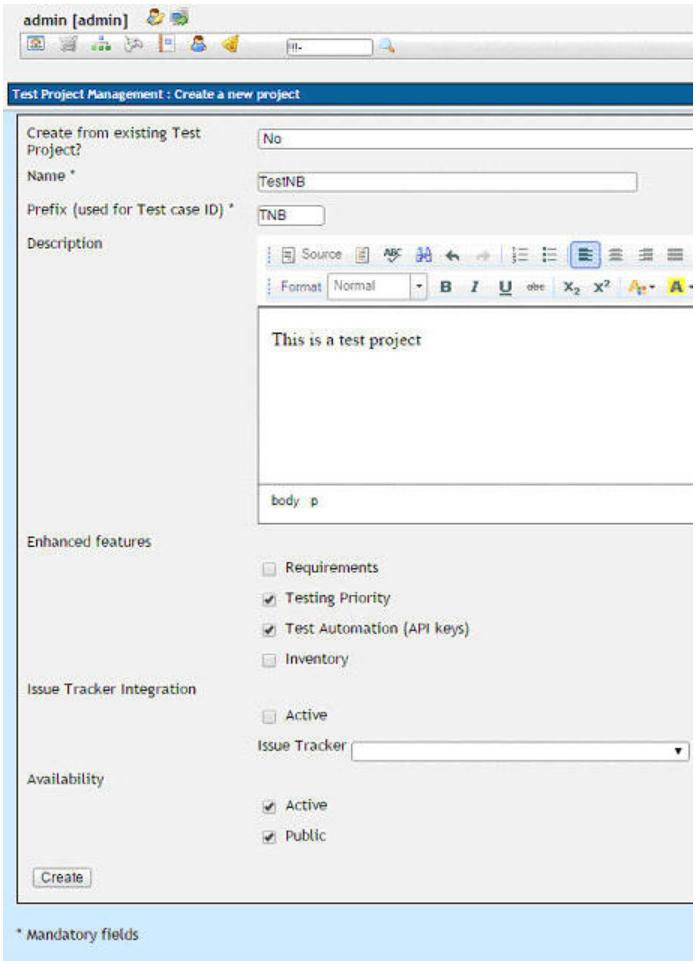
#2) Communication Errors:

These errors occur in communication from software to end-user. **Anything that the end user needs to know in order to use the software should be made available on screen**.

Few examples of communication errors are – No Help instructions/menu provided, features that are part of the release but are not documented in the help menu, a button named 'Save' should not erase a file etc.

#3) Missing command errors:

This happens to occur when an expected command is missing. See this screenshot:

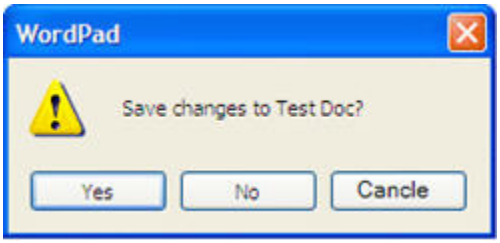


This window allows the user to create a new project. However, there is no option for the user to exit from this window without creating the project. Since 'Cancel' option/button is not provided to the user, this is a missing command error.

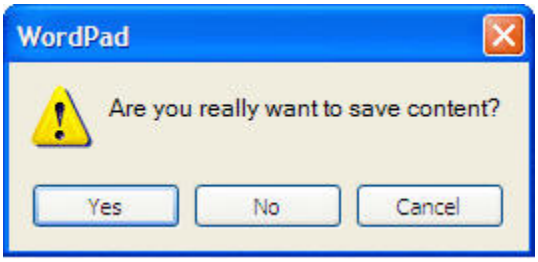
#4) Syntactic Error:

Syntactic errors are misspelled words or grammatically incorrect sentences and are very evident while testing software GUI. Please note that we are NOT referring to syntax errors in code. The compiler will warn the developer about any syntax errors that occur in the code

Note the misspelled word 'Cancel':



Note the grammatically incorrect message:



#5) Error handling errors:

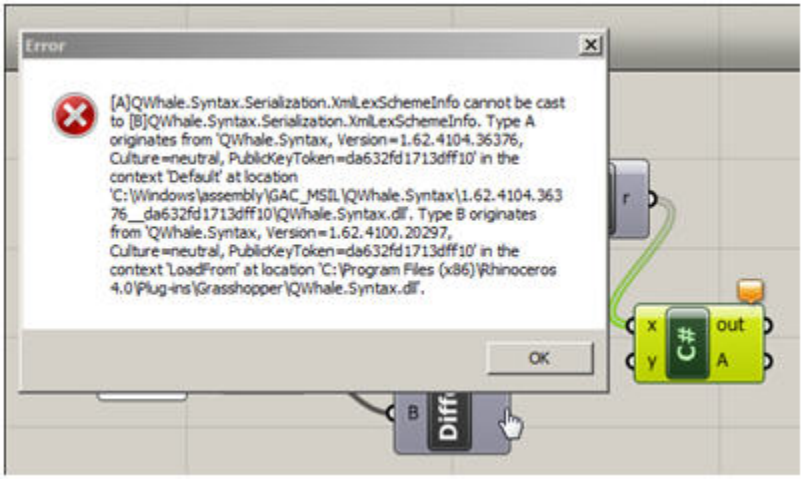
Any errors that occur while the user is interacting with the software needs to be handled in a clear and meaningful manner. If not, it is called as an Error Handling Error.

Take a look at this image. The error message gives no indication of what the error actually is. Is it missing mandatory field, saving error, page loading error or is it a system error? Hence, this is an 'Error Handing Error'.

When possible, further steps should be listed for the user to follow.

If the software has certain mandatory fields that need to be filled before they can save the information on a form, the validation messages should be clear and indicative of the action that is required by the user.

Here are other examples:



#6) Calculation Errors:

These errors occur due to any of the following reasons:

- Bad logic
- Incorrect formulae
- Data type mismatch
- Coding errors
- Function call issues , etc.

In 1999, NASA lost its Mars climate orbiter because one of the subcontractors NASA employed had used English units instead of the intended metric system, which caused the orbiter's thrusters to work incorrectly. Due to this bug, the orbiter crashed almost immediately when it arrived at Mars.

#7) Control flow errors:

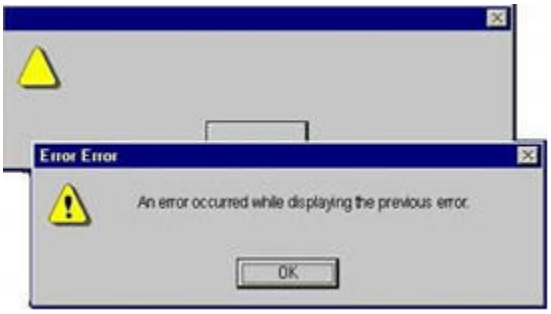
The control flow of a software describes what it will do next and on what condition.

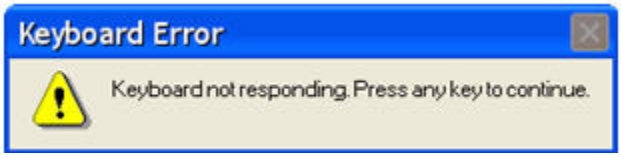
For example, consider a system where user has to fill in a form and the options available to user are: Save, Save and Close, and Cancel. If a user clicks on 'Save and Close' button, the user information in the form should be saved and the form should close. If clicking on the button does not close the form, then it is a control flow error.

An Exercise:

Let's identify what error categories the following fall into:

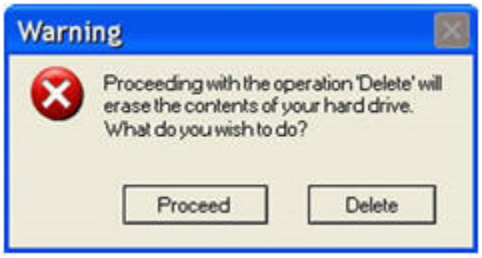
Exercise #1:





These are Error Handling Errors.

Exercise #2:



This is a Missing command error. Cancel button is required but is missing. Also, both buttons 'Proceed' and 'Delete' are redundant and perform the same function.

Exercise #3

This is Syntactic Error.

Next step:

Reporting an error once identified is essential. For best results, report immediately.

Include the description, priority, severity, the triggers and steps to recreate the scenario, screen captures (if any) in the bug report.

For more information on writing effective defect reports, check this post.

Conclusion

Defect identification, categorization, reporting and eventually removal are all part of Quality Control activities.

But, Prevention is better than cure. The very crux of Software Quality Assurance is to establish monitoring and inspecting processes at each stage of the Software Development Life Cycle.

The aim is to detect errors as early as possible. This is because the costs to find and fix errors increase dramatically as software development progresses. Hence identifying errors early on is essential.

Fixing an error is the cheapest during the requirement analysis stage, gets progressively expensive with each stage and is most expensive in the post release maintenance phase.

As QA engineers, we may or may not be directly involved in requirements definition. We also may have little or no direct control on the quality of the requirements.

Therefore, it is essential that we are able to identify, seek and report any errors that we come across during testing phase.