SPM: SOFTWARE PROJECT MANAGEMENT
SPM: UNIT— 04/17

12 APRIL 2022 / IT-3rd year, VIth semester
FALL SEMESTER, YEAR (VIth, 3rd)
FALL SESSION (2021-22)
(SPM)
MS. SHWETA TIWARI
Published: April 12, 2022

SPM: SOFTWARE PROJECT MANAGEMENT

TOPIC On : Difference between Bug, Defect, Error, Fault & Failure

—

By SHWETA TIWARI

Under On: SOFTWARE PROJECT MANAGEMENT

TOPIC On : Difference between Bug, Defect, Error, Fault & Failure

Difference between Bug, Defect, Error, Fault & Failure

Difference between the **Bug, Defect, Error, Fault & Failure** as we understood that all the terms are used whenever the system or an application acts abnormally.

Sometimes we call it an **error** and sometimes a **bug** or a **defect** and so on. In software testing, many of the new test engineers have confusion in using these terminologies.

Generally, we used these terms in the Software Development Life Cycle (SDLC)

based on the phases. But there is a conflict in the usage of these terms.

In other words, we can say that in the era of **software testing,** the terms **bugs, defects, error, fault, and failure** come across every second of the day.

44.3M

892

C++ vs Java

But for a beginner or the inexperienced in this field, all these terminologies may seem synonyms. It became essential to understand each of these terms independently if the software doesn't work as expected.

What is a bug?

In software testing

, a bug

is the informal name of defects, which means that software or application is not working as per the requirement. When we have some coding error, it leads a program to its breakdown, which is known as **a bug**. The **test engineers** use the terminology **Bug**.

If a **QA (Quality Analyst)**

detect a bug, they can reproduce the bug and record it with the help of the **bug report template**.

What is a Defect?

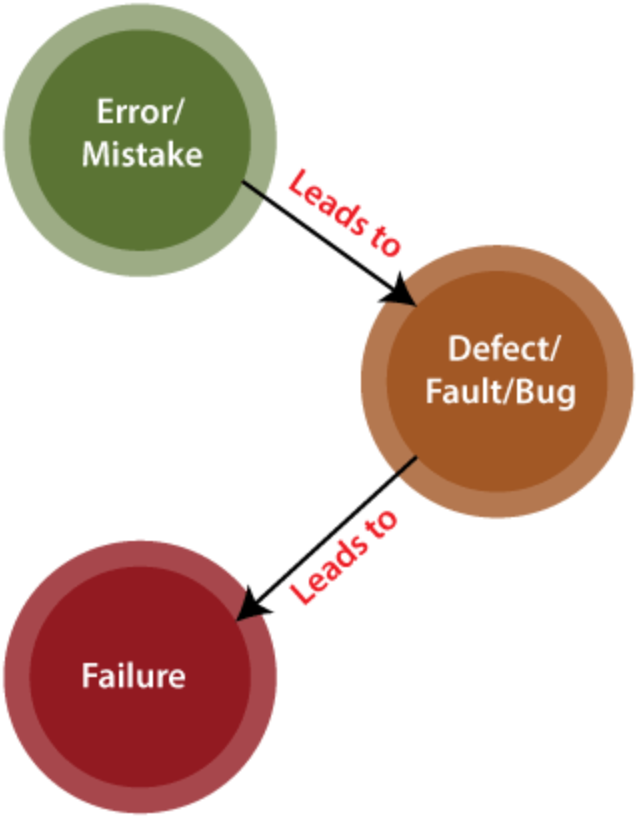
When the application is not working as per the requirement is knows as **defects.** It is specified as the aberration from the **actual and expected result** of the application or software.

In other words, we can say that the bug announced by the **programmer** and inside the code is called a **Defect**

.

What is Error?

The Problem in code leads to errors, which means that a mistake can occur due to the developer's coding error as the developer misunderstood the requirement or the requirement was not defined correctly. The **developers** use the term **error**.



What is Fault?

The fault may occur in software because it has not added the code for fault tolerance, making an application act up.

A fault may happen in a program because of the following reasons:

- Lack of resources
- An invalid step
- Inappropriate data definition

What is Failure?

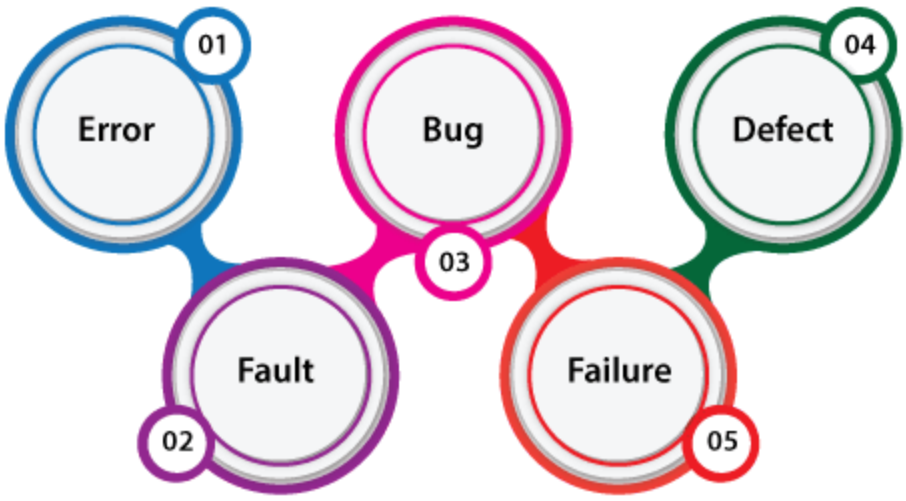
Many defects lead to the **software's failure**, which means that a loss specifies a fatal issue in software/ application or in its module, which makes the system unresponsive or broken.

In other words, we can say that if an end-user detects an issue in the product, then that particular issue is called a **failure**.

Possibilities are there is one defect that might lead to one failure or several failures.

For example, in a bank application if the **Amount Transfer** module is not working for end-users when the end-user tries to **transfer money**, the submit button is not working. Hence, this is a **failure**.

The flow of the above terminologies are shown in the following image:



Bug Vs. Defect Vs. Error Vs. Fault Vs. Failure

We have listed some of the vital differences between **bug, defect, error, fault, and failure in the below table**.

Comparison basis	Bug	Defect	Error	Fault	Failure

Definition	It is an informal name specified to the defect.	The **Defect** is the difference between the actual outcomes and expected outputs.	An **Error** is a mistake made in the code; that's why we cannot execute or compile code.	The **Fault** is a state that causes the software to fail to accomplish its essential function.	If the software has lots of defects, it leads to failure or causes failure.
Raised by	The **Test Engineers** submit the bug.	The **Testers** identify the defect. And it was also solved by the developer in the development phase or stage.	The **Developers and automation test engineers** raise the error.	**Human mistakes** cause fault.	The failure finds by the manual test engineer through the **development cycle**.

Different types	Different type of bugs are as follows: • Logic bugs • Algorithmic bugs • Resource bugs	Different type of Defects are as follows: Based on **priority**: • High • Medium • Low And based on the severity: • Critical • Major • Minor • Trivial	Different type of Error is as below: • Syntactic Error • User interface error • Flow control error • Error handling error • Calculation error • Hardware error	Different type of Fault are as follows: • Business Logic Faults • Functional and Logical Faults • Faulty GUI • Performance Faults • Security Faults • Softw are/ hardw	-----

			• Testing Error	are fault	

Reasons behind	Following are reasons which may cause the **bugs:** Missing coding Wrong coding Extra coding	The below reason leads to the **defects**: Giving incorrect and wrong inputs. Dilemmas and errors in the outside behavior and inside structure and design. An error in coding or logic affects the software and causes it to break down or fail.	The reasons for having an **error** are as follows: Errors in the code. The Mistake of some values. If a developer is unable **to compile or run a program successfully**. Confusions and issues in programming. Invalid login, loop, and syntax.	The reasons behind the **fault** are as follows: A Fault may occur by an improper step in the initial stage, process, or data definition. Inconsistency or issue in the program. An irregularity or loophole in the software that leads the software to perform improperly.	Following are some of the most important reasons behind the **failure:** Environmental condition System usage Users Human error

			Inconsistency between actual and expected outcomes. Blunders in design or requirement actions. Misperception in understanding the requirements of the application.		

Way to prevent the reasons	Following are the way to stop the **bugs**: Test-driven development. Offer programming language support. Adjusting, advanced, and operative development procedures. Evaluating the code systematically.	With the help of the following, we can prevent the **Defects**: Implementing several innovative programming methods. Use of primary and correct software development techniques. Peer review It is executing consistent code reviews to evaluate its quality and correctness.	Below are ways to prevent the **Errors**: Enhance the software quality with system review and programming. Detect the issues and prepare a suitable mitigation plan. Validate the fixes and verify their quality and precision.	The **fault** can be prevented with the help of the following: Peer review. Assess the functional necessities of the software. Execute the detailed code analysis. Verify the correctness of software design and programming.	The way to prevent **failure** are as follows: Confirm re-testing. Review the requirements and revisit the specifications. Implement current protective techniques. Categorize and evaluate errors and issues.

Conclusion

After seeing all the significant differences between **bug, defect, error, fault, and failure**, we can say that the several issues and inconsistencies found throughout software are linked and dependent on each other.

All the above terminology affects and changes different parts of the software and differs from one another massively. However, all these differences between **bug, defect, errors, faults, and failures** slow down the software's excellence and performance.