

Bug taxonomies

1. Taxonomy from Catolino et al.

- Based on *Gemma Catolino, Fabio Palomba, Andy Zaidman, and Filomena Ferrucci. Not all bugs are the same: Understanding, characterizing, and classifying bug types. Journal of Systems and Software, 152:165–181, June 2019.*

Defect Type	Definition
Configuration issue	Bugs related to the configuration of the project. Most are problems with: versions of external libraries
Network issue	Bugs having connection or server issues, due to network problems, unexpected server shutdowns, or communication protocols that are not properly used within the source code.
Database-related issue	Bugs that report problems with the connection between the main application and a database. Note that bugs in SQL statements are also part of this category.
GUI-related issue	Bugs occurring within the (Graphical) User Interface (GUI) of a software project, such as stylistic errors (layouts, padding, buttons), and unexpected failures appearing to the user.
Performance issue	Bugs that report performance issues, including memory overuse, energy leaks, and methods causing endless loops.
Permission/Deprecation issue	Bugs related to: (i) the presence, modification, or removal of deprecated method calls or APIs; (ii) unused API permissions.
Security issue	Problems related to vulnerabilities in the system.
Program Anomaly issue	Bugs concerned with specific circumstances that appear when enhancing existing source code, such as exceptions, return values, issues in logic.

2. Taxonomy from Seaman et al.

- Based on Carolyn B. Seaman, Forrest Shull, Myrna Regardie, Denis Elbert, Raimund L. Feldmann, Yuepu Guo, and Sally Godfrey. *Defect categorization: making use of a decade of widely varying historical data. In Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement, pages 149–157, Kaiserslautern Germany, October 2008. ACM.*

Defect Type	Definition
Algorithm / Method	An error in the sequence or set of steps used to solve a particular problem or computation (e.g. mistakes in computations, incorrect implementation of algorithms, or calls to an inappropriate function for the algorithm being implemented).
Assignment / Initialization	A variable or data item that is assigned a value incorrectly or is initialized improperly (e.g. changing the signature of a method)
Checking	Inadequate checking or handling of potential errors.
Data	Error in specifying or manipulating data items (e.g. incorrectly defined data structure, or incorrect type conversions).
External Interface	Errors in the user interface (including usability problems) or the interfaces with other systems.
Internal Interface	Errors in the interfaces between system components, including mismatched calling sequences and incorrect opening, reading, writing or closing of files and databases.
Logic	Incorrect logical conditions (e.g. inside an if, case or pattern, including incorrect boundary conditions) or incorrect expression (e.g. incorrect use of parentheses in a mathematical expression).
Non-Functional Defects	Includes non-compliance with standards, non-functional requirements such as portability and performance constraints, and lack of clarity of the design or code to the reader both in the comments and the code itself.
Timing / Optimization	Errors that will cause timing (e.g., potential race conditions) or performance problems (e.g., unnecessarily slow implementation of an algorithm).
Other	Anything that does NOT fit in the types above.