

Subtypes in the taxonomy of code review changes

SubType	Macro Category	Taxonomy Type	Description
Naming	Evolvability	Textual	Naming changes usually occur when the name of a software element does not respect the policy of the system or is not meaningful.
Comments	Evolvability	Textual	Comment changes occur when comments need to be updated: for instance, because new information are needed to describe the code after a modification.
Debug Info	Evolvability	Textual	Debug changes occur when logging messages or debug messages need to be update as the information they contain is not relevant anymore.
Other Textual	Evolvability	Textual	Other Textual changes include all textual changes that do not belong in another existing textual subtype.
Element Type	Evolvability	Supported by Language	Element Type changes occur when the type of a software element is incorrect (only cases not causing run-time errors). For instance, a method returning the wrong variable type.
Visibility	Evolvability	Supported by Language	Visibility changes occur when a software element has the wrong scope. For instance, a public variable that is changed to private or vice versa.
Void Parameter	Evolvability	Supported by Language	Void Parameter changes mostly refer to an incorrect use of the keyword “void”.
Element Reference	Evolvability	Supported by Language	Element Reference changes occur when incomplete names are used to refer to a software element. A common example is the misuse of the “this” keyword in Java.
Bracket Usage	Evolvability	Visual Representation	Bracket Usage changes happen when the use of brackets in the code is incorrect.
Indentation	Evolvability	Visual Representation	Indentation changes when the indentation of the code is wrong or does not respect the project policies.
BlankLine Usage	Evolvability	Visual Representation	Blank Line usage changes occur when there are too many or too few blank lines in the code formatting.
Long Line	Evolvability	Visual Representation	Long Line changes occur when a long line need to be split in separate lines to grant code readability.
Space Usage	Evolvability	Visual Representation	Space Usage changes occur as the code contains too many or too few blank space characters.
Grouping	Evolvability	Visual Representation	Grouping changes occur when software elements of the same type need to be grouped in the same packages, modules or classes or software elements of different type need to be splitted in separated packages, modules or classes.

SubType	Macro Category	Taxonomy Type	Description
Move Functionality	Evolvability	Organization	Move Functionality changes occur when a function, a part of a function, a module or a class need to be moved to a different part of the code or system. We included in this sub-type also cases where a functionality has been moved across the same class.
Long Sub-Routine	Evolvability	Organization	Long Sub-Routine changes occur as complex loops or functions need to be split because of its excessive length or functionality
Dead Code	Evolvability	Organization	Dead Code changes occur when parts of the code are removed because they are never used.
Duplication	Evolvability	Organization	Duplication changes occur when the code contains duplicated parts that need to be removed.
Complex Code	Evolvability	Organization	Complex Code changes occur when the structure of the code is excessively complex.
Statement Issue	Evolvability	Organization	Statement Issue changes occur when statements in the code need to be combined or reorganised.
Consistency	Evolvability	Organization	Consistency changes occur when similar software elements have different behaviours and they need to be modified to make their workflow similar. For instance, similar tasks should be implemented similarly in similar classes.
Other Organization	Evolvability	Organization	Other Organization changes group all organization defects that cannot be assigned to any another organization subtype.
Semantic Duplication	Evolvability	Solution Approach	Semantic Duplication changes occur when different parts of the code have the same goal but their implementation is different.
Semantic Dead Code	Evolvability	Solution Approach	Semantic Dead Code changes occur when parts of the code are executed but they do not have any meaningful purpose.
Change Function	Evolvability	Solution Approach	Change Function changes occur when old or deprecated functions in the code need to be updated.
Standard Method	Evolvability	Solution Approach	Standard Method changes occur when parts of code are replaced by standardised way of working. For instance, substituting magic number with predefined constants.
New Functionality	Evolvability	Solution Approach	New Functionality changes occur when a new functionality is required to make the software more evolvable.
Other Solution Approach	Evolvability	Solution Approach	Other solution Approach changes include solution approach defects that do not belong in any other solution approach subtype.

SubType	Macro Category	Taxonomy Type	Description
Variable Initialization	Functional	Resource	Variable Initialization changes occur when a variable needs to be initialized before its use in order to avoid run-time failures.
Memory Management	Functional	Resource	Memory Management changes occur when the system memory is not handled correctly.
Data & Resource Manipulation	Functional	Resource	Data & Resource Manipulation changes occur when data need to be manipulated in a certain way to avoid problem related to resources, like closing buffers or threads management. Furthermore, we extended this subtype to include changes concerning wrong variables assignments.
Check Function	Functional	Check	Check Function changes occur when the return value of a function needs to be checked prior its usage.
Check Variable	Functional	Check	Check Variable changes occur when a variable needs to be checked before its usage.
Check User Input	Functional	Check	Check User Input changes occur when the user input needs to be controlled before its usage.
Function Call	Functional	Interface	Function Call changes occur when a function call is missing or incorrect.
Parameter	Functional	Interface	Parameter change occur when a function call has wrong or missing parameters.
Compare	Functional	Logic	Compare changes occur when there is a mistake in a comparison statement.
Compute	Functional	Logic	Compute changes occur when a computation produces incorrect results.
Wrong Location	Functional	Logic	Wrong Location changes occur when a correct operation have been executed in the wrong place inside the code: it should have been performed earlier or later in the code flow.
Algorithm/Performance	Functional	Logic	Algorithm/Performance changes occur when the performance of an algorithm must be improved.
Other Logic	Functional	Logic	Other Logic changes include logic defects that can not be included in any other logic defects subtype.
Completeness	Functional	Larger defects	Completeness changes occur when a feature is not completely implemented, producing runtime failure or incorrect results.
GUI	Functional	Larger defects	GUI changes occur when there are inconsistencies in the user interfaced leading to errors.
Check outside code	Functional	Larger defects	Check outside code changes are modifications that require to check parts of code that were not among the ones under review, as they might contain defects based on the code under inspection.