Defect Type for Design/Code (IBM ODC v 5.11)

Assignment/Initialization

Value(s) assigned incorrectly or not assigned at all; but note that a fix involving multiple assignment corrections may be of type Algorithm.

Examples:

1) Internal variable or variable within a control block did not have correct value, or did not have any value at all.
2) Initialization of parameters
3) Resetting a variable's value.
4) The instance variable capturing a characteristic of an object (e.g., the color of a car) is omitted.
5) The instance variables that capture the state of an object are not correctly initialized.

Checking

Errors caused by missing or incorrect validation of parameters or data in conditional statements. It might be expected that a consequence of checking for a value would require additional code such as a do while loop or branch. If the missing or incorrect check is the critical error, checking would still be the type chosen.

Examples:

1) Value greater than 100 is not valid, but the check to make sure that the value was less than 100 was missing.
2) The conditional loop should have stopped on the ninth iteration. But it kept looping while the counter was <=10.

Algorithm/Method

Efficiency or correctness problems that affect the task and can be fixed by (re)implementing an algorithm or local data structure without the need for requesting a design change. Problem in the procedure, template, or overloaded function that describes a service offered by an object.

Examples:

1) The low-level design called for the use of an algorithm that improves throughput over the link by delaying transmission of some messages, but the implementation transmitted all messages as soon as they arrived. The algorithm that delayed transmission was missing.
2) The algorithm for searching a chain of control blocks was corrected to use a linear-linked list instead of a circular-linked list.
3) The number and/or types of parameters of a method or an operation are incorrectly specified.
4) A method or an operation is not made public in the specification of a class.

Function/Class/Object

The error should require a formal design change, as it affects significant capability, end-user interfaces, product interfaces, interface with hardware architecture, or global data structure(s); The error occurred when implementing the state and capabilities of a real or an abstract entity.

Examples:

1) A database did not include a field for street address, although the requirements specified it.
2) A database included a field for postal zip code, but it was too small to contain international postal codes as specified in the requirements.
3) A C++ or SmallTalk class was omitted during system design.

Timing/Serialization

Necessary serialization of shared resource was missing, the wrong resource was serialized, or the wrong serialization technique was employed.

Examples:

1) Serialization is missing when making updates to a shared control block.
2) A hierarchical locking scheme is in use, but the defective code failed to acquire the locks in the prescribed sequence.

Interface/O-O Messages

Communication problems between:
1) modules, 2) components, 3) device drivers, 4) objects, 5) functions
Via 1)macros, 2)call statements, 3)control blocks, 4)parameter lists

Examples:

1) A database implements both insertion and deletion functions, but the deletion interface was not made callable.
2) The interfaces specifies a pointer to a number, but the implementation is expecting a pointer to a character.
3) The OO-message incorrectly specifies the name of a service.
4) The number and/or types of parameters of the OO-message do not conform with the signature of the requested service.

Relationship

Problems related to associations among procedures, data structures and objects. Such associations may be conditional.

Examples:

1) The structure of code/data in one place assumes a certain structure of code/data in another. Without appropriate consideration of their relationship, program will not execute or it executes incorrectly.
2) The inheritance relationship between two classes is missing or incorrectly specified.
3) The limit on the number of objects that may be instantiated from a given class is incorrect and causes performance degradation of the system.