

Root Cause Checklist

Debugging by Thinking: A Multidisciplinary Approach

Robert Charles Metzger, Elsevier Digital Press, 2004

www.debuggingbythinking.com

1. Design errors
 1. Data structures
 1. A data definition is missing.
 2. A data definition is incorrect.
 3. A data definition is unclear.
 4. A data definition is contradictory.
 5. A data definition is out of order.
 6. A shared-data access control is missing.
 7. A shared-data access control is incorrect.
 8. A shared-data access control is out of order.
 2. Algorithms
 1. A logic sequence is missing.
 2. A logic sequence is superfluous.
 3. A logic sequence is incorrect.
 4. A logic sequence is out of order.
 5. An input check is missing.
 6. An input check is superfluous.
 7. An input check is incorrect.
 8. An input check is out of order.
 9. An output definition is missing.
 10. An output definition is superfluous.
 11. An output definition is incorrect.
 12. An output definition is out of order.
 13. A special-condition handler is missing.
 14. A special-condition handler is superfluous.
 15. A special-condition handler is incorrect.
 16. A special-condition handler is out of order.
3. User-interface specification
 1. An assumption about the user is invalid.
 2. A specification item is missing.
 3. A specification item is superfluous.
 4. A specification item is incorrect.
 5. A specification item is unclear.
 6. Specification items are out of order.
4. Software-interface specification
 1. An assumption about collateral software is invalid.
 2. A specification item is missing.
 3. A specification item is superfluous.
 4. A specification item is incorrect.

5. A specification item is unclear.
6. Specification items are out of order.
5. Hardware-interface specification
 1. An assumption about the hardware is invalid.
 2. A specification item is missing.
 3. A specification item is superfluous.
 4. A specification item is incorrect.
 5. A specification item is unclear.
 6. Specification items are out of order.
2. Coding errors
 1. Initialization errors
 1. A simple variable is always uninitialized.
 2. A simple variable is sometimes uninitialized.
 3. A simple variable is initialized with the wrong value.
 4. An aggregate variable is always uninitialized.
 5. An aggregate variable is sometimes uninitialized.
 6. An aggregate variable is initialized with the wrong value.
 7. An aggregate variable is partially uninitialized.
 8. An aggregate variable is not allocated.
 9. An aggregate variable is allocated the wrong size.
 10. A resource is not allocated.
 2. Finalization errors
 1. An aggregate variable is not freed.
 2. A resource is not freed.
 3. Binding Errors
 1. A variable is declared with the wrong scope.
 2. A procedure call is missing arguments.
 3. A procedure call has the wrong argument order.
 4. A procedure call has extra arguments.
 5. The actual argument passing mechanism does not match the usage of the formal argument.
 6. A procedure returns no value.
 7. A procedure returns the wrong value.
 4. Reference errors
 1. The wrong procedure is called.
 2. The wrong variable is referenced.
 3. The wrong constant is referenced.
 4. The wrong variable is assigned.
 5. A variable is not assigned.
 5. Static data-structure problems
 1. A simple variable has the wrong data type.
 2. An element of an aggregate variable has the wrong data type.
 3. An aggregate variable has the wrong aggregate size.
 6. Dynamic data-structure problems
 1. An array subscript is out of bounds.

2. An array subscript is incorrect.
 3. An uninitialized pointer has been dereferenced.
 4. A null pointer has been dereferenced.
 5. A pointer to freed memory has been dereferenced.
 6. An uninitialized pointer has been freed.
 7. A pointer stored in freed memory has been dereferenced.
 8. A null pointer has been freed.
 9. A pointer to freed memory has been freed.
 10. A pointer to static memory has been freed.
 11. A pointer to automatic (stack) memory has been freed.
7. Object-oriented problems
 1. A class containing dynamically allocated memory does not have the required methods.
 2. A base class has methods declared incorrectly for the derived class to override.
 3. The wrong method signature is used to invoke an overloaded method.
 4. A method from the wrong class in the inheritance hierarchy is used.
 5. A derived class does not completely implement the required functionality.
8. Memory problems
 1. Memory is corrupted.
 2. The stack is corrupted.
 3. The stack overflows.
 4. The heap is corrupted.
 5. A pointer is invalid for the address space.
 6. An address has an invalid alignment.
9. Missing operations
 1. A return code or flag has not been set.
 2. A return code or flag has not been checked.
 3. An exception has not been thrown.
 4. An exception thrown has not been handled.
 5. An event sequence has not been anticipated.
 6. A program state has not been anticipated.
 7. Statements are missing.
 8. Procedure calls are missing.
10. Extra operations
 1. A return code or flag is set when not needed.
 2. An exception is thrown when not valid.
 3. Extraneous statements are executed.
 4. Extraneous procedure calls are executed.
11. Control-flow problems
 1. Statements are controlled by the wrong control-flow condition.
 2. Loop iterations are off by one.

3. A loop terminates prematurely.
 4. A loop runs indefinitely.
 5. A case in a multiway branch is missing.
 6. A multiway branch takes the wrong case.
 7. A statement is executed too many times.
 8. A statement is executed too few times.
12. Value-corruption problems
1. An arithmetic operation has underflow or overflow.
 2. Precision is lost.
 3. Signed and unsigned integers are mixed.
 4. Floating-point numbers are compared incorrectly.
13. Invalid expressions
1. The wrong variable is used.
 2. Operator input is invalid.
 3. The wrong arithmetic operator is used.
 4. The wrong arithmetic expression order is used.
 5. The wrong relational operator is used.
 6. The wrong relational expression order is used.
 7. The wrong Boolean operator is used.
 8. The wrong Boolean expression order is used.
 9. A term is missing.
 10. There is an extra term.
14. Typographical errors
1. Characters are missing.
 2. There are extra characters.
 3. Characters are out of order.
15. Other people's problems
1. A compiler has been used incorrectly.
 2. A software tool has been used incorrectly.
 3. A system library has been used incorrectly.
 4. A third-party library has been used incorrectly.
 5. There is a defect in a system library.
 6. There is a defect in a compiler.
 7. There is a defect in a software tool.
 8. There is a defect in the operating system.
 9. There is a defect in the third-party software.