Identify global top 10 software defects due to lack of testing.

1. Null pointer dereference: This is caused when a program tries to access memory through a pointer that has not been initialized.
2. Off-by-one errors: These are among the most common faults. They occur when a loop iterates one time too many or too few.
3. Uninitialized variable: This occurs when a variable is used before it has been initialized.
4. Memory leaks: They occur when a program fails to release memory it has allocated. This can cause an application to run out of memory.
5. Buffer overflow: This happens when more data is written to a piece of memory than it can handle, causing it to overwrite adjacent memory.
6. Race condition: This happens when the behavior of a system depends on the sequence or timing of other uncontrollable events.
7. Use after free: This defect occurs when a program continues to use a pointer after it has been freed.
8. Integer overflow: This happens when an integer value is increased beyond its maximum value, causing it to wrap around and become a very small, possibly negative number.
9. Division by zero: This is caused by any operation that attempts to divide by zero.
10. Deadlock: This happens when two or more tasks permanently block each other by each task waiting for the other to release a resource.