



Debugging Approaches

Debugging software under test

By; Paul Gichure

@GichureMkenya

Introduction

- Debugging is a methodical process of finding and reducing the number of bugs, or defects, in a computer system (Software, hardware or a combination), thus making it behave as expected.
- Process;
 - to start with a problem
 - isolate the source of the problem
 - and then fix it

Debugging Vs Testing

- Debugging is carried out by the development team (or developer), after getting the test report from the testing team about defect(s)
- The purpose of testing is to show that the software works, OR doesn't work, with intention to increase SW quality

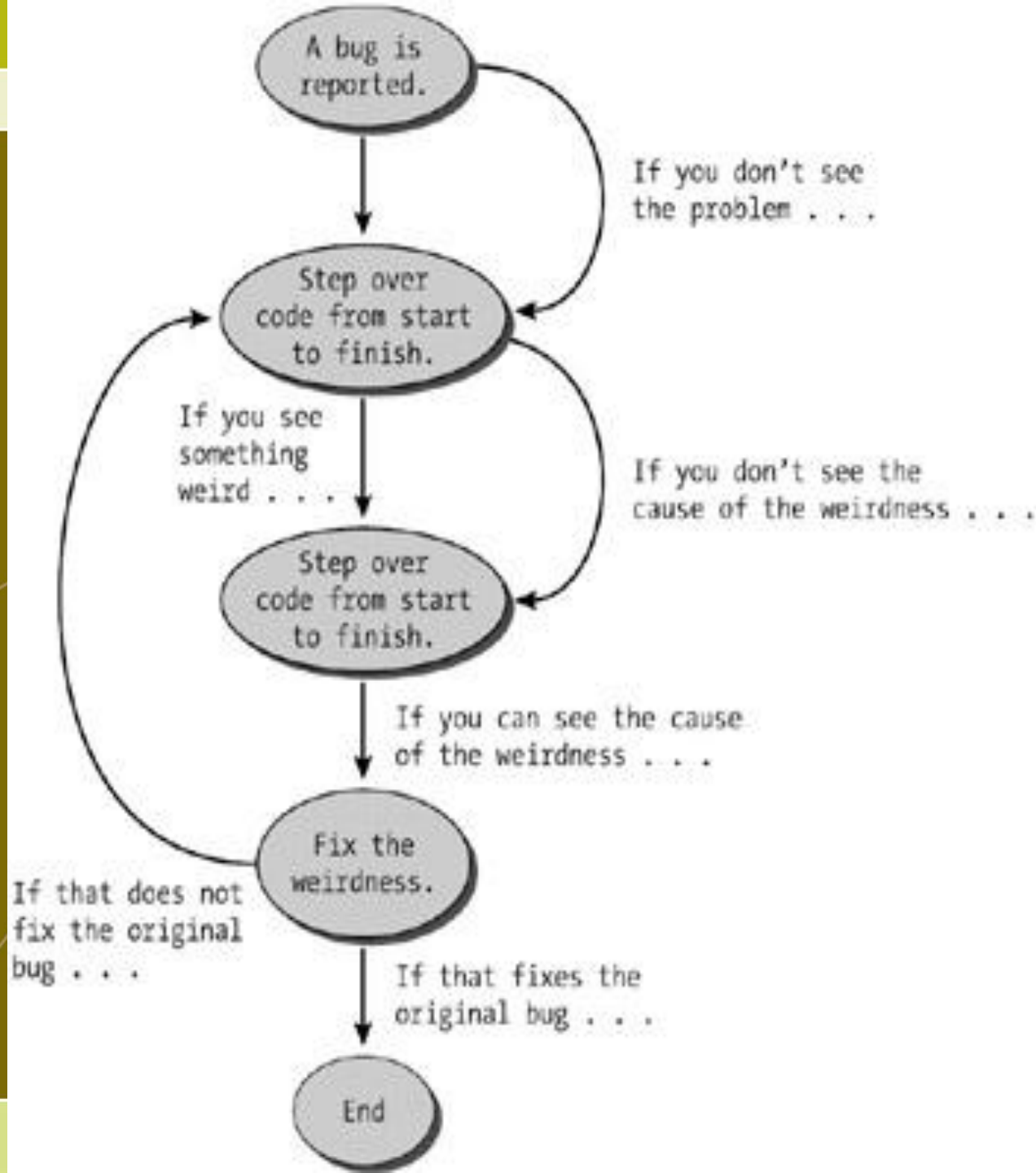
TESTING	DEBUGGING
a) Finding and locating of a defect	a) Fixing that defect
b) Done by Testing Team	b) Done by Development team
c) Intention behind is to find as many defect as possible	c) Intention is to remove those defects

Debugging Approaches

- Brute Force Method
- Back Tracking Method
- Cause Elimination
- Each of the above debugging approaches can be supplemented with debugging tools such as debugging compilers, dynamic debugging aids, automatic test case generators, memory dumps and cross reference maps.

1. Brute Force Method

- most common and least efficient for isolating the cause of a software error.
- a printout of all registers and relevant memory locations is obtained and studied. All dumps should be well documented and retained for possible use on subsequent
- involves using the debugger to step across the code from start to finish until you notice something odd.



2. Back Tracking Method

- It is a quite popular approach of debugging which is used effectively in case of small applications.
- The process starts from the site where a particular symptom gets detected, from there on backward tracing is done across the entire source code till we are able to lay our hands on the site being the cause.
- Unfortunately, as the number of source lines increases, the number of potential backward paths may become unmanageably large.

3. Cause Elimination

- manifested by induction or deduction and introduces the concept of binary partitioning
- data related to the error occurrence are organized to isolate potential causes.
- a "cause hypothesis" is devised and the data are used to prove or disprove the hypothesis. Alternatively, a list of all possible causes is developed and tests are conducted to eliminate each. If initial tests indicate that a particular cause hypothesis shows promise, the data are refined in an attempt to isolate the bug.



Thank you