

Debugging

4010-350
Personal Software Engineering

Don't Blame the Computer!

“All bugs are caused by computers doing exactly what they are told.”

--Terence Parr (Learn the Essentials of Debugging)

“Debugging is twice as hard as writing the code in the first place.

Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it.”

What is Debugging?

- n What's a *bug*? How we ended up having *bugs* in software programs?
- n **Debugging**: the process of identifying the root cause of a problem and fixing it.
- n How is debugging different from, or related to, testing?

How do You Find Bugs?

- n Printing out messages to the screen
- n Reviewing code on paper
- n Asking someone to look at the code
- n Using a debugger
- n Thinking about what the program is doing
- n *Running the program over and over ... insanity*
- n *Making random changes ... hope over experience*

A systematic approach is needed for locating a defect, understanding its cause, and then applying a fix.

Blocks to Effective Debugging

Psychology of Debugging

- n Denial, blaming, finger pointing...
 - Embrace debugging as *problem solving*

ppt#24: Fix the Problem. Not the Blame

- n Panic
 - Panic and thinking don't mix well.
 - Accept there is a problem and start thinking about it.

ppt#25: Don't Panic (and Always Carry a Towel)

Psychology of Debugging

n Blindness (psychological mind set).

```
if ( x < y )  
    swap = x ;  
    x = y ;  
    y = swap ;
```

```
if ( x < y ) {  
    swap = x ;  
    x = y ;  
    y = swap ;  
}
```

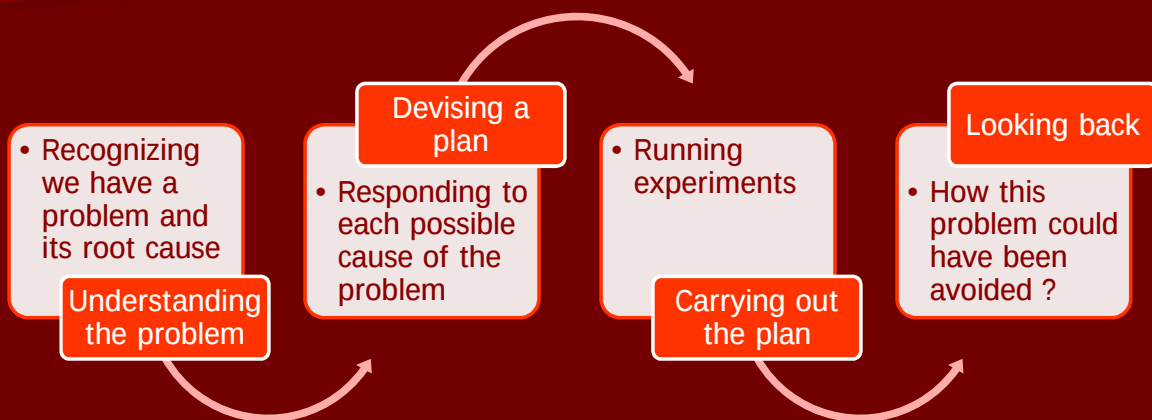
What's wrong with this picture?

Debugging Strategies

Three Ineffective Approaches

- i Find defects by guessing
 - § Scattered print statements
 - § No backup copy of original version
 - § Multiple changes simultaneously
- i Proceed without understanding the problem
 - § The problem is trivial; just fix it!
- i Fix the error with the most obvious fix
 - § Fix a very specific case, instead finding a generic solution

Debugging as Problem Solving (like *House's* medical debugging)



*“Programmers get a couple of books on their first day here. One of them, is **How to Solve It**, by George Polya.”*

Debugging and the Scientific Method

- n Stabilize the error
- n Locate the source of the error
 - Gather data that produces the defect
 - Analyze the data and form hypothesis
 - Determine how to evaluate the hypothesis
 - Confirm or reject the hypothesis
 - *Possibly use a debugger to evaluate the hypothesis.*
- n Fix the defect
- n Test the fix
- n Look for similar errors