Lessons from Bugs

Lawrence Crowl
April 2012
THE COMPUTER ALWAYS WINS!
NOT A BUG,
WORKING AS INTENDED.
The customers' expectations define quality and bugs.
Microsoft Windows 7 Backup
Could not open file.
Diagnostics should be precise and actionable.
Could not connect to network.
Disconnected.
Device busy.
Cannot unmount drive, please wait.
Working as intended.
ANGLE OF GUN?
300 IF A<9 GOTO 303
301 IF A>10 GOTO 303
302 IF D=60 GOTO 500

500 PRINT "HIT"
ATAN
Understand your tools.
BOY SCOUT SURVEY
\text{CNT}(X,Y,Z) = \text{CNT}(X,Y,Z) + 1
IF(X.EQ.3 .AND. Y.EQ.2)
C CNT32(Z)=CNT32(Z)+1
FUNCTION
SUBROUTINE
7 respondents
Make sure you have a problem before you write the program.
VAX/VMS Login
2500 accounts
linear username search
Excess computational complexity is a bug.
compiler for test language
unbalanced binary tree
for symbol table
machine generated identifiers
Programs must handle extreme input.
print file.txt
25% CPU utilization
Open-loop timing is fragile.
humans and robots
68020  ->  SPARC
robots became instant death
120 processors
1 program
occasional failure
improperly handled data race
only found in bench checking
You cannot debug all problems.
Design and review carefully.
if ( a = 0 )
assignment is an expression
implicit conversion of int to bool
unconventional use of operator
The test space for a feature set is much worse than the conventional $O(n^2)$. 
port parallel programming system to the Alliant FX
GCC global register variables
FX caller saves registers
FX system libraries
disassemble bcopy with debugger
change registers used
reassemble, relink, and run
debugger asm != system asm
Fidelity in representation prevents problems.
struct { int field; };
makes up a unique tag name then checks for a redefinition of that tag

... pattern matches against tag to find that it is not really a tag
Elmwood Multiprocessor Operating System
most bugs happened when one person was both implementer and user of an interface
If the interface is important, make sure it is negotiated.
Palm Grafiti auto-capitalization
correcting automatic mistakes is frustrating and slow
If your error rate is low, your program is broken.
Sun mangler and demangler
single YACC source
for
published grammar
and
reference demangler
reference demangler's interpretation is part of the test suite
Automate for consistency.
demangler output ~= C++ source
test from C++ source
to object files
to list of mangled names
to list of demangled names
compared to original source
When possible, close the loop on your tests.
abbreviations needed more testing
too late!

ship now!
bug could not be fixed without breaking compatibility
Never slack off on testing the ABI.
const array
or
array of const
compiler's data structure changed
mangling changed
tests were not deep enough to detect the change
Your product is defined by your tests.
assert( p != NULL );
...

if ( p != NULL )
{
    ...

}
NO!
Asserts encode an understanding of the program.

Changing them requires proof of confusion.
undiagnosed user error
failure to push for the root cause
Sun's root cause field in its bug database was almost never filled out.
Software quality is achieved through socially shared commitment.