Secure Software Development
Risk Based Security Testing

Objectives

- Define Reliability, Resiliency, and Recoverability
- Explain Load Testing and Stress Testing
- Define Means, motive, and opportunity as it pertains to security
- Explain the steps in fuzz testing
- Explain the difference between dumb fuzzing and smart fuzzing
- List advantages and disadvantages of fuzz testing

Definitions

- Reliability

The implication that sw is performing properly for the customer.

- Resiliency

Measure how strong is a system against attacks.

- Recoverability

The sw's ability to be restored after attack

How do we test regular software

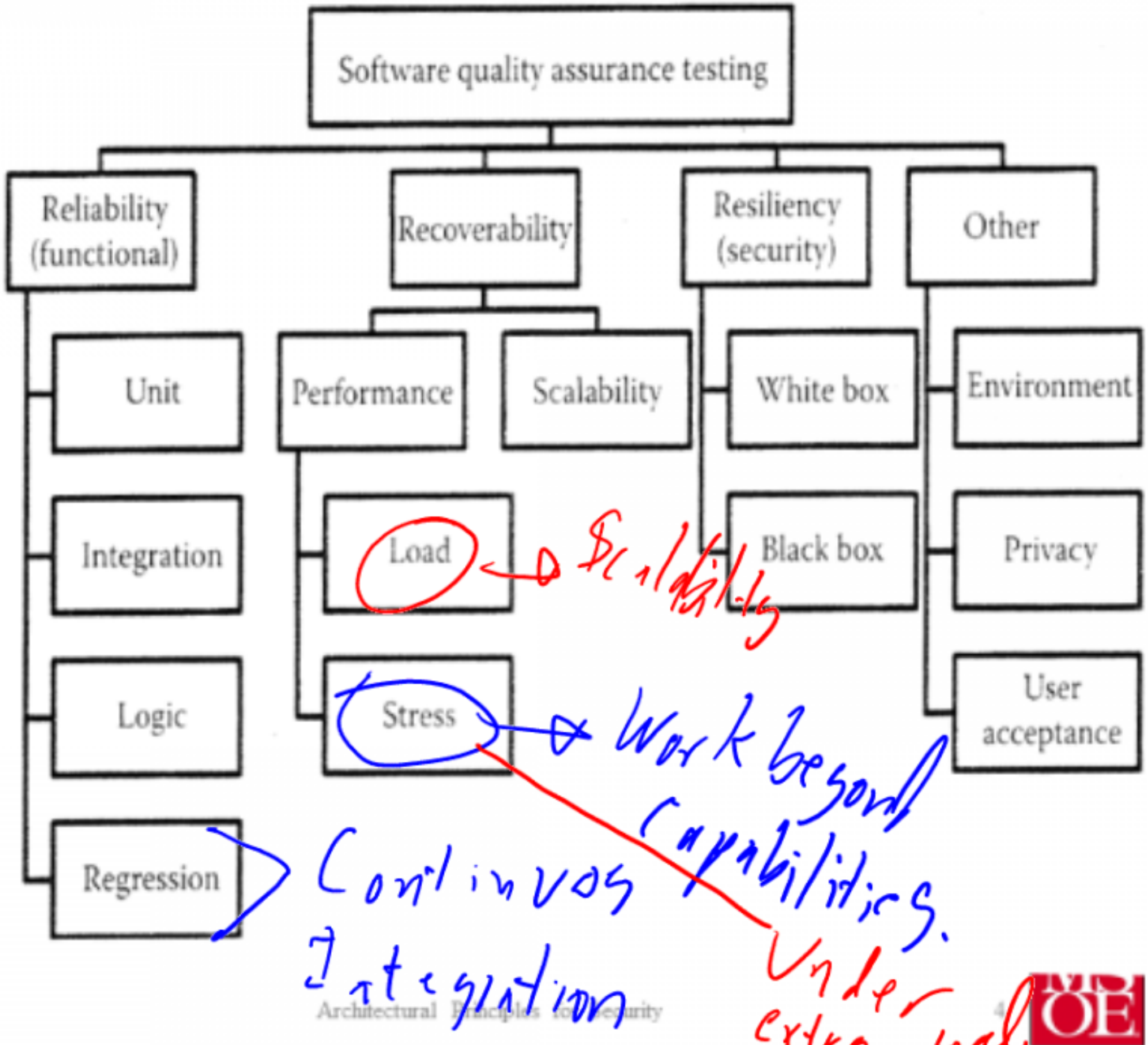
Stupid MOS is only half of testing

System Testing

Integration testing

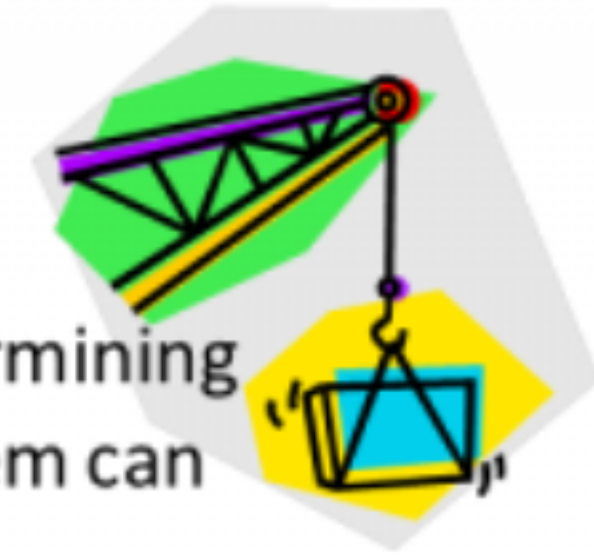
Unit testing

Software Quality Assurance Testing



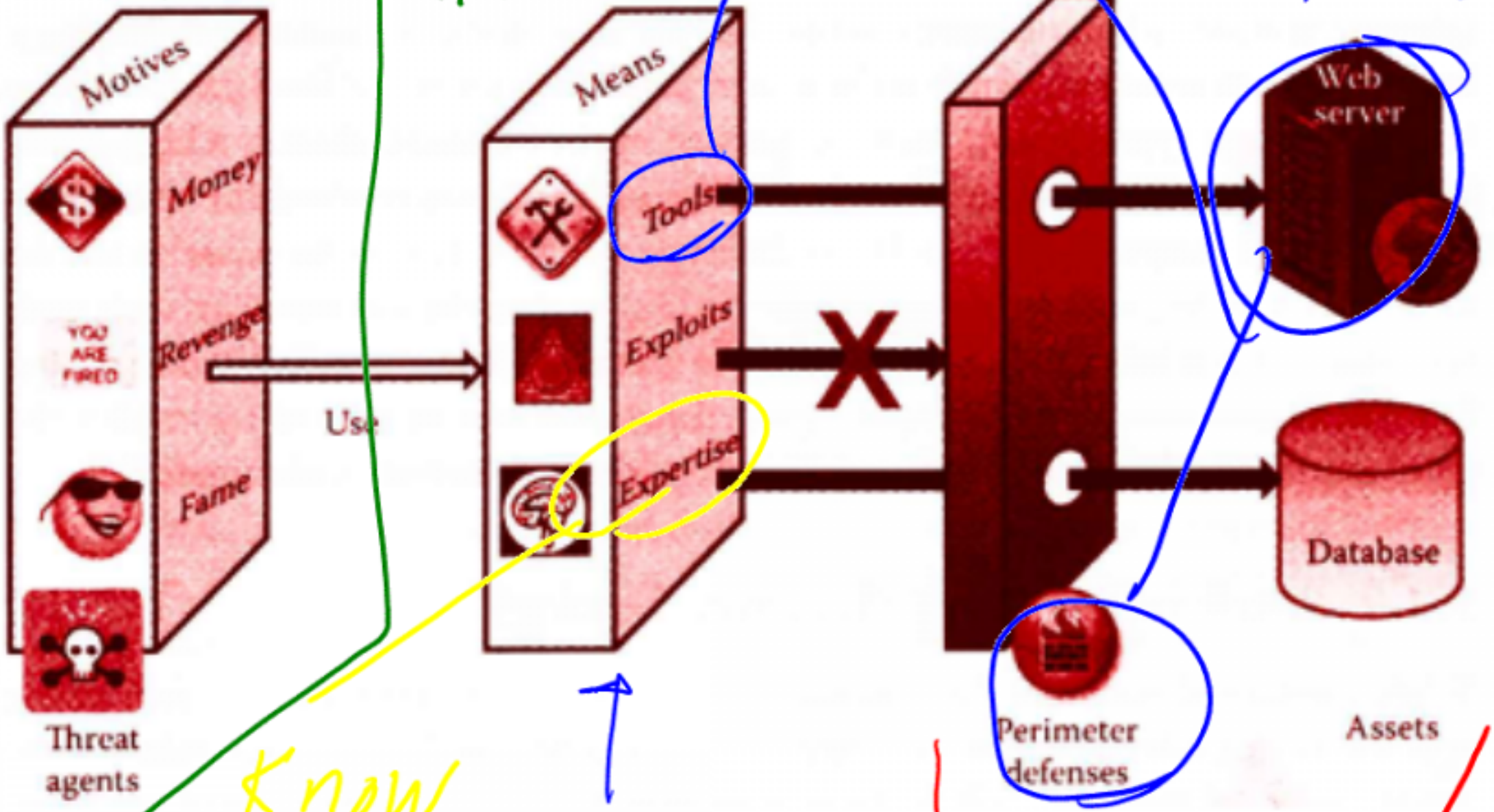
Load and Stress Testing

- Load Testing
  - Load testinjg is the process of determining the maximum load a software system can handle
  - Also known as longevity, endurance, or volume testing



- Stress Testing
  - Goal is to determine the breaking point of software, at which point the software either fails to operate or crashes
  - During stress testing, software is subjected to extreme conditions such as maximum concurrency, limited computing resources, or heavy loads

Motive, Means and Opportunities



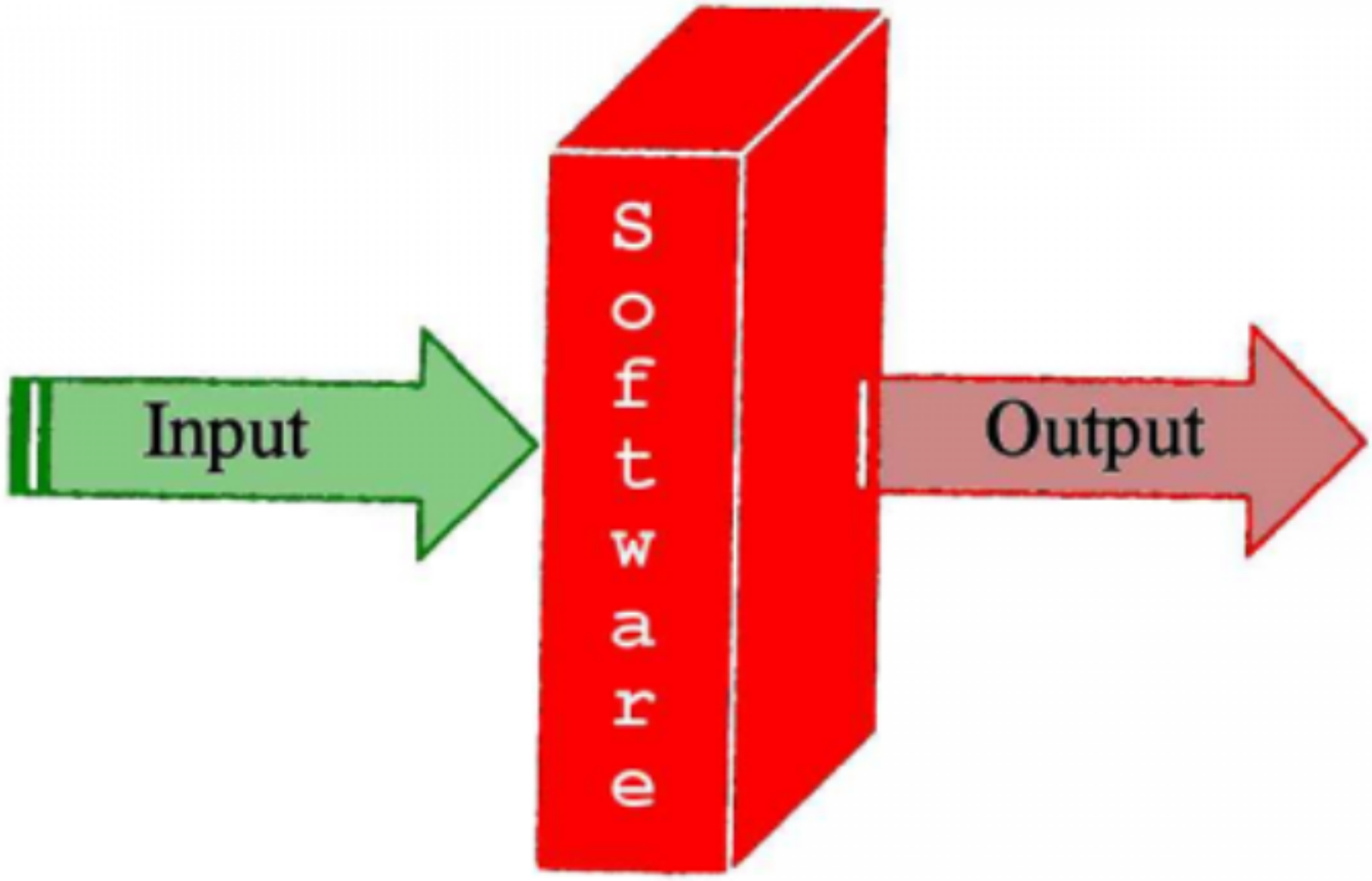
Fuzz Testing

- One night (it was a dark and stormy night) in 1990, Bart Miller (U Wisc.) was logged in over dialup
  - There was a lot of line noise due to the storm
  - His shell and editors kept *crashing*

Fuzz Testing

- Bart Miller et al., "An Empirical Study of the Reliability of UNIX Utilities"
  - Idea: feed "fuzz" (streams of pure randomness, noise from /dev/urandom pretty much) to OS & utility code
    - Watch it break!
    - In 1990, could crash 25-33% of utilities
    - Reports every few years since then
    - Some of the bugs are the same ones in common security exploits (particularly buffer overruns)

Fuzz Testing



- Also known as Fault injection testing
  - Brute force type of testing in which random faults are injected into the software

Fuzz Testing

- Dumb vs. Smart
  - Dumb fuzzing generates data with no regard to the format
  - Smart fuzzing requires knowledge of the data format or how the data is consumed
- Generation vs. Mutation
  - The generation technique creates new files from scratch
  - The mutation technique transforms a sample input file to create a new one
- Most fuzzing tools are a mix of each approach

Example of Fuzz Testing

- Java Calculator
  - Ask the user to input a textual string representing an equation
  - Result will print out on the screen

2+2

2+2 = 4

7 + (2 * 5)

7 + (2 * 5) = 17

.

.

.

Expression Format

expr = term | expr "+" term | expr "-" term

term = factor | term "*" factor | term "/" factor

factor = atom | atom "^" factor

atom = number | "(" expr ")"

Fuzz Testing Steps

- Generate Random Strings
- Pass the string to the calculator
  - One of three things will happen
    1. Return without crashing
    2. Crash
    3. Hang Up

Lets look at Example Java Code

- Note: This is not necessarily the cleanest implementation…

Advantages / Disadvantages of Fuzz Testing

- High volume of testing
- Highly automated
- Finds many problems related to reliability
  - Many of which are potential security holes
  - Over 70% of security vulnerabilities Microsoft patched in 2006 were discovered by fuzzing
- Fuzz testing does not typically validate proper reaction to invalid data
  - "Failures" may be logged as "successes"