



coverity

Detecting Critical Defects on the Developer's Desktop

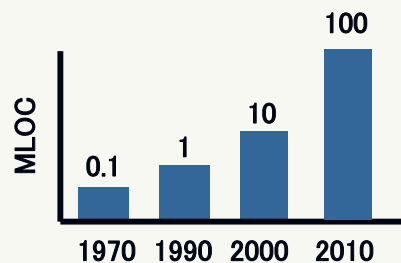
Seth Hallem
CEO
Coverity, Inc.

Copyright © Coverity, Inc. 2006. All Rights Reserved. This publication, in whole or in part, may not be reproduced, stored in a computerized, or other retrieval system or transmitted in any form, or by any means whatsoever without the prior written permission of Coverity, Inc.

Significant Challenge: High Quality Software

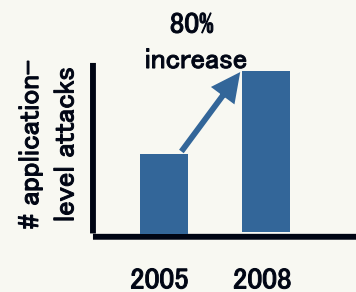
Code is increasingly complex	Code is increasing in size and complexity
The cost of Failure is high	A single defect or security vulnerability can have an enormous impact on the customer
Software bugs are costly	Bugs delay development efforts and impact new feature development

Exponential LOC growth in typical GM car



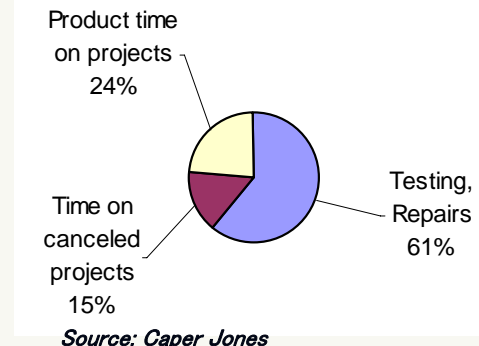
Source: Tony Scott CIO, GM

Application-level security attacks on the rise



Source: Gartner

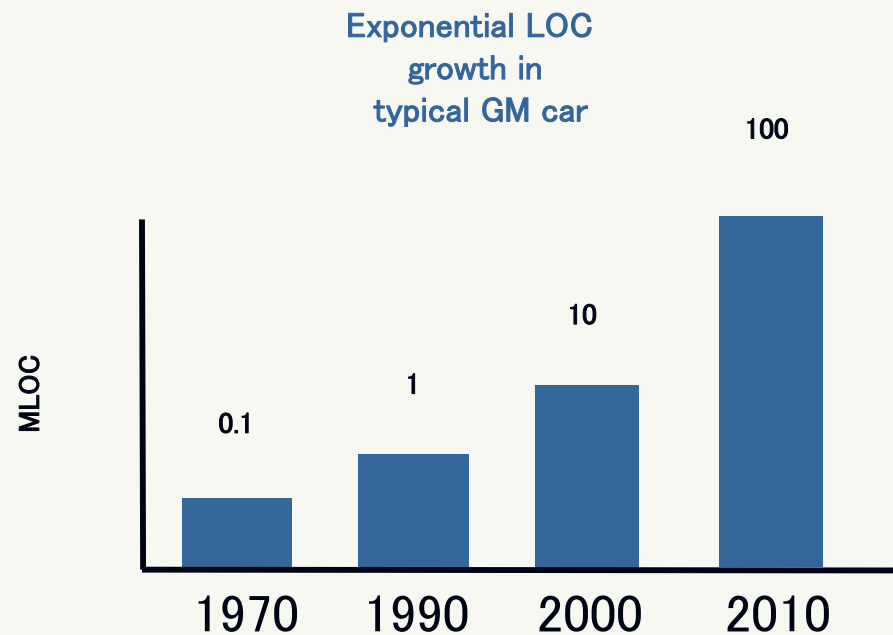
Developers spend significant time testing & fixing bugs



Source: Caper Jones

Software Complexity is Rising

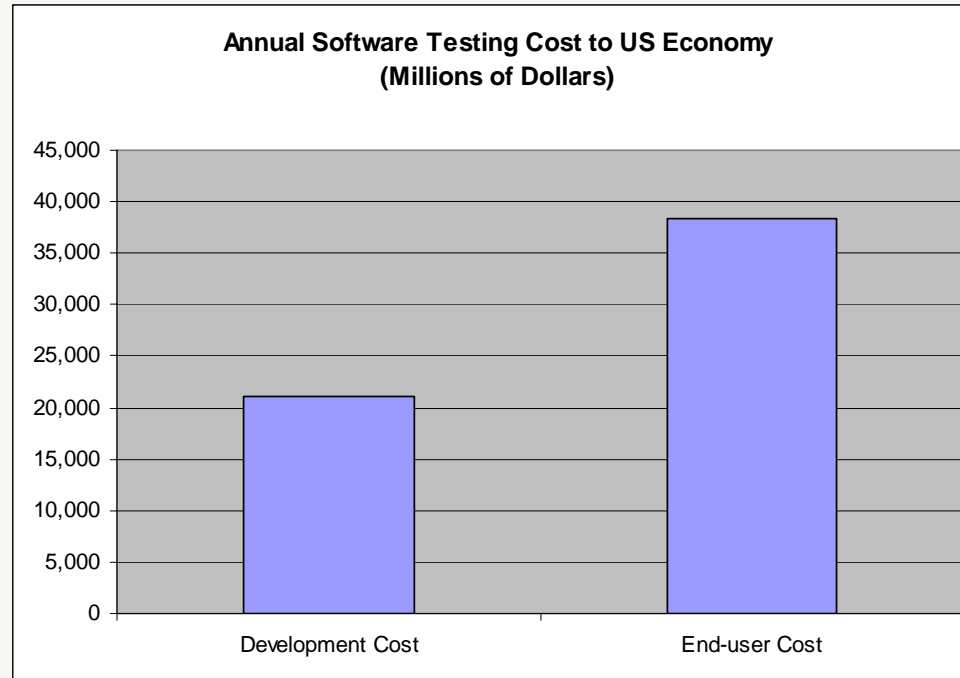
- By 2010, cars will have 100 million lines of code



Source: Tony Scott CIO, GM

Rising Cost

- The cost of inadequate software testing is rising
- In the United States:
 - The annual cost to software developers is over 22 billion dollars
 - The annual cost to end-users is over 35 billion dollars



NIST Planning Report 02-3. May, 2002.

Traditional Challenges in Static Analysis

Software Development Process



Static Analysis

```

q = __sigqueue_alloc(v, GFP_ATOMIC);
if (q) {
    list_add_tail(&q->list, &signals->list);
    switch ((unsigned long) info) {
    case 0:
        q->info.si_signo = sig;
        q->info.si_errno = 0;
        q->info.si_code = SI_USER;
        q->info.si_pid = current->pid;
        q->info.si_uid = current->uid;
        break;
    case 1:
        q->info.si_signo = sig;
        q->info.si_errno = 0;
        q->info.si_code = SI_KERNEL;
        q->info.si_pid = 0;
        q->info.si_uid = 0;
        break;
    default:
        copy_siginfo(&q->info, info);
        break;
    }
} else {
    if (sig >= SIGRTMIN && info && (unsigned long)info != 1
        && info->si_code != SI_USER)
        /*
         * Queue overflow, abort. We may abort if the sig
         * and sent by user using something other than kill().
         */
        return -EAGAIN;
    if (((unsigned long)info > 1) && (info->si_code == SI
        /*
         * Set up a return to indicate that we dropped
         * the signal.
         */
        ret = info->si_sys_private;
}
}
out_ret:
sigaddset(&signals->signal, sig);
return ret;
}
#define LEGACY_QUEUE(sigptr, sig) \
(((sig) < SIGRTMIN) && sigismember(&(sigptr)->signal, (sig)))
  
```

Warnings

False
Positives

TRADITIONAL FAILURES

High Cost
Of Ownership

- Hard to integrate
- Significant configuration & tuning
- Does not scale

Poor Results

- Partial code path coverage
- Shallow analysis
- Uninteresting results
- Rife with False Positives

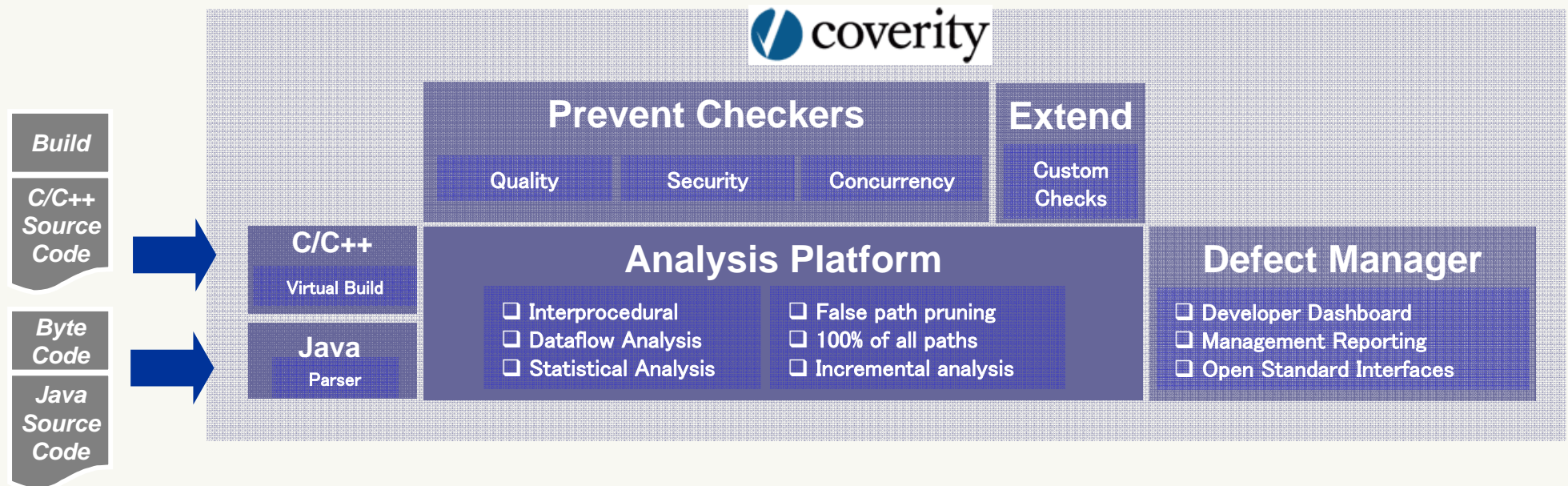
Coverity: Breakthrough Technology

Breakthrough Research
At Stanford University
Computer Systems Lab



Analysis Depth	<ul style="list-style-type: none"> • 100% of all code paths • Interprocedural analysis
Analysis Accuracy	<ul style="list-style-type: none"> • 20% false positive rate
Scalability	<ul style="list-style-type: none"> • Millions of lines of code

Coverity: Core Technologies



- Uses innovative source code analysis algorithms originating from compiler research
- Performs a whole program analysis
- Integrates easily into the software development process
- Integrated database application enables complete workflow and reporting

Coverity: Core Features

What defects can it find?

- Security Vulnerabilities
- System and Process Crashes
- Infinite Loops
- Performance Degradations
- Denial of Service
- Privilege Escalation
- Data, Memory and File Corruption
- Unpredictable Behavior
- Concurrency issues

How does it work?

- Do not run the code
- Zero test cases
- Runs at compile time

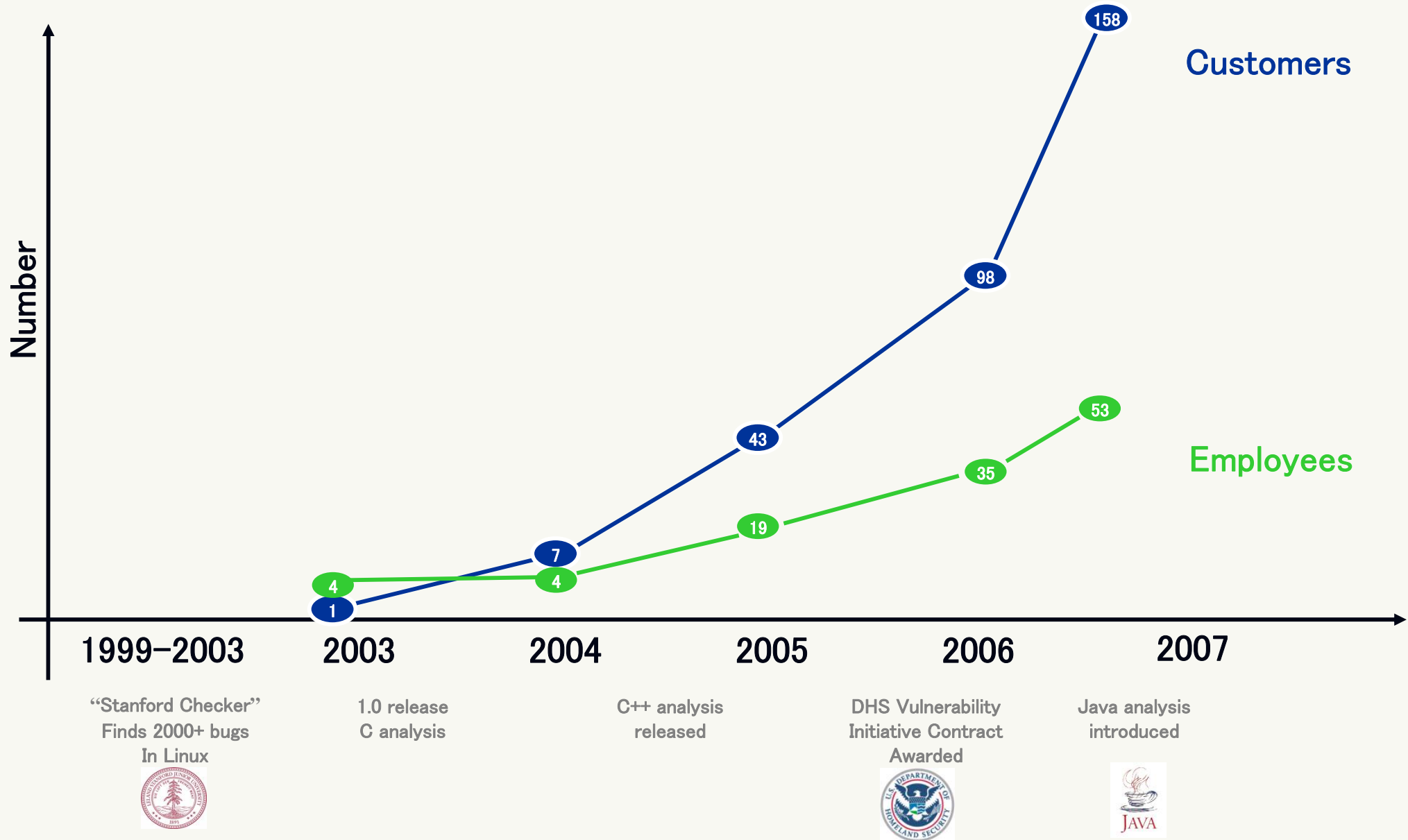
Coverity: Market Leader

Accuracy	Finds the most valuable flaws in your software
Integration	Minimal impact on the development process
False Positives	Avoids reporting costly noise
Likelihood of use	Built for developers to use and appreciate

Sample of Coverity Customers



Coverity History



Customer Success: Wall Street Journal

Tech Companies Check Software Earlier for Flaws

By Thomas H. Davenport

WHEN BLACKBERRY tested Research In Motion Ltd. development software in the past, its engineers worked quickly to spot problems, sometimes overlooking bugs that were caught later in the process. The result: when issues cropped up after a program had been built, it took business days and even weeks to have the code fixed.

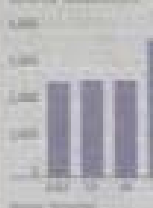
RIM won't share, but its engineers pointed to how teams with new software, and checking for bugs that could later be reported by hackers who often view it as a waste of time. That has begun to change in the past few years as new laws have the disclosure of security holes and reactions, and some

positive attention is being paid to software through the Wall Street Journal's "Security" column. Some firms, including Intel, are testing programmers to write code and test their security as software is built, not afterward.

While the Black Berry had engaged with out-sourcing for security tests, Steve Little, a RIM security director, wanted the company to test its code more often. He said the company hadn't paid enough attention to the software that runs on the BlackBerry and other devices.

Finding Flaws

Software bugs are often caught earlier in the development process.



"The idea was that we could be doing more," says Mr. Little, who is based at RIM's Waterloo, Ontario, headquarters. "We had to raise the bar."

Mr. Little soon discovered Coverity Inc., a San Francisco startup that sells tools to automatically check for software flaws. Now Mr. Little uses Coverity every night to scan the code written by its engineers. The tool sends Mr. Little an email for any potential red flags, the figures and which problems are real and which ones are just offending programmers, who has to fix the flaws before moving on. Mr. Little has also reduced his security training and requires programmers to double-check each other's code more regularly.

Software vulnerabilities throughout the industry have been on the rise. In February, for example, the U.S. Computer Emergency Response Team, a government organization, pointed out a flaw in Apple Computer Inc.'s Safari Web browser that could allow a hacker to take control of a computer by persuading a user to view a specially crafted Web page. Overall, Symantec Corp., a Cupertino, Calif., maker of security software, found 2,100 vulnerabilities in software

Photo: Steve Delaney for The Wall Street Journal



“Many companies, including RIM, are teaching programmers to write safer code and test their security as software is built, not afterward.”

Coverity Success: Wall Street Journal

Tech Companies Check Software Earlier for Flaws

By Thomas H. Davenport

WHEN BLACKBERRY mobile devices in Mexico City developed software in the past, IT engineers worked quickly to meet deadlines, sometimes overlooking bugs that would surface later in the process. The result: when issues cropped up after a program had been built, it took business time and energy to locate the errors.

RIM won't share, but its executives pointed to how teams with few software and checking for bugs that could later be reported by hackers who often view as a waste of time. That has begun to change in the past few years as and how have the discovery of security holes and reactions, and how

positive technology is being used to improve through the Wall Street Journal's "Finding Flaws" column. Some firms, however, are not using the tool, and testing programmers to generate code and test their security as software is built, not afterward.

While the Black Berry had engaged with out-sourcing for security tests, Steve Little, a RIM security director, admitted the company hadn't paid enough attention to the software that runs on the BlackBerry and other devices.

"The idea was that we could be doing more," says Mr. Little, who is based at RIM's Waterloo, the firm's headquarters. "We had to make the test."

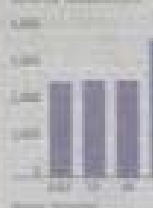
Mr. Little soon discovered Coverity Inc., a San Francisco startup that helps firms to automatically check for software flaws. Now Mr. Little uses Coverity every night to scan the code turned in by engineers. The tool sends Mr. Little an email listing potential red flags, the figures and which problems are real and which ones are just offending programmers, who has to fix the flaws before moving on. Mr. Little has also reduced an security training and requires programmers to double-check each other's code more regularly.

Software vulnerabilities throughout the industry have been on the rise. In February, for example, the U.S. Computer Emergency Response Team, a professional organization, pointed out a flaw in Apple Computer Inc.'s Safari Web browser that could allow a hacker to take control of a computer by persuading a user to view a specially crafted Web page. Overall, Symantec Corp., a Cupertino, Calif., maker of security software, found a 20% increase in software

Photo: Steve Delaney for The Wall Street Journal

Finding Flaws

Software vulnerabilities found by engineers using the Coverity tool.



“Many companies, including RIM, are

“Now, Mr. Little uses Coverity every night to scan the code turned in by engineers. The tool sends Mr. Little an email listing red flags.”

– WSJ 05/04/06

Coverity Success: **WIND RIVER**

- Quality improvement is top priority designated by executive management
- Complex requirements for development tools:
 - Had to fit into the existing infrastructure
 - Had to fit into the Capability Maturity Model (CMM)
- According to WindRiver's Director of Engineering:
 - "We compared and evaluated a number of programming and error detection tools and Coverity was superior."

Coverity Success: **WIND RIVER**

- Ease of integration was critical
 - “integration with Coverity Prevent is seamless and the usage is straightforward. We went from trial to purchase in 3 weeks.”
- Coverity’s impact:
 - Immediate
 - “We found several important defects. It does validate the purchase of the tool.”
 - Ongoing
 - Development productivity up 30%
 - Time to market cut by 20%