

# Managing Software Integrity in an Agile World

Behrooz Zahiri  
Coverity



- The Software Integrity Crisis
- What is Agile?
- How Static Analysis Improves Software Integrity
- About Coverity

---

---

# The Software Integrity Crisis

---

---



# Software vulnerability is everywhere!



**REUTERS** EDITION: U.S. News Sectors Industries Analysis & Opinion

**Michael Krigsman**  
Get IT Project Failures via: Mobile RSS Email Alerts Show: Michael's Bio  
Pick a blog category: [dropdown] View

## February 21st, 2008 System update kicks Heathrow baggage system offline

Posted by Michael Krigsman @ 7:32 pm  
Categories: [Project failures](#), [Government contracts](#), [IT issues](#), [CIO issues](#)  
Tags: [Business](#), [Information Technology](#), [System Update](#), [Tools & Techniques](#), [Software](#), [Upgrade](#), [Transportation](#), [System Management](#), [Internet](#), [Software](#), [Software](#)

### Essential Topics

- Green IT Center
  - ESG Report: Emerging Green Initiatives and needs in IT
  - Watch The Webcast: Going Green
  - White Paper: Mastering carbon management. Balancing Trade-Offs To Optimize Supply Chain Efficiencies.

SPONSORED BY **IBM**

### The HOT Spot

Printers

- 'Green' Font Cuts Costs and Saves Trees (BNET)
- Three Ways to Save Paper (BNET)
- CNET Reviews printer buying guide (CNET)

Non-profits don't have to provide services. It could be a start.

Learn more >



HOME INVESTING COMPANIES TECHNOLOGY INNOVATION MANAGING SMALL BIZ B-SCHOOLS ASIA

Current Issue Past Issues Cover Story Podcasts Figures of the Week Small Biz Mag

COVER STORY April 10, 2008, 5:00PM EST

## The New E-spying Threat

A BusinessWeek probe of rising attacks on America's most networks uncovers startling security gaps



The com  
4 was li  
result, c  
6000 pa  
A terse j

**SFGate**  
home of the **San Francisco Chronicle**  
Home Delivery | Today's Paper | Ads

Find a buyer for your TV. Sell stuff on Kaango.  
FREE ads and photos reaching millions of SFGate visitors.

Sell Stuff Now

SEARCH SFGate Web Search by YAHOO! Advanced Search Sign In Register

Home News Sports Business Entertainment Food Living Travel Columns Classifieds Jobs Homes Cars Index

Bay Area & State | Nation | World | Politics | Crime | Tech | Obituaries | Education | Environment | Science | Health | Weird

## BART'S BLUNDER

### 35,000 evening commuters left in lurch -- system takes blame for software snafu

Simone Sebastian, Carl T. Hall, Cicero A. Estrella, Chronicle Staff Writers  
Thursday, March 30, 2006

PRINT EMAIL SHARE COMMENTS FONT SIZE TOOLS SPONSOR: verizon

Embarrassed BART officials are taking the blame for a computer crash that stranded 35,000 commuters for more than an hour at the height of the Wednesday evening commute.

your couch.  
Sell stuff on Kaango.

Sell Stuff Now

FREE ads and photos reaching millions of SFGate visitors.

MOST COMMENTED MOST READ MOST E-MAILED

## CORRECTED - UPDATE 4-Software problem causes US airline flight delays

(Corrects previous glitch in August 2008; AirTran based in Orlando, not Atlanta)

Thu Nov 19, 2009 6:12pm EST

STOCKS | INDUSTRIALS

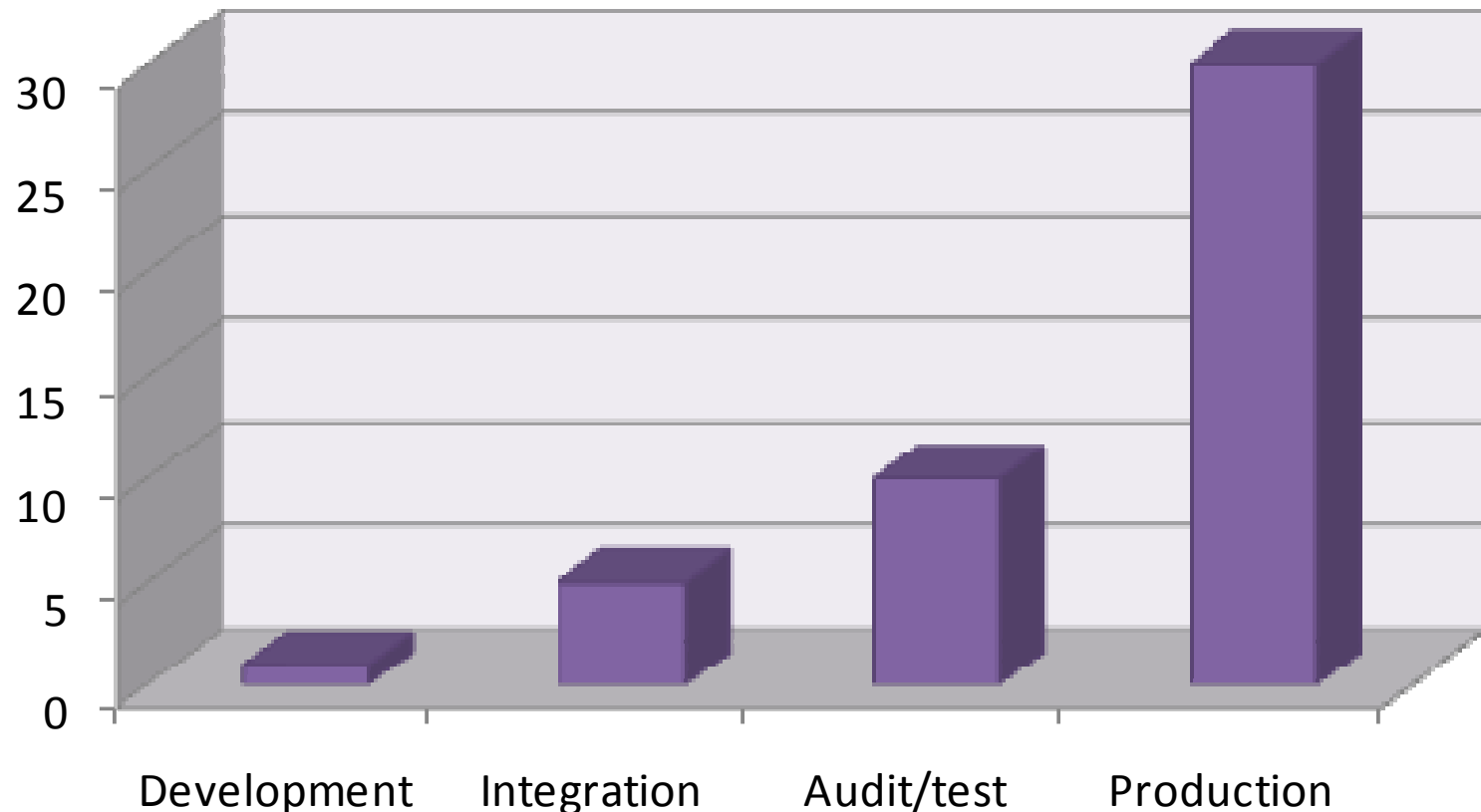
Related News  
UPDATE 2-US

\* FAA says fault linked to software problem

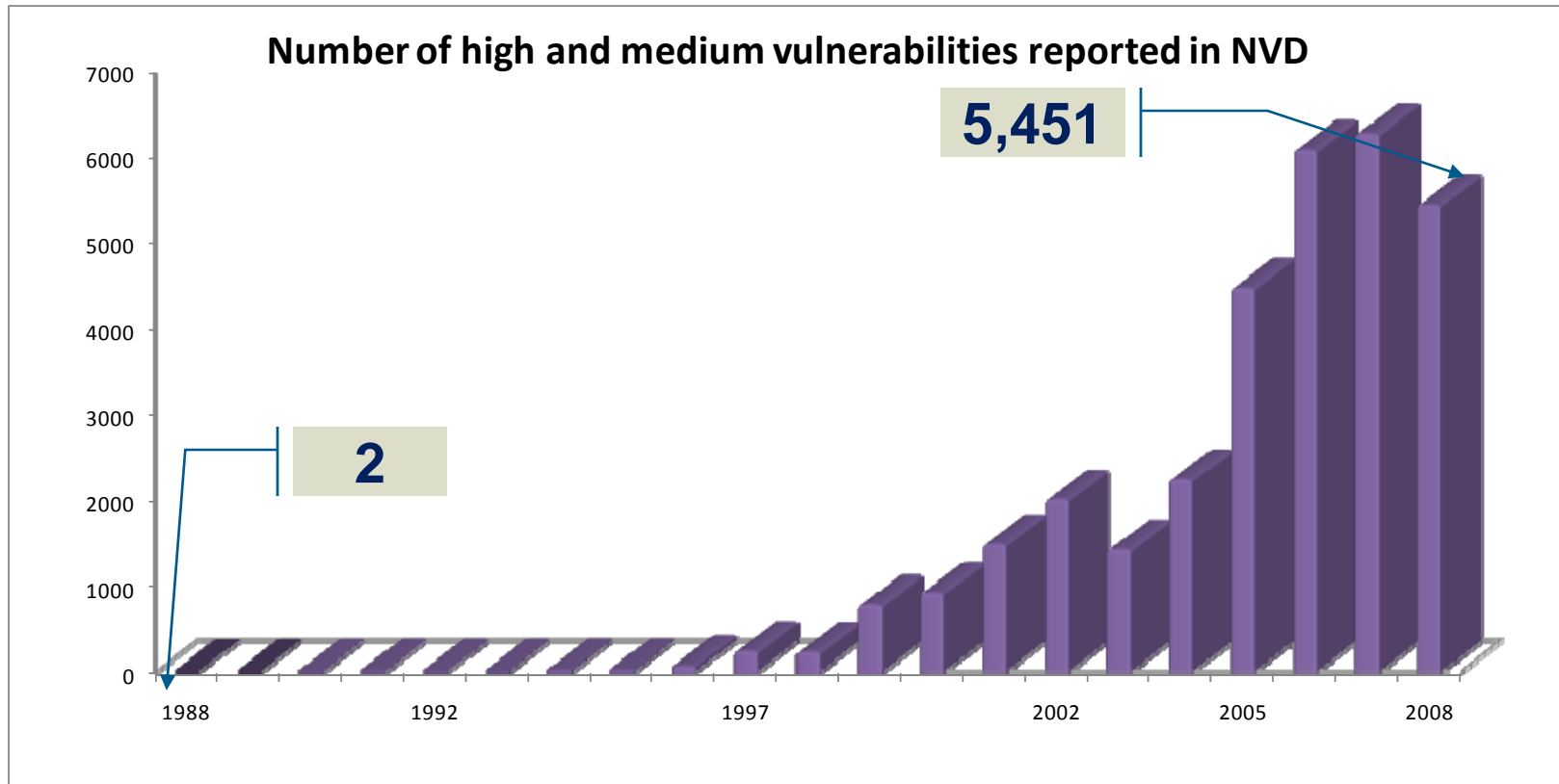
The earlier you find a defect...  
...the cheaper it is to fix



## Cost for defect fixes



# Yet more defects are released in production each year



Source: NIST

## Bloomberg.com

▶ [BloombergAnywhere](#) ▶ [BloombergProfessional](#) ▶ [AboutBloomberg](#)

Updated: New York, Dec 14 17:34 London, Dec 14 22:34 Tokyo, Dec 15 07:34

QUOTE SEARCH NEWS SYMBOL LOOKUP

FEEDBACK LOG IN/REGISTER



Bringing broadband to all of America can create 1.2 million new jobs.

Help connect everyone in America.

ROLL OVER TO SIGN THE PETITION



HOME NEWS MARKET DATA PERSONAL FINANCE TV and RADIO BUSINESSWEEK BUSINESS EXCHANGE

Bloomberg Innovators Technology Currencies Forex Trading Videos ETFs CEO Commodities

### news

- Exclusive
- Worldwide
- Regions**
  - Africa
  - Asia
  - Australia & New Zealand
  - Canada**
  - China
  - Eastern Europe
  - Europe
  - France
  - Germany
  - India & Pakistan
  - Italy
  - Japan
  - Latin America
  - Middle East
  - U.K. & Ireland

## Bombardier Says Software Flaw Causes CRJ1000 Delay (Update2)

Share [BX](#) [t](#) [f](#) | [Email](#) | [Print](#) | [A A A](#)

By Will Daley

Dec. 3 (Bloomberg) -- **Bombardier Inc.**, the world's third-largest maker of commercial planes, said a software flaw grounded flight testing of the CRJ1000 regional jet, pushing back the aircraft's introduction.

Tests will resume after Christmas, and the plane will begin service in the second half of the fiscal year starting in February, said **Guy Hachey**, president and chief operating officer of the Montreal-based company's aerospace unit. Deliveries of the CRJ1000 were set to begin in the year's first quarter, **Isabelle Rondeau**, a spokeswoman, said in an e-mail.

Bombardier earlier this year found a software problem on the 100-seat aircraft, thought it had found the cause and then experienced a similar problem a month and a half later, Hachey said today on a conference call. The company has determined the cause and will make changes to remedy it, he said.

Bringing broadband to all of America can create 1.2 million new jobs.

Help connect everyone in America.

ROLL OVER TO SIGN THE PETITION

More News

- [Canadian Stocks Rise as Gas Producers Rally on Exxon's Takeover of XTO](#)

# Why software integrity is so important



## Increased Complexity

- Codebases are growing
- Increased tension between development and QA; wasted cycles trying to reproduce bugs

## Increased Mission Criticality

- Cost of failure is increasing, increasing the visibility and scrutiny of testing and defect management
- High-profile firefights on the rise

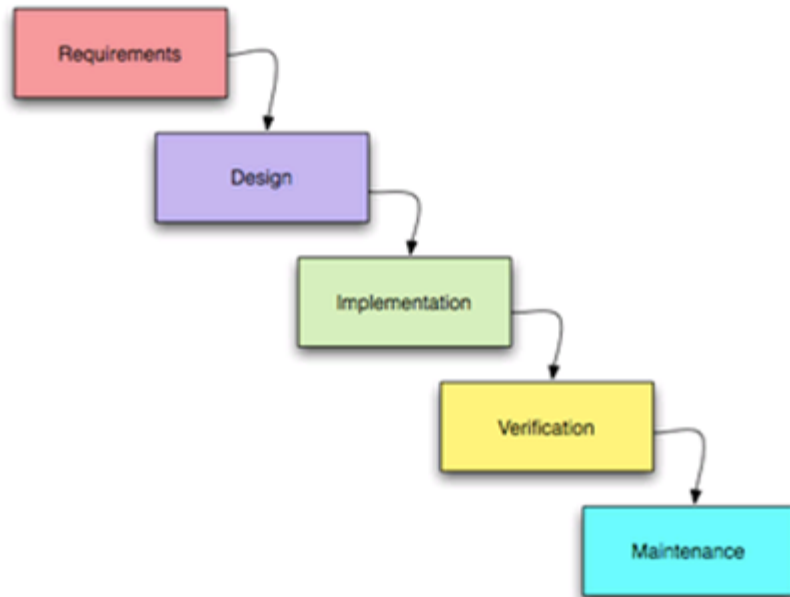
## Increased Time to Market Pressure

- Development teams are under pressure to release higher quality software even faster
- Traditional development methodologies can't keep pace

# What is Agile?

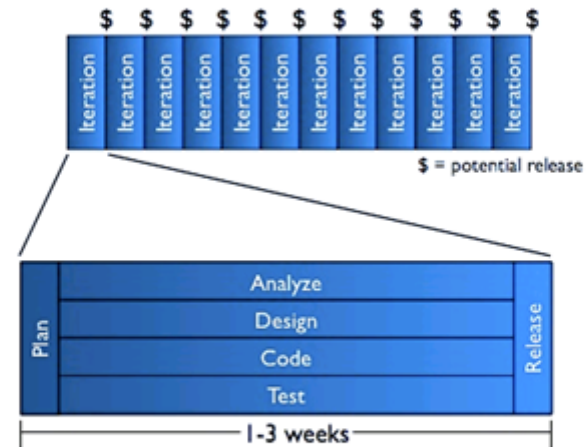
# How is Agile development different?

## Waterfall

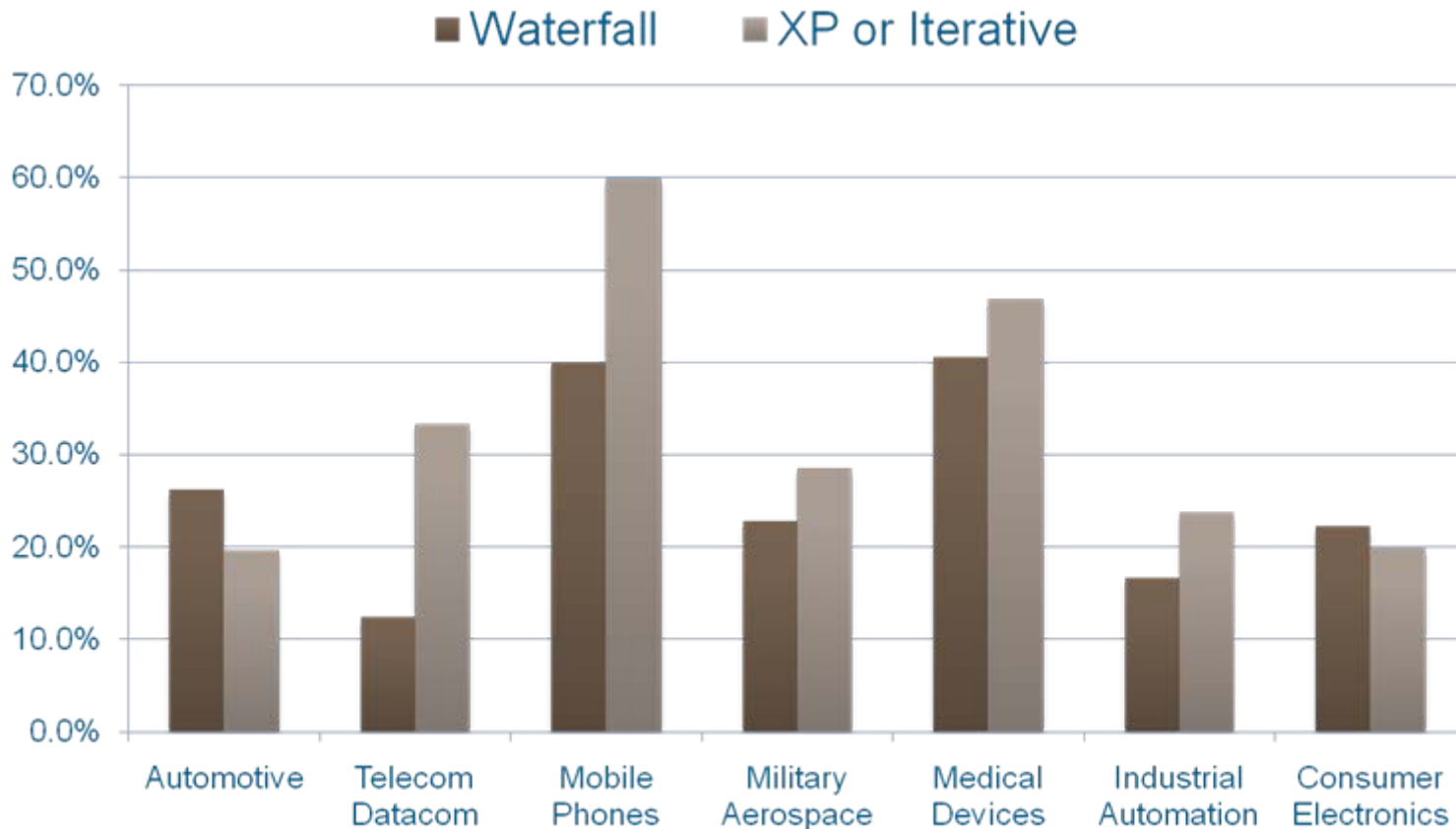


## Agile (Scrum, XP, Lean, etc.)

### The XP Lifecycle



# Agile is becoming more popular



# Agile requires software integrity to become everyone's responsibility!



## Benefits of Agile

- Increased visibility
- Increased flexibility

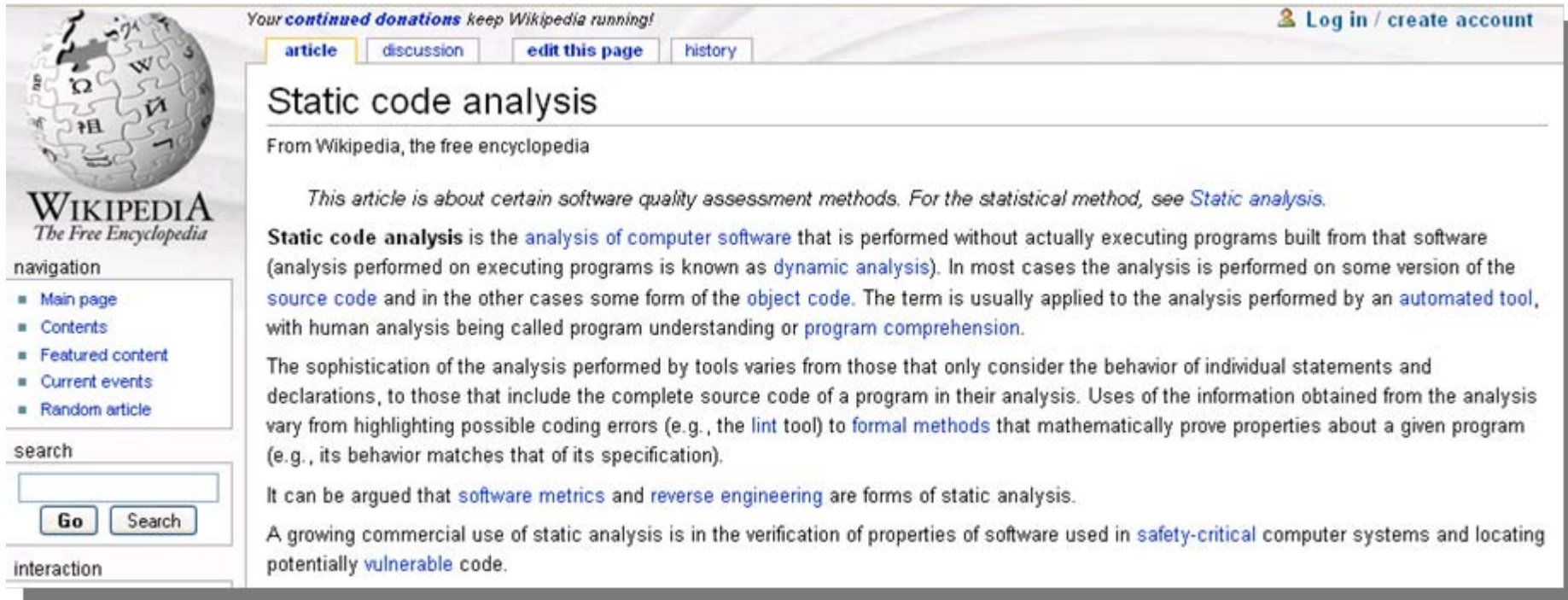
## Risks to Software Integrity

- Frequent “potentially shippable” iterations, each requiring a QA cycle
- Shortened QA cycles – QA can't keep up!
- Increased pressure on full development team to deliver high-integrity code

# How Static Analysis Improves Software Integrity



# What is static analysis?



The screenshot shows the Wikipedia article for "Static code analysis". At the top left is the Wikipedia logo, a globe made of puzzle pieces. Below it is the text "WIKIPEDIA The Free Encyclopedia". To the right of the logo is a navigation menu with links for "Main page", "Contents", "Featured content", "Current events", and "Random article". Below the navigation menu is a search box with "Go" and "Search" buttons. At the top right of the article page, there is a "Log in / create account" link. The article title "Static code analysis" is prominently displayed, followed by the text "From Wikipedia, the free encyclopedia". Below the title is a summary paragraph: "This article is about certain software quality assessment methods. For the statistical method, see *Static analysis*." The main body of the article starts with a definition: "Static code analysis is the analysis of computer software that is performed without actually executing programs built from that software (analysis performed on executing programs is known as dynamic analysis). In most cases the analysis is performed on some version of the source code and in the other cases some form of the object code. The term is usually applied to the analysis performed by an automated tool, with human analysis being called program understanding or program comprehension." The article then discusses the sophistication of the analysis performed by tools, ranging from individual statements and declarations to complete source code analysis. It mentions uses like highlighting coding errors (e.g., the lint tool) and formal methods that mathematically prove properties about a given program (e.g., its behavior matches that of its specification). The article concludes by stating that it can be argued that software metrics and reverse engineering are forms of static analysis, and that a growing commercial use of static analysis is in the verification of properties of software used in safety-critical computer systems and locating potentially vulnerable code.

Think of it as a spell checker that finds your most difficult bugs!

- Nullable references were invented by [Hoare](#) in 1965 as part of the [Algol W](#) language.
- Hoare later (2009) described his invention as a 'billion dollar mistake':[\[4\]\[5\]](#)

“ I call it my billion-dollar mistake. At that time, I was designing the first comprehensive type system for references in an object oriented language ([ALGOL W](#)). My goal was to ensure that all use of references should be absolutely safe, with checking performed automatically by the compiler. But I couldn't resist the temptation to put in a null reference, simply because it was so easy to implement. This has led to innumerable errors, vulnerabilities, and system crashes, which have probably caused a billion dollars of pain and damage in the last forty years”

- Since a null-valued pointer does not refer to a meaningful object, an attempt to dereference a null pointer usually causes a run-time error.

# Static analysis strengthens your existing testing approaches!



Existing Testing	Static Analysis
<ul style="list-style-type: none"><li>▪ Unit Tests</li><li>▪ Integration and Build Tests</li><li>▪ Standard Compliance Tests</li><li>▪ Performance Tests</li><li>▪ Code Reviews</li></ul>	<ul style="list-style-type: none"><li>▪ Automatic Tests</li><li>▪ Exhaustive Tests</li><li>▪ Concurrency Tests</li><li>▪ Resource Leakage</li><li>▪ Security Tests</li><li>▪ Library and API misuse</li><li>▪ ....</li></ul>

# Will a code review catch this?



```
#include <pthread.h>
pthread_mutex_t mtx;
int i;
void foo() {
    i++;
    if (i == 1) {
        pthread_mutex_lock(&mtx);
        i++;
        pthread_mutex_unlock(&mtx);
    }
}
```

i not initialized and i++ not locked

# What kind of defects might be in *your* software?

- Improper Locking/Concurrency
- NULL Pointer Dereference
- Resource Leak
- Unintentional Ignored Expressions
- Use Before Test (NULL)
- Buffer Overrun (statically allocated)
- Unsafe use of Returned NULL

- Use After Free
- Uninitialized Values Read
- Unsafe use of Returned Negative
- Type and Allocation Size Mismatch
- Buffer Overrun (dynamically allocated)
- SQL Injection
- Use Before Test (negative)
- And more!



# Benefits of static analysis in Agile environments

- **Finds defects sooner** - as soon as they're created, when they're cheapest to fix
- **Empowers development teams** to remain nimble and rapidly deliver quality software
- **Strengthens** existing testing methodologies with an objective lens



# About Coverity



Companies that have zero tolerance for software failures choose Coverity.

## High Integrity Software



Software that doesn't fail.



Software that performs.



Software that is secure.

**900 Customers Worldwide**

# Over 900 customers. Billions of lines of code.



# Steps To Mitigate Risk

```
long int SomeFunction();  
/* int OtherFunction();  
/* int */ CallingFunction(),  
{  
    long int test1;  
    register /* int */ test2;  
  
    test1 = SomeFunction();  
    if (test1 > 0)  
        test2 = 0;  
    else  
        test2 = OtherFunction();  
  
    return test2;  
}
```



Scan  
your  
software

## List of Defects

- \_10001 **critical**
- \_10002 **major**
- \_10003 **major**
- \_10004 **critical**
- \_10005 **major**

Find  
priority  
defects

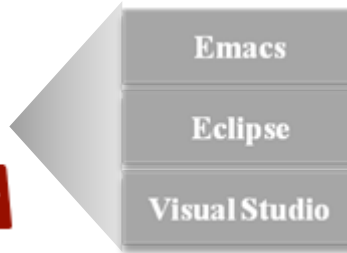
## Browse code

```
long int SomeFunction();  
/* int OtherFunction(); */  
/* int */ CallingFunction()  
  
long int test1;  
register /* int */ test2;  
  
meFunction();  
> 0)  
2 = 0;
```

Impact Rankings



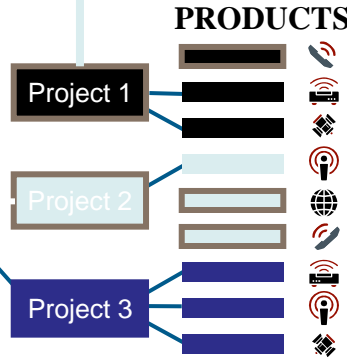
Fix  
priority  
defects

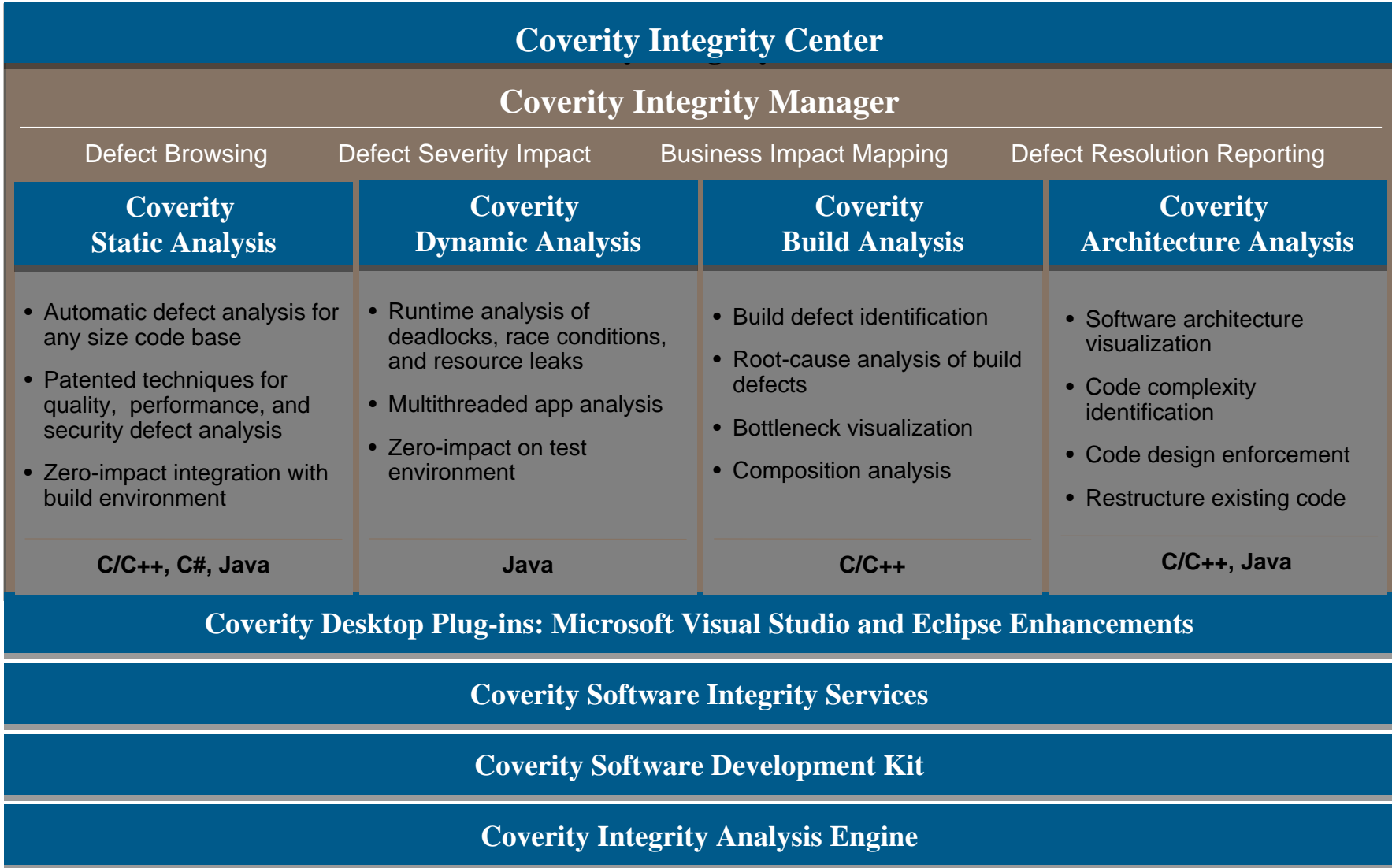


Report  
defect  
remediation

Map  
business  
impact

Code  
base





# Request a free trial today!



Please visit us at [www.coverity.com](http://www.coverity.com)  
to request a free trial

The screenshot shows the Coverity website's 'Request a Free Trial' page for the Integrity Center. At the top, there is a navigation bar with the Coverity logo, a phone number (1-415-321-5200 or contact sales), a search bar, and a 'Request A Free Trial' button. Below the navigation bar, there is a circular graphic with the Coverity logo in the center and the text 'Integrity Center' below it. The circular graphic is surrounded by four segments: 'Architectural Analysis', 'Dynamic Analysis', 'Build Analysis', and 'Static Code Analysis'. To the right of the graphic, the text reads 'Coverity Integrity Center' followed by 'Improve Your Source Code Quality and Security'. Below this, three bullet points list the benefits: 'Enhance Application Quality and Security', 'Lower Development Costs', and 'Improve Development Team Efficiency'. A red 'Request a Free Trial' button is positioned below the text. At the bottom of the page, there is a form with two input fields for 'First Name\*' and 'Last Name\*', both marked as '\*Required'.

coverity 1-415-321-5200 or contact sales

Solutions Products Services Customers News and Events Company Research Library Request A Free Trial

coverity Integrity Center

Architectural Analysis Dynamic Analysis Build Analysis Static Code Analysis

## Coverity Integrity Center

Improve Your Source Code Quality and Security

- Enhance Application Quality and Security
- Lower Development Costs
- Improve Development Team Efficiency

**Request a Free Trial**

Over 600 companies and 100,000 developers rely on Coverity products to identify, triage, and remediate critical software errors. See first hand how Coverity Software Architecture, Code, and Build Analysis can help improve the quality and security of your products.

\*Required

First Name\*

Last Name\*

---

---

# Thank you!

