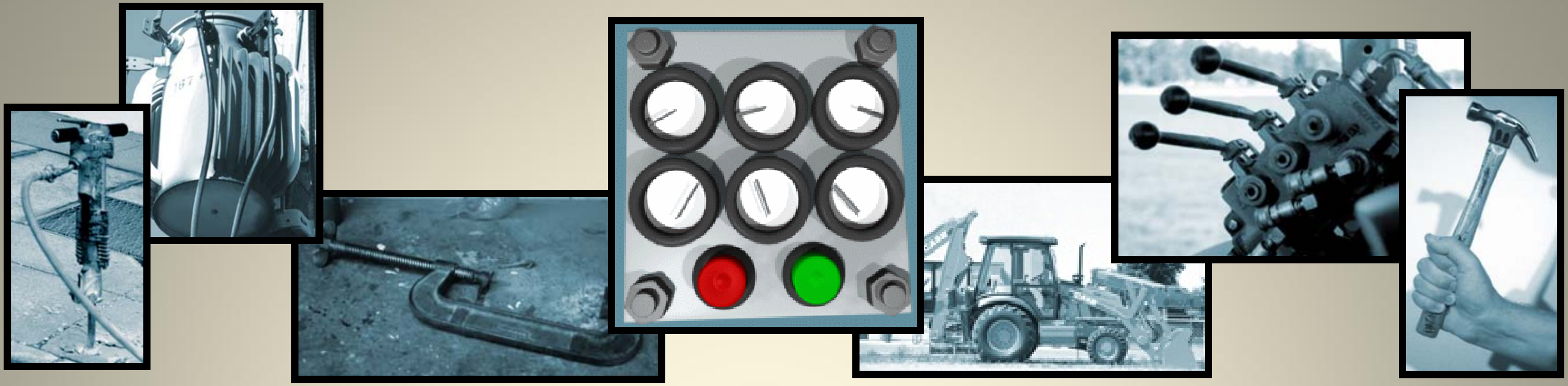


Security Measurement and Assuring Reliability through metrics Technology (SMART)

Applying Reliability Metrics to Security Vulnerabilities

Wayne Zage, Dolores Zage, Blake Self

Presentation Outline

1. Background/overview of S^2ERC and design metrics
2. Vulnerability Analyses

Security and Software Engineering Research Center (S^{2}ERC)

- An NSF Industry/University Cooperative Research Center established in 1986 and extended in 2010
- Participation by ten+ universities representing >50 researchers
- Collaborative, customized projects and technology transfer to affiliates are the hallmarks of the S^{2}ERC

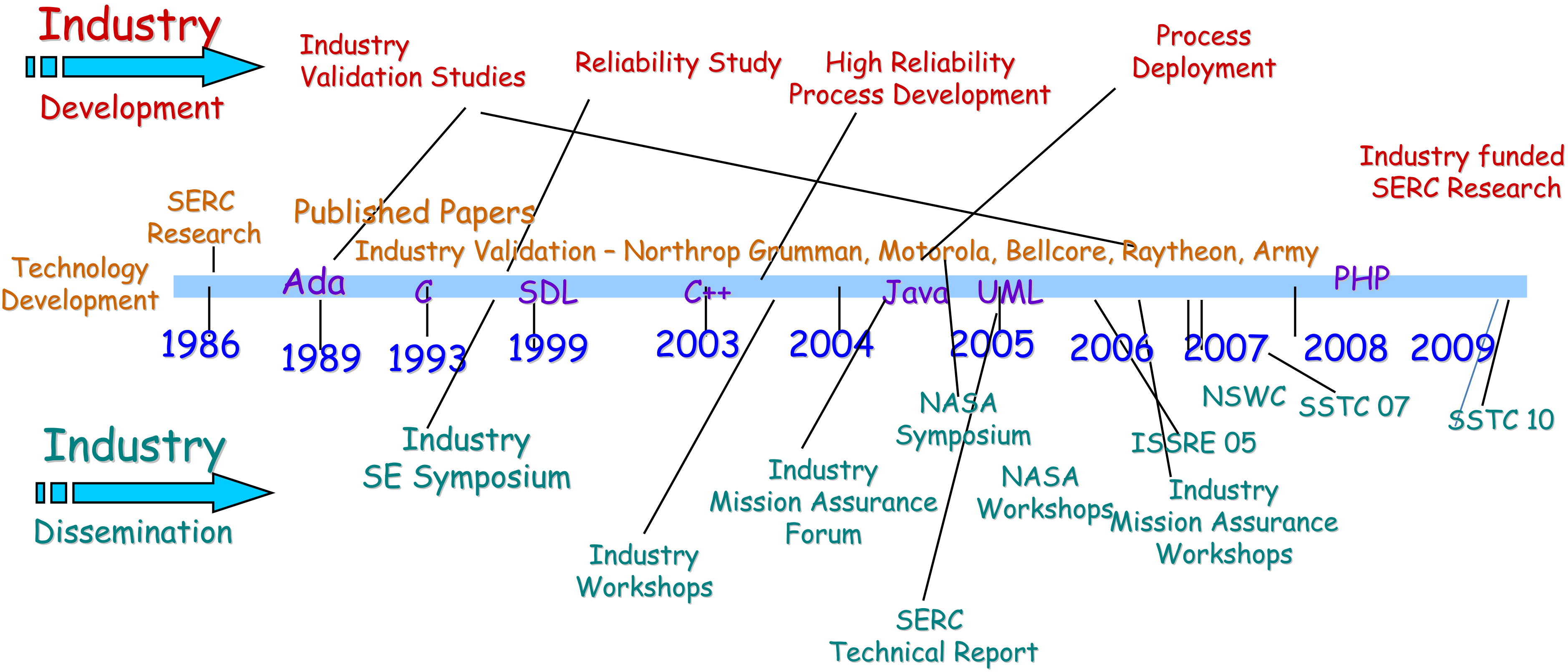
S^{2}ERC Participating Universities

- Ball State
- DePaul
- Illinois at Chicago
- Iowa State
- IUPUI
- IPFW
- Penn State
- Purdue
- West Florida
- Limerick

S^{2}ERC Industrial Affiliates

- Angie’s List
- Bingham McHale
- Blue Cross Blue Shield
- Boeing
- John Deere
- Intelligent Information Technologies
- Iowa Dept. of Transportation
- Lockheed Martin
- MacAulay Brown
- NASA
- Northrop Grumman
- Ontario Systems
- Raytheon
- NSWC – Crane
- Rockwell Collins
- TIAA
- Union Pacific
- US Army Research Lab
- US Dept. of Homeland Security

The S²ERC Design Metrics Research Timeline



Industry Development

Industry Validation Studies

Reliability Study

High Reliability Process Development

Process Deployment

Industry funded SERC Research

SERC Research

Published Papers

Technology Development

Industry Validation – Northrop Grumman, Motorola, Bellcore, Raytheon, Army

Ada C SDL C++ Java UML PHP

1986 1989 1993 1999 2003 2004 2005 2006 2007 2008 2009

Industry Dissemination

Industry SE Symposium

Industry Workshops

Industry Mission Assurance Forum

NASA Symposium

SERC Technical Report

NASA Workshops

ISSRE 05

NSWC

SSTC 07

Industry Mission Assurance Workshops

SSTC 10

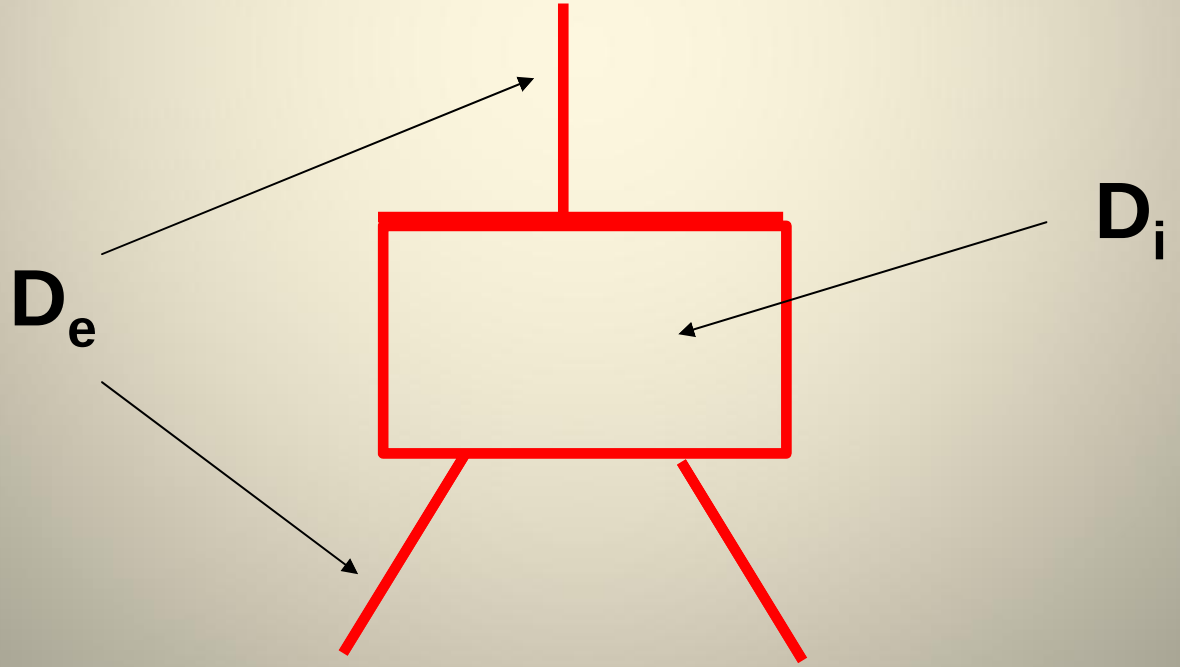
S^{2}ERC National Recognition

The **2007 Alexander Schwarzkopf Prize for Technological Innovation** from the NSF I/UCRC Association recognized the achievements in developing software design metrics that identify fault-prone modules early in the software lifecycle, thereby allowing significant improvements in software quality and productivity.

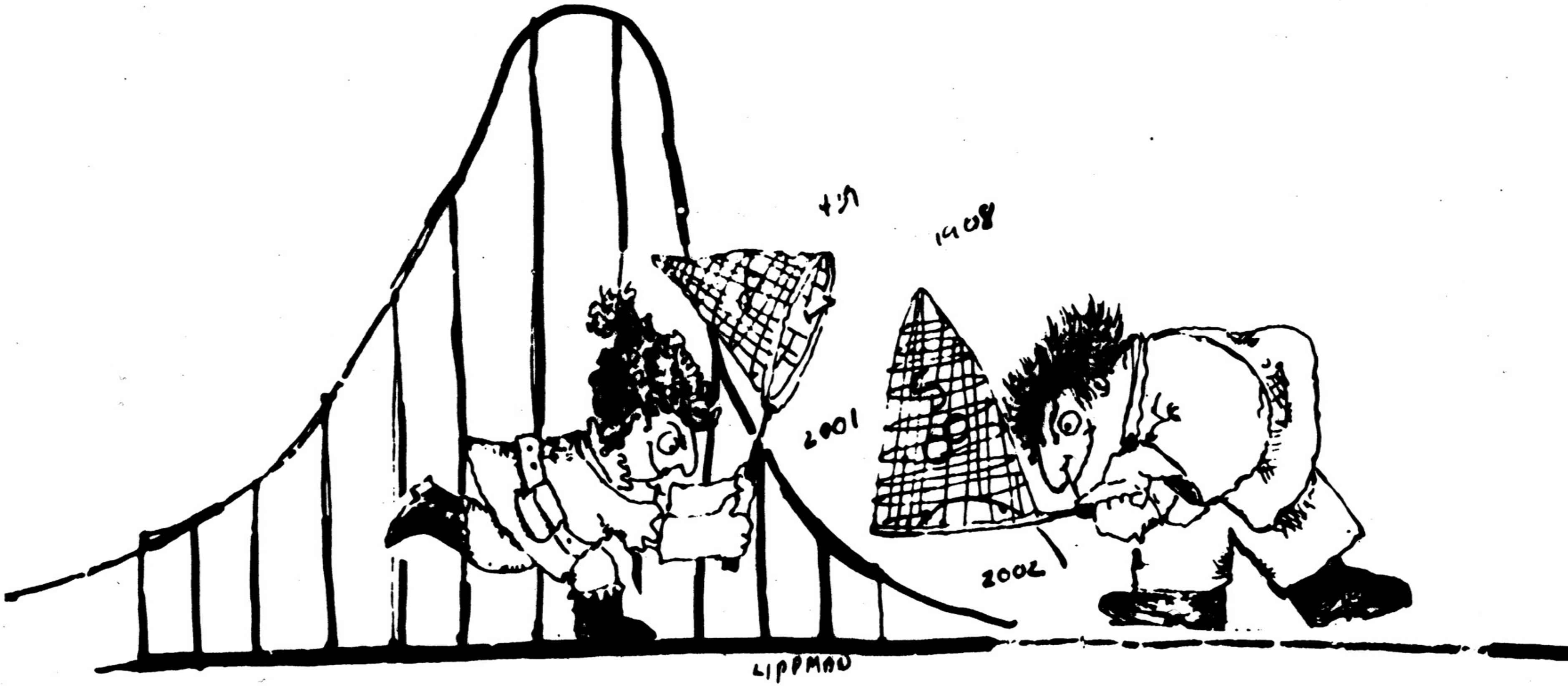
Overview of the Design Metrics, D_e and D_i

D_e - an external view of design complexity

D_i - an internal view of design complexity



Finding Outliers



adapted from *Introductory Statistical Analysis*, Anderson and Sclove, Houghton Mifflin, 1974.

S^{2}ERC Design Metrics Research has been funded by ...

National Science Foundation
Motorola Corp.
Nortel Technologies
Telcordia Technologies
Northrop Grumman Corp.
Computer Sciences Corp.
GTE Data Services
Magnavox Electronic Systems Co.
Harris Corp.
Raytheon
US Army Research Lab
Ball State University

The design metrics have been computed on

university-based projects

CSC’s STANFINS project

systems from the US Army Research Lab

Harris’ ROCC project

Magnavox’s AFATDS project

PBX system from Telcordia Technologies

three Northrop Grumman projects

three Raytheon projects

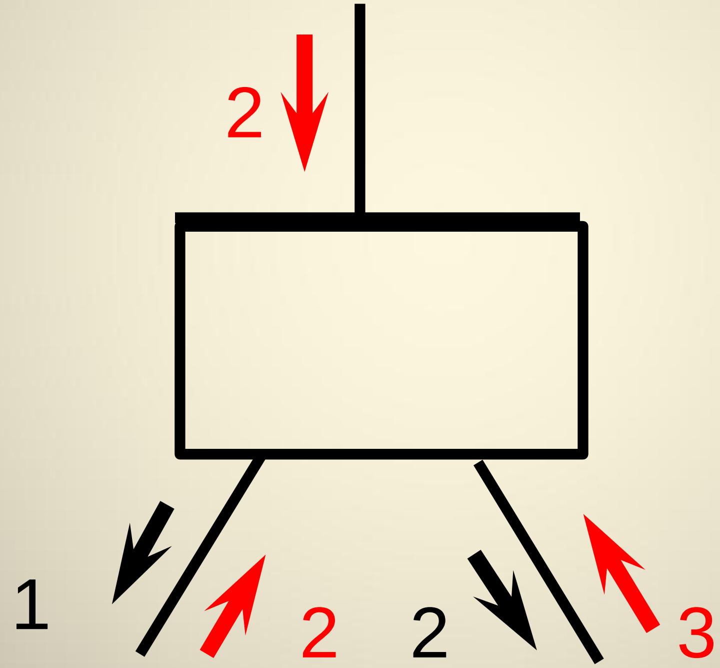
telecommunications systems from Motorola

Results:

The design metrics have correctly identified at least 76% of the defect-prone modules 100% of the time.

The External Metric D_e

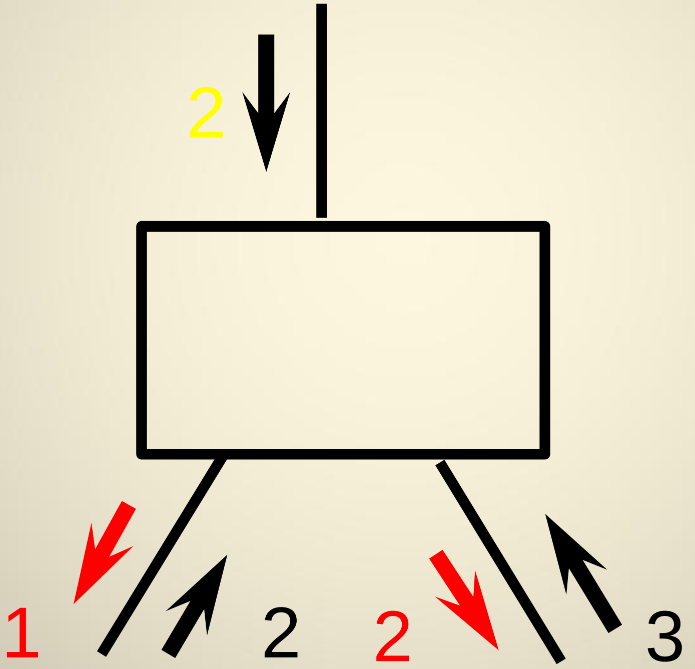
$D_e = e_1$(inflows * outflows) + e_2 (fan in * fan out)



D_e = ((2+2+3) * (1+2)) + (1*2) = 23

The External Metric D_e

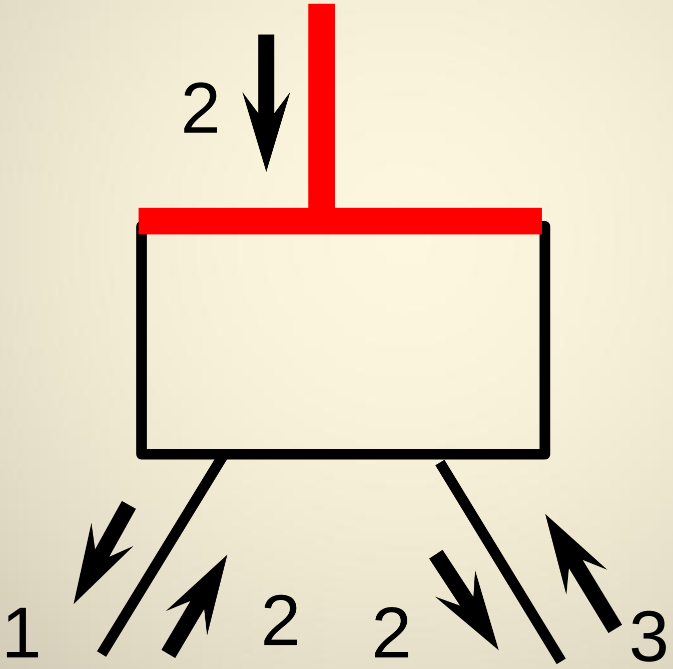
D_e = e_1(inflows * outflows) + e_2 (fan in * fan out)



D_e = ((2+2+3) * (1+2)) + (1*2) = 23

The External Metric D_e

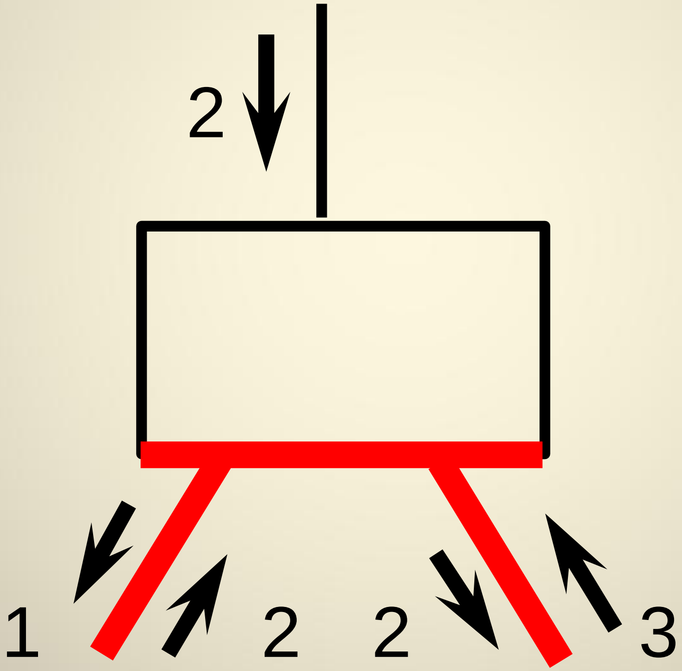
D_e = e_1(inflows * outflows) + e_2 (fan in * fan out)



D_e = ((2+2+3) * (1+2)) + (1*2) = 23

The External Metric D_e

D_e = e_1(inflows * outflows) + e_2 (fan in * fan out)



D_e = ((2+2+3) * (1+2)) + (1*2) = 23

The Internal Metric D_i

$$D_i = w_1(CC) + w_2(DSM) + w_3(I/O)$$

where:

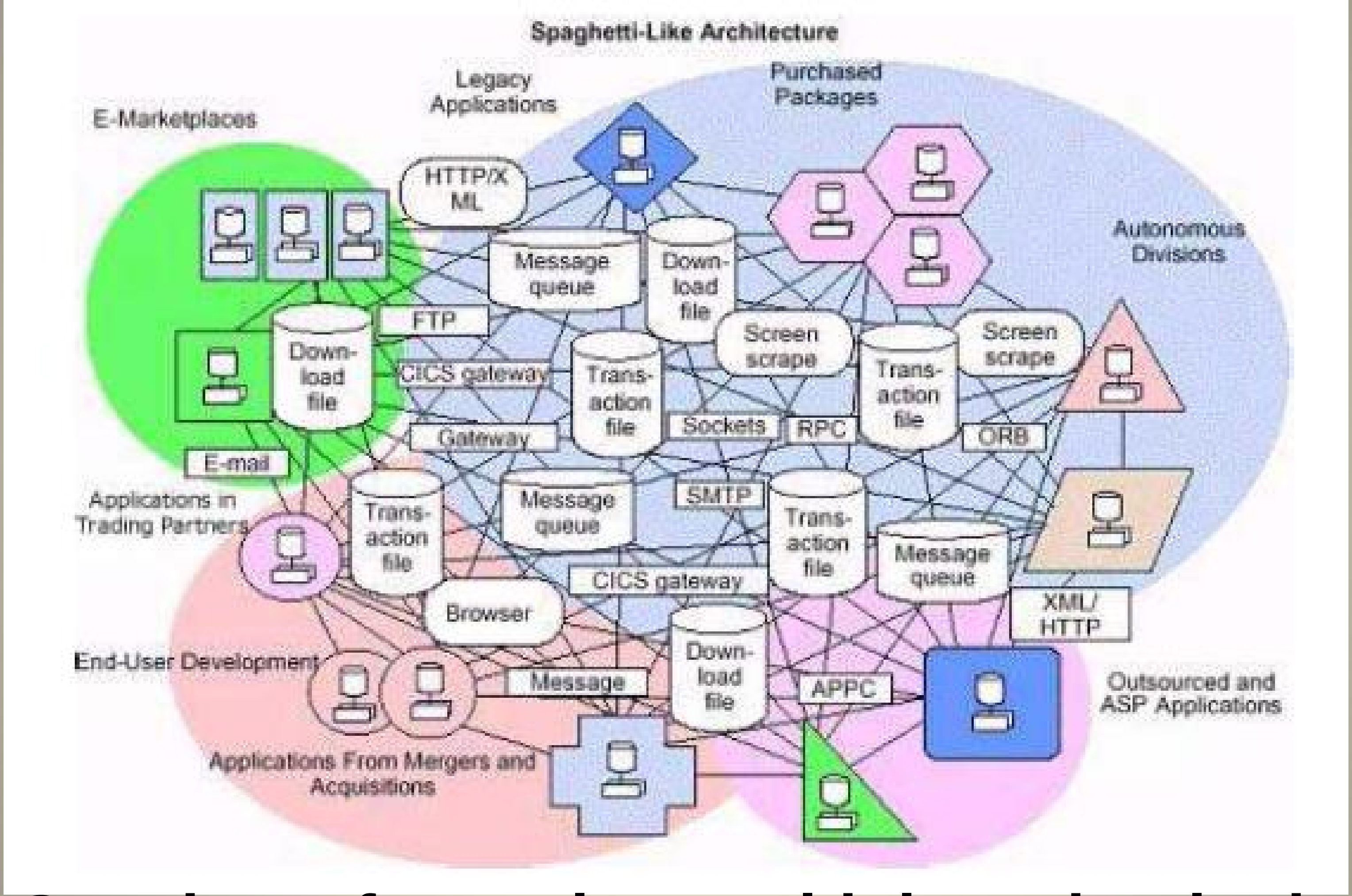
CC (Central Calls) are procedure or function invocations

DSM (Data Structure Manipulations) are references to complex data types

I/O (Input/Output) are external device accesses

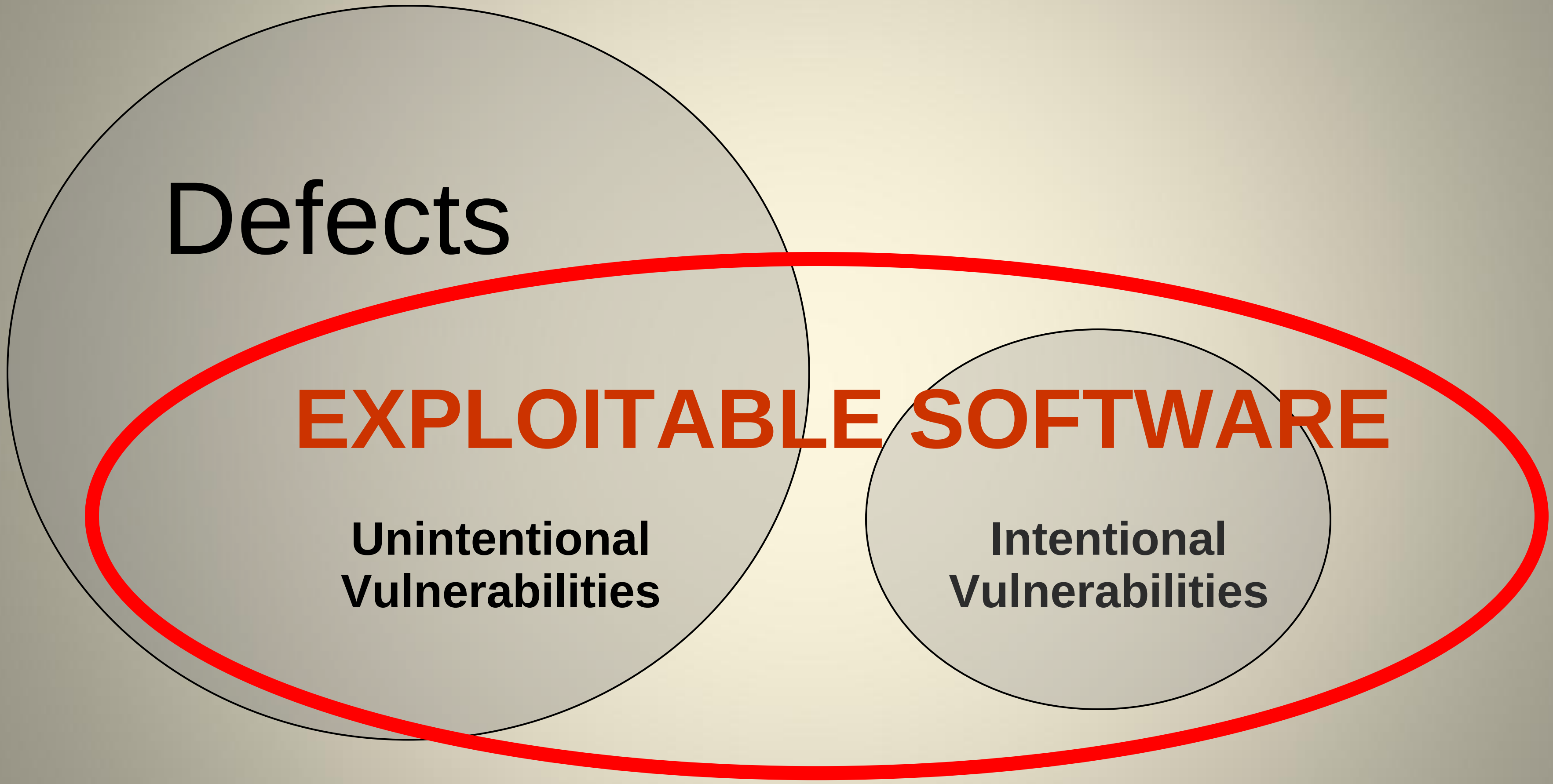
Extending Design Metrics Technology to a Software Security Engineering Process

Software Reality



Consists of complex, multiple technologies with multiple suppliers

Software Vulnerabilities



Defects

EXPLOITABLE SOFTWARE

Unintentional Vulnerabilities

Intentional Vulnerabilities

Exploitation potential of vulnerability independent of “intent”

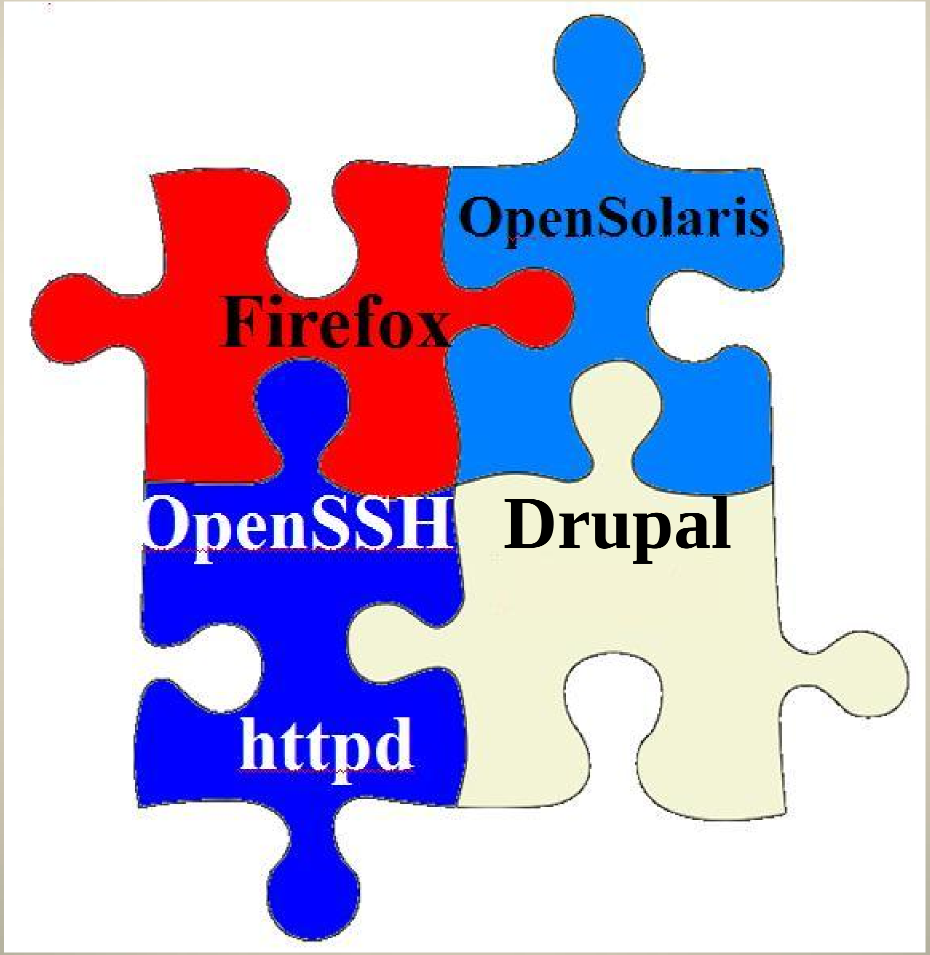
Reliability and Security Parallels

Reliability context (well-established) – Design Metrics Reliability Research		Security context (to be established) the SMART project funded by ARL
Fault-prone component Likely to contain faults	→	**Vulnerability-prone component** Likely to contain vulnerabilities
Failure-prone component Likely to have failures in field	→	**Attack-prone component** Likely to be exploited in the field

SMART Project Objectives

- Investigate the overlap and interrelationships in the software constructs that affect the reliability and security of software
- Develop security metrics to identify, categorize and analyze security weaknesses

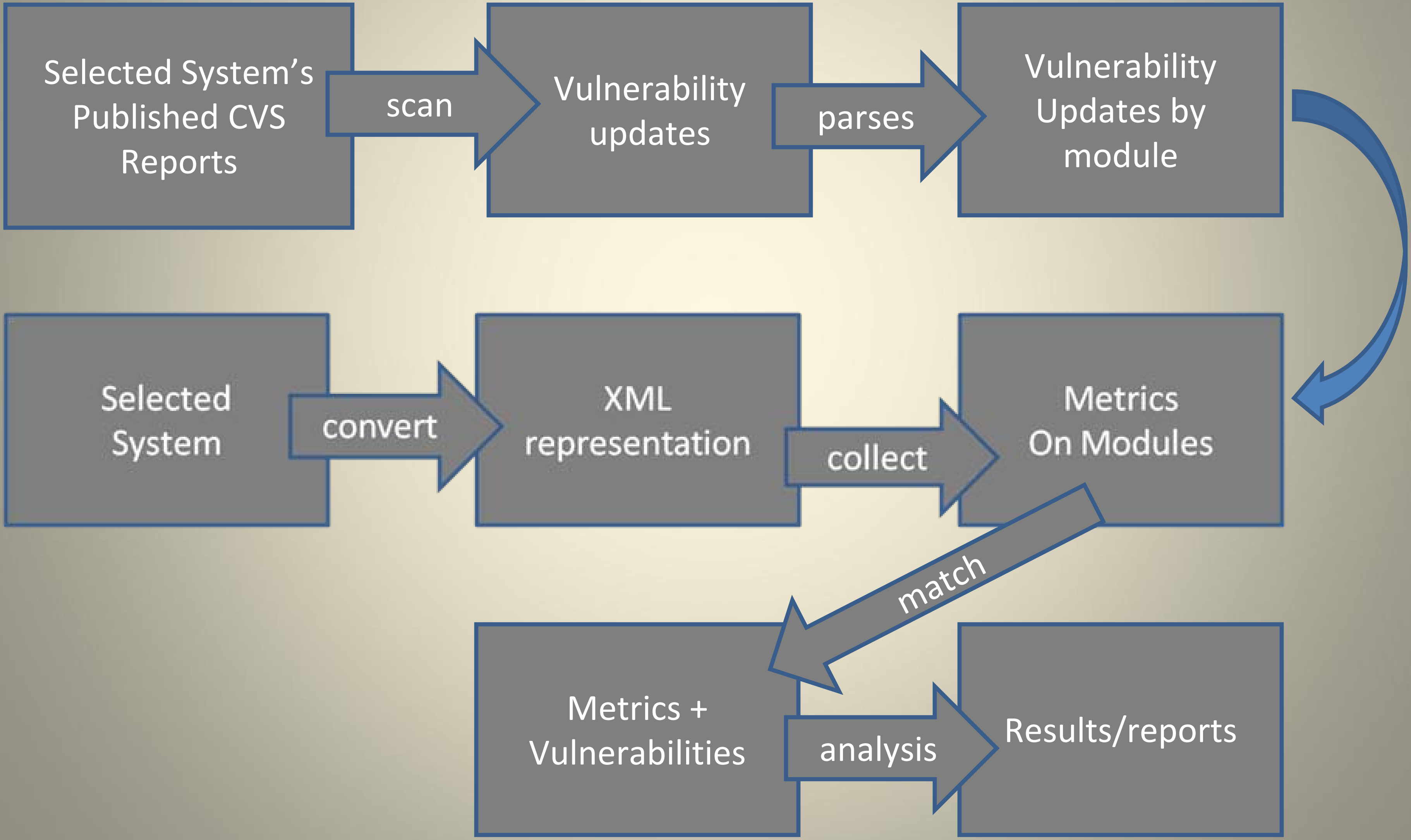
Our Representative Systems/Technologies



Security: Expectations and Directions

- Some vulnerable components will be identified by design metrics.
- A huge “win” if 50% of the components containing documented vulnerabilities are identified by design metrics.
- New security-related primitives will be needed to increase that percentage.

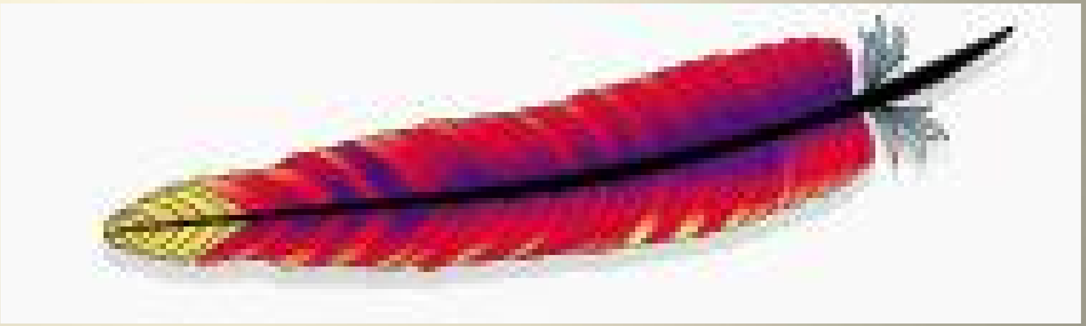
Our Team's General Process of Metric/Vulnerability Analysis



Common Weakness Enumeration and Systems

CWE Category	OPEN SOLARIS	FIREFOX 2.0.0.1	FIREFOX 2.0.0.2	FIREFOX 2.0.0.5	Open-SSH	HTTPD
Code Quality	6	1	0	2	2	11
Data Handling	5	10	30	69	13*	19
Security Features	8	2	0	20	2	5
Time and State	5	0	0	0	0	0
Error Handling	0	5	0	0	1	1
API Abuse	0	0	0	0	2	1
	24	18	30	91	20	37

Secret life of the open source code



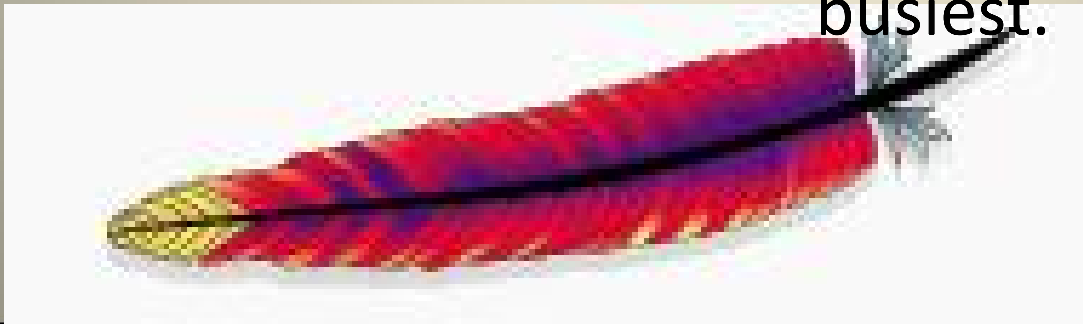
Ongoing Work

- Apache
- Open Solaris
- FireFox
- OpenSSH
- Drupal

Apache HTTP Server

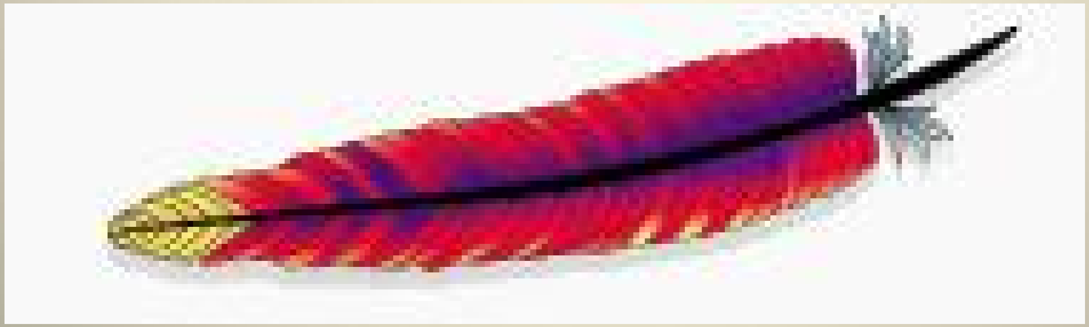
- Designed by Robert McCool
- Developed by Apache Software foundation
- Initial release 1995
- Latest release 2.2.15 on 03/06/2010

- Since April 1996 Apache has been the most popular HTTP server on the WWW.
- As of March 2009 Apache served over 46% of all websites and over 66% of the million busiest.



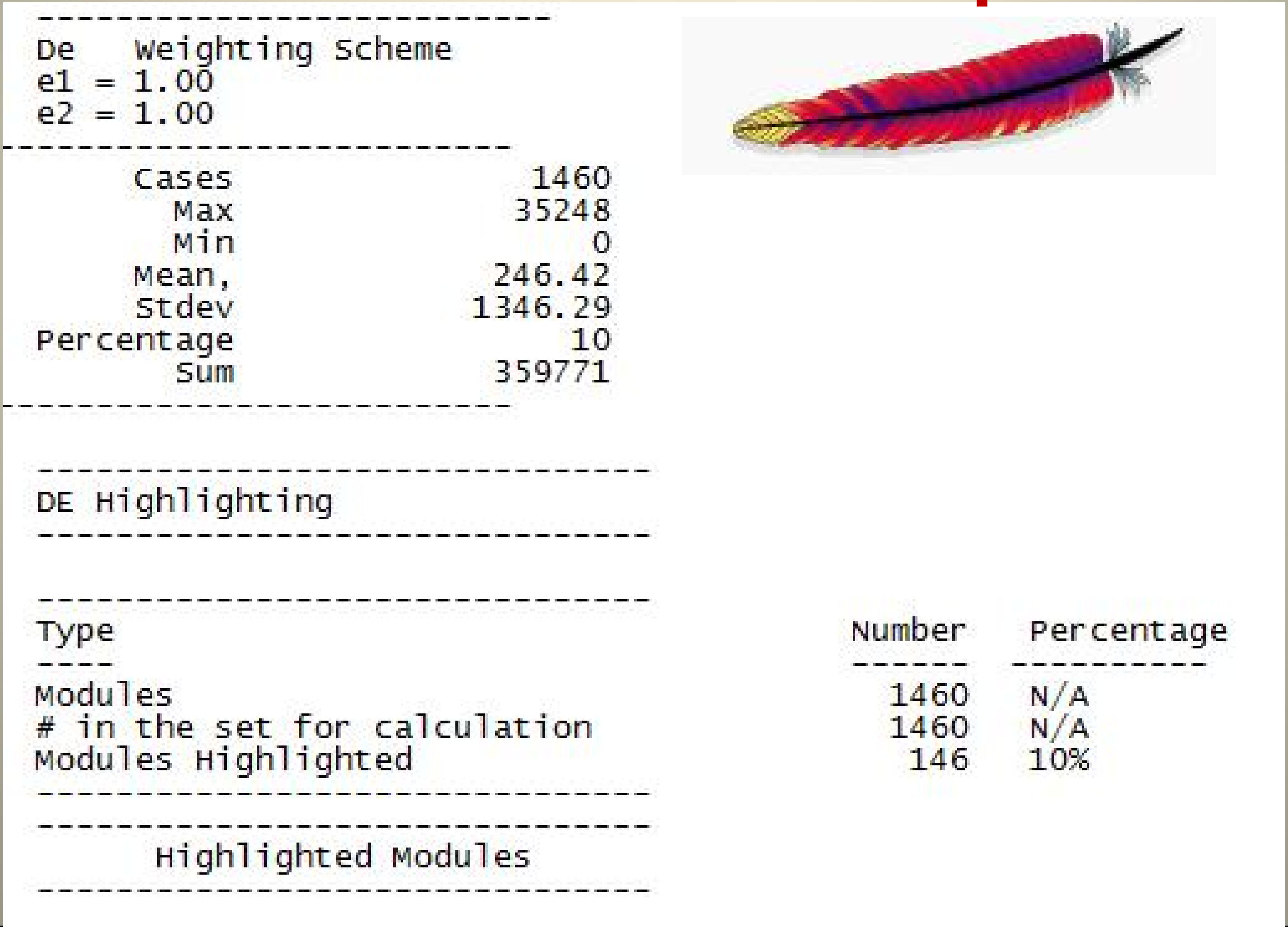
Apache Vulnerabilities

- Apache
- For version 1.3.1
  - 144 files
  - 8 vulnerabilities identified



D_e Metric Analysis on Apache 1.3.1

87.5% or 7 out of the 8 vulnerable modules were identified in the top 10%

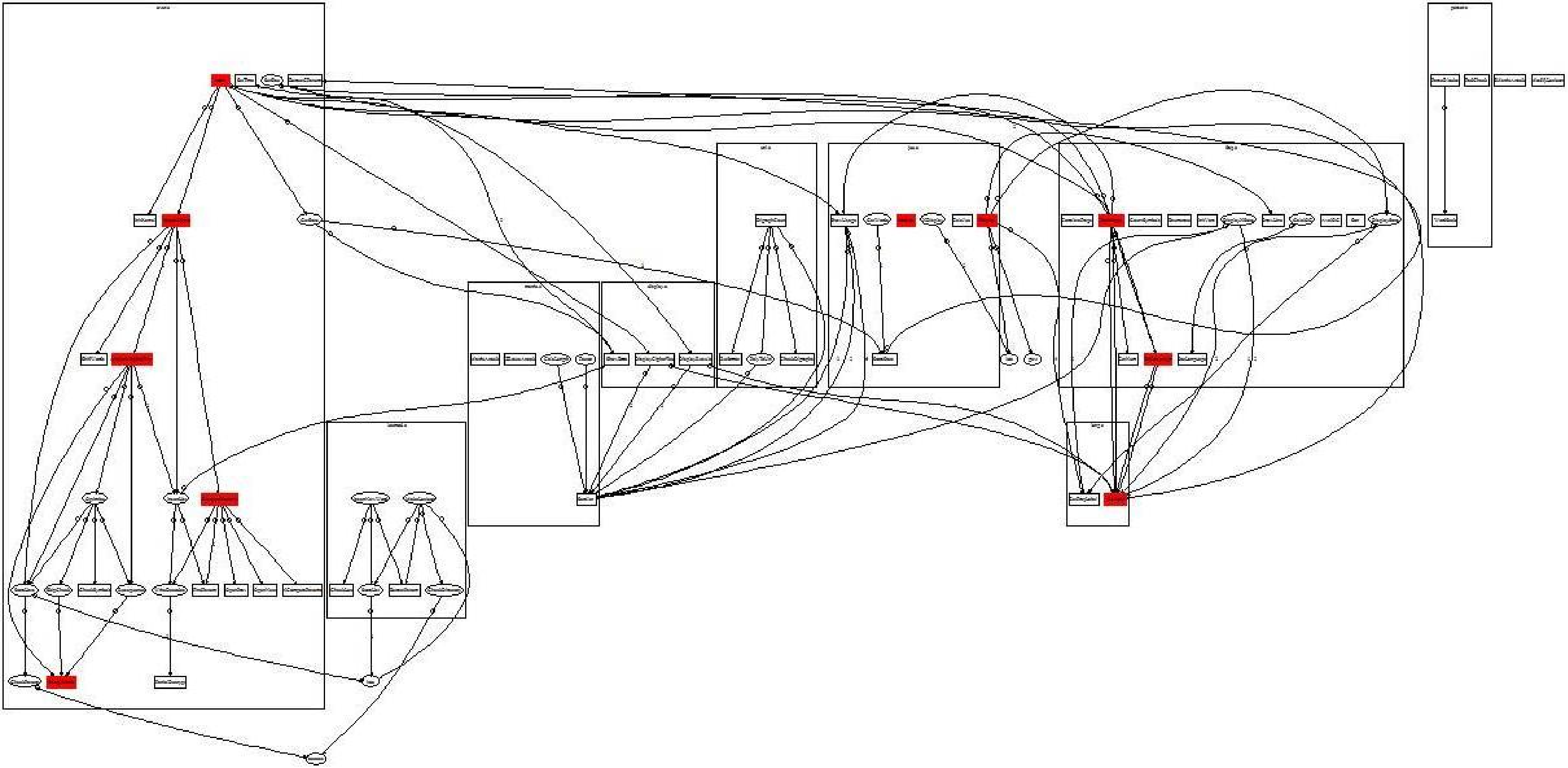


```
--------------------------
De    Weighting Scheme
e1 = 1.00
e2 = 1.00
--------------------------
     Cases              1460
       Max             35248
       Min                 0
     Mean,            246.42
     Stdev           1346.29
Percentage                10
       Sum            359771
--------------------------


--------------------------------
DE Highlighting
--------------------------------

--------------------------------
Type                                      Number   Percentage
----                                      ------   ----------
Modules                                     1460   N/A
# in the set for calculation                1460   N/A
Modules Highlighted                          146   10%
--------------------------------
--------------------------------
      Highlighted Modules
--------------------------------
```

Graphic of Indicated Stress Modules



OpenSolaris

- based on Sun Microsystems' Solaris
- Latest release 2009.06 on 06/01/2009

- OpenSolaris is derived from the Unix System V Release 4 codebase, with significant modifications made by Sun since it bought the rights to that code in 1994.

OpenSolaris Vulnerabilities

OpenSolaris Vulnerabilities and Source

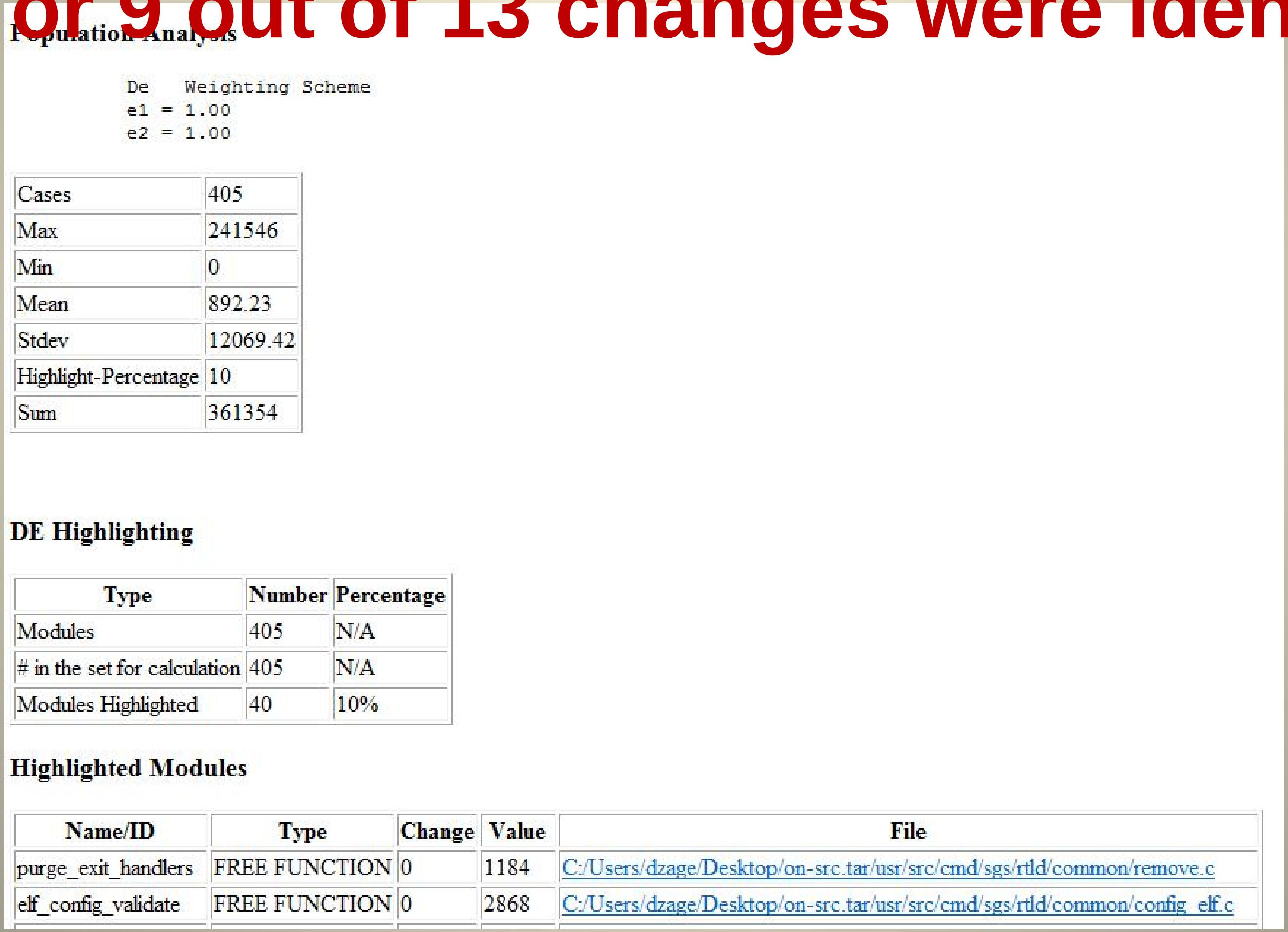
- 22,600 files in the downloadable tar file
- 23 module updates from vulnerabilities totaling 37 changes
- 5 is the largest # of changes on one module
- We want to identify the 23 modules out of the 22,600 x modules in files, approximately 1 in 10,000

Divide and Conquer

- 18 module vulnerability updates located in the /usr/src/uts/common stem (3983 files)
- 5,946,281 xml tags in the 3983 files
- 5 module vulnerability updates located in the /usr/src/cmd/sgs/rtld/common stem (29 files)
- 90,417 xml tags in the 29 files

D_e Metric Analysis on /usr/src/cmd/sgs/rtld/common

60% or 3 out of the 5 vulnerable modules or 69% or 9 out of 13 changes were identified



Population Analysis

```
De   Weighting Scheme
e1 = 1.00
e2 = 1.00
```

Cases	405
Max	241546
Min	0
Mean	892.23
Stdev	12069.42
Highlight-Percentage	10
Sum	361354

DE Highlighting

Type	Number	Percentage
Modules	405	N/A
# in the set for calculation	405	N/A
Modules Highlighted	40	10%

Highlighted Modules

Name/ID	Type	Change	Value	File
purge_exit_handlers	FREE FUNCTION	0	1184	C:/Users/dzage/Desktop/on-src.tar/usr/src/cmd/sgs/rtld/common/remove.c
elf_config_validate	FREE FUNCTION	0	2868	C:/Users/dzage/Desktop/on-src.tar/usr/src/cmd/sgs/rtld/common/config_elf.c

FireFox

- **Mozilla Firefox** is a web browser
- Designed by and developed by Mozilla Corporation
- Initial release November 9, 2004
- Latest release 3.6.6 on 06/26/2010
- Firefox had 22.05% of the recorded usage share of web browsers as of March 2009, making it the second most popular browser in terms of current use worldwide, after Internet Explorer.

FireFox Vulnerabilities

General Analysis

- Multiple versions of three software systems studied
- Source Code is THE primary source of vulnerability reports
- At least 51% and at most 86% of reported vulnerabilities are in Data Handling
- Data Structure Manipulations (DSM) is the best predictor of vulnerabilities

OpenSSH

- Developed by OpenBSD Project
- OpenSSH first appeared in OpenBSD 2.6 and the first portable release was made in October 1999
- Latest release 5.5/5.5p1 on 04/16/2010
- Set of computer programs providing encrypted communication sessions over a computer network using the ssh protocol.

OpenSSH Vulnerabilities

OpenSSH 3.8p1

- Files: 243
- Modules: 2,437
- Definitions: 2,992
- preprocessor directives: 5,147
- user defined include files: 1,101
- conditional expressions: 61,815
- xml tags: 703,850
- 31 vulnerable modules identified

- Ranked modules by De
- 18 of the 31 vulnerable modules in the top 10% (58%)
- 23 of the 31 vulnerable modules in top 20% (74%)
- The remaining 8 had additional suspicious patterns :
  - Modules were named “x” and had a “mate” x1
  - Duplicate names

Drupal

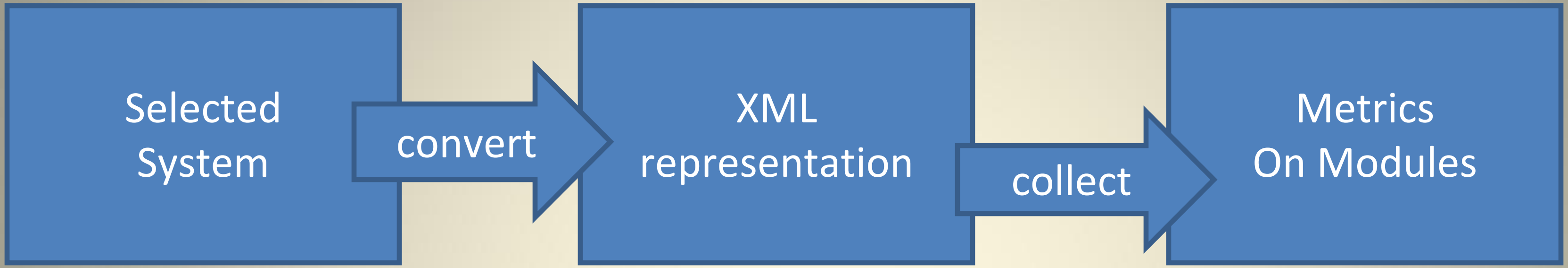
- Initial release January 2001 (2001-01)
- Latest release 6.17 June 2010
- Written in PHP
- Operating system cross-platform

- Open Source Content Management System (CMS)
- Over 350,000 subscribed members today
- Consists of PHP, INC, JavaScript, Perl, XML files
- DrupalSites.net is a directory that list thousands of websites powered by Drupal
- Winner of Best Overall 2008 Open Source CMS Award for Second Year in a Row
- Listed as one of the Open Source PHP applications that changed the world

Drupal Vulnerabilities and Source

- PHP Process of Metric/Vulnerability Analysis
  - Constructed the **D**rupal **V**ulnerabilities **M**iner (DVM)
  - the Drupal CVS web site (http://www.drupal.org/security).
  - DVM isolated 277 RCS file patches identified from 140 vulnerability updates.
- Approximately 105 Drupal PHP files

Generic Source Analysis



Selected System

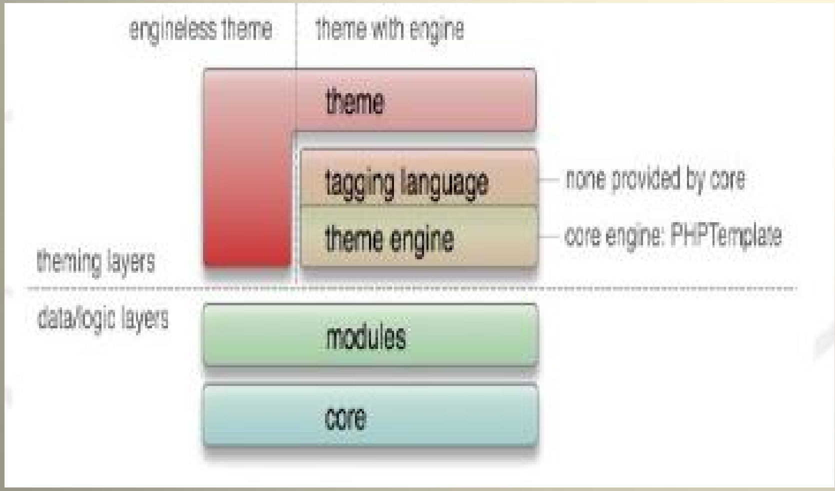
convert

XML representation

collect

Metrics On Modules

PHP Source Analysis



Designing a PHP2XML tool

Mapping PHP primitive tags to primitive design metrics

Benefits

- A knowledge of *where* vulnerabilities are most likely to reside can help prioritize security efforts.
- Analyzing multiple technologies and mapping the vulnerabilities to the CWE to ensure coverage
- Merge multiple technologies through mapping the individual xml representations, isolates commonality and individuality

Next Research Steps

- Continue the analysis of the open source systems
- Investigate other primitives to identify, categorize security weakness
- Apply technologies to systems other than open source (YOURS?)
- Combine into the network model