REPOSITORY MINING AND PROGRAM ANALYSIS & TESTING: BETTER TOGETHER?

Alessandro (Alex) Orso

School of Computer Science – College of Computing
Georgia Institute of Technology
http://www.cc.gatech.edu/~orso/

Partially supported by: NSF, DHS, and US Air Force

MSR PAPERS AND PROGRAM ANALYSIS

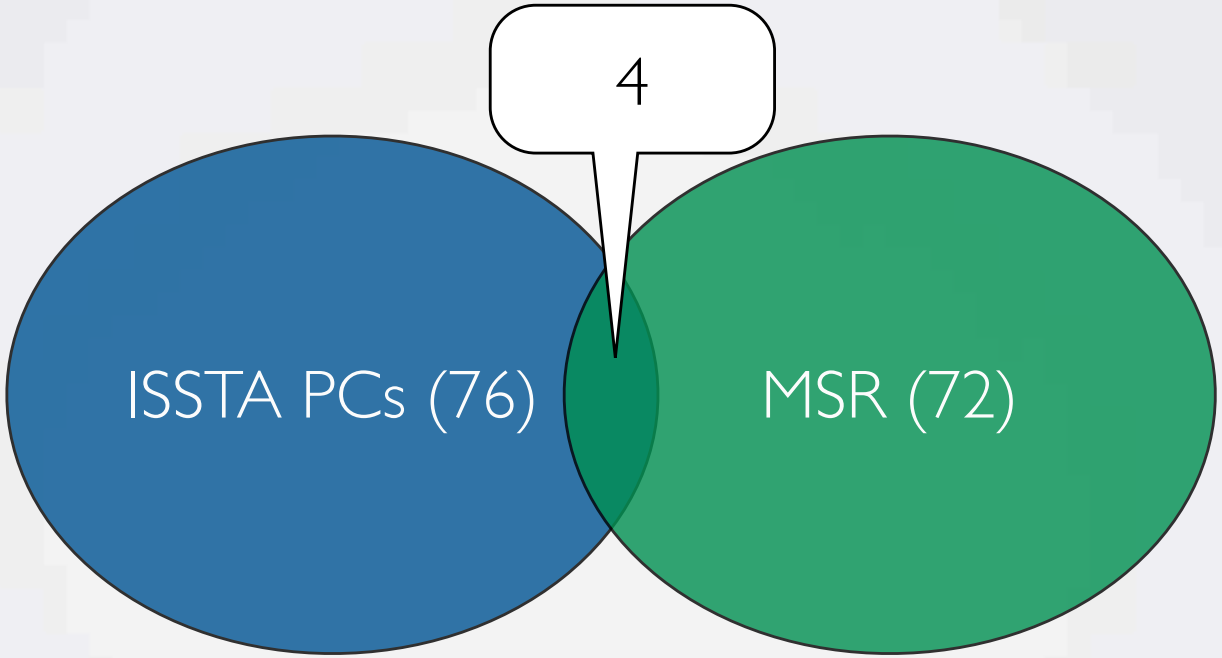
PROGRAM ANALYSIS AND SOFTWARE ARCHIVES

- **Mini-history of software archives**
  - < 1996 – Mostly small examples, limited evaluation
  - 1996 – Siemens suite (<500 LOC)
  - 2005 – Software-artifact Infrastructure Repository
  - 2006 – Eclipse Bug Data
  - 2007 – iBUGS

- In 2010, much (most?) research still uses the Siemens suite

ISSUE #1

Communication



ISSUE #2

Mismatch in assumptions (or schisms)

- **(Most) program analyses**
  - Complete programs
  - Single language
  - Restricted set of features
  - Soundness
  - False positives problematic
- **Mining techniques**
  - Incomplete programs
  - Multiple languages
  - Complete languages
  - Noisy data
  - False positives acceptable

ISSUE #3

Infrastructure

- **Program analysis tools**
  - Unavailable
  - Unusable
  - Limited
- **Mining infrastructure**
  - No standard format
  - Complicated setup
  - Unusable

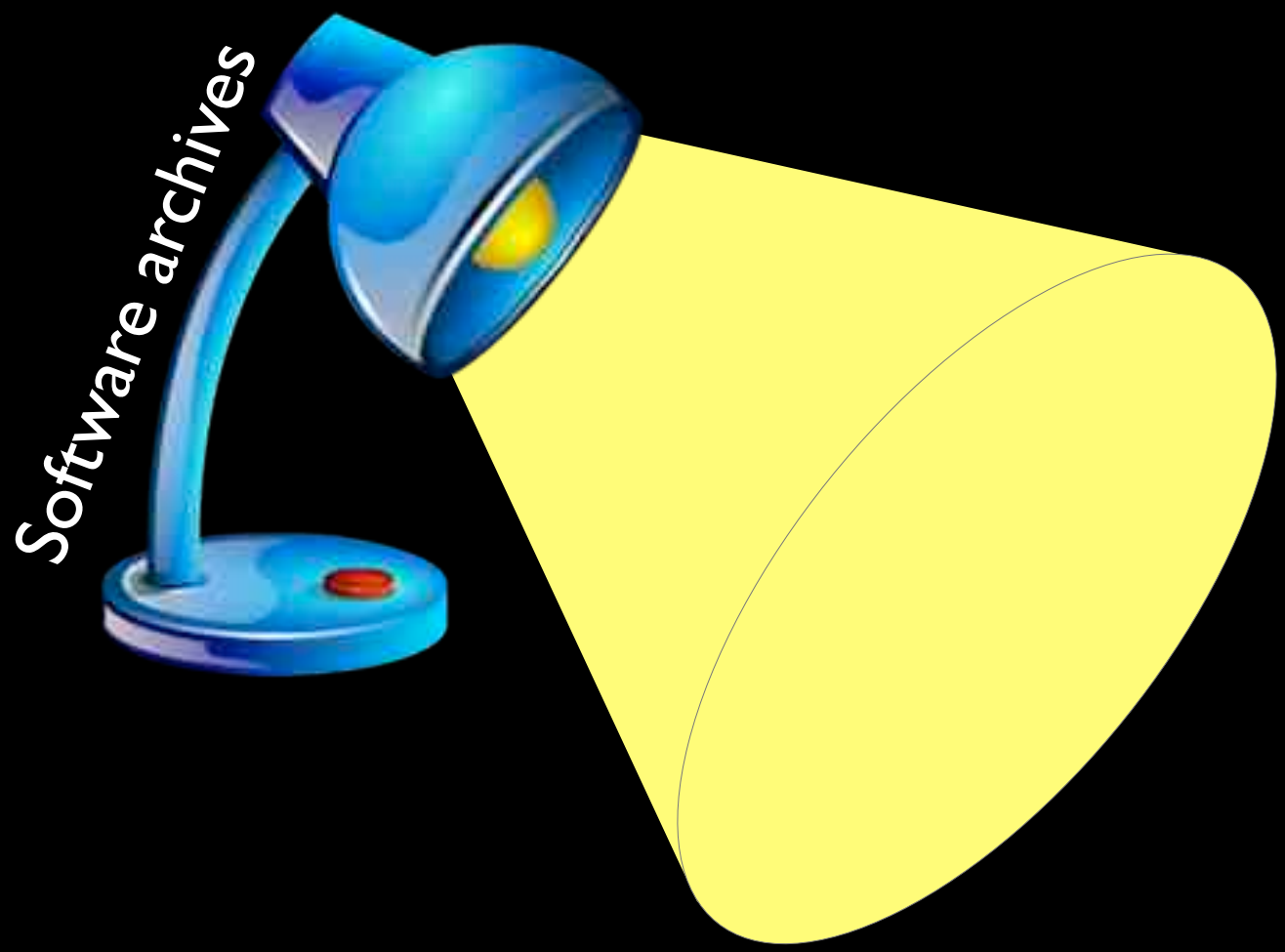
ISSUE #4

Narrow focus of some MSA research

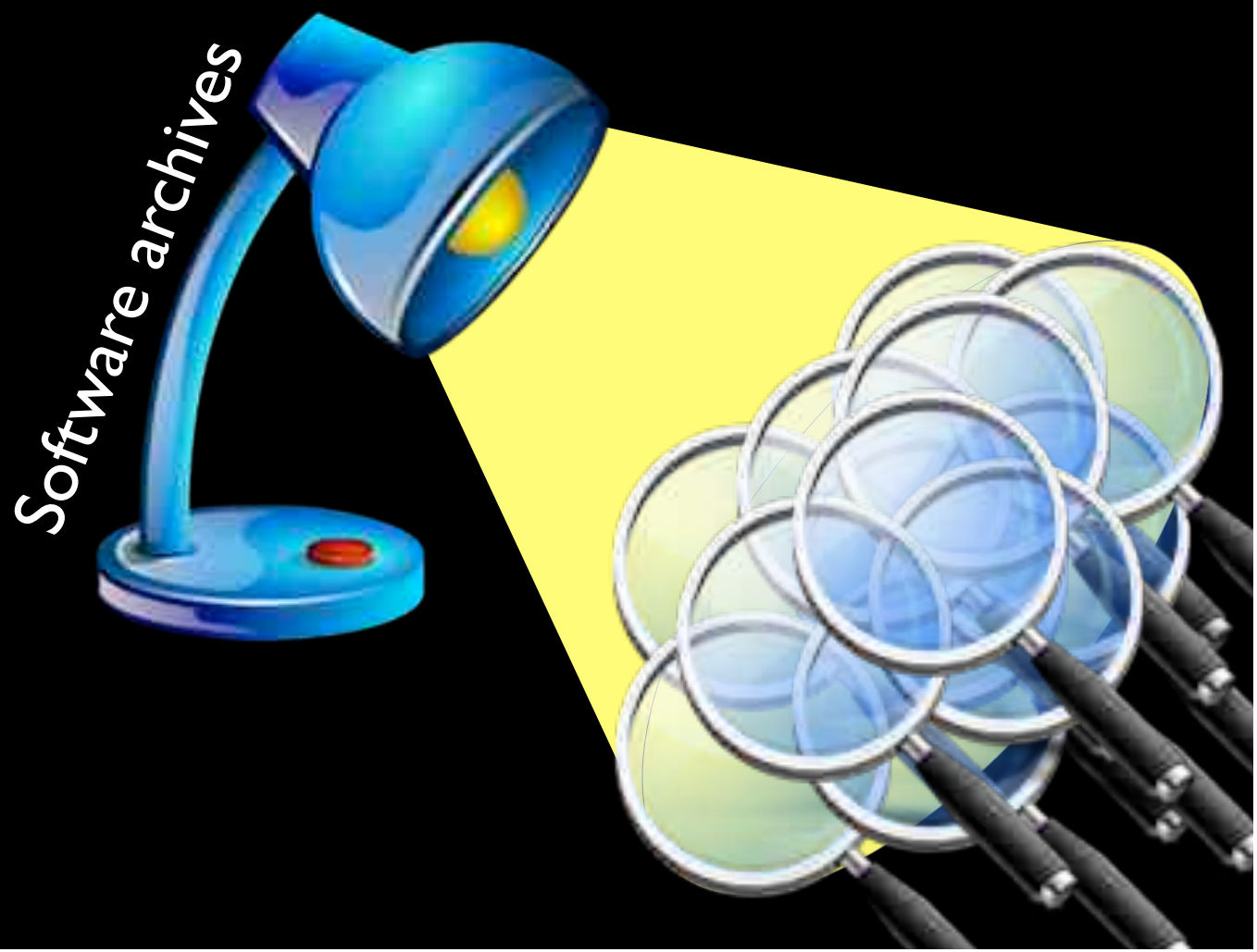
LOOKING FOR GOLD...

LOOKING FOR KEYS...

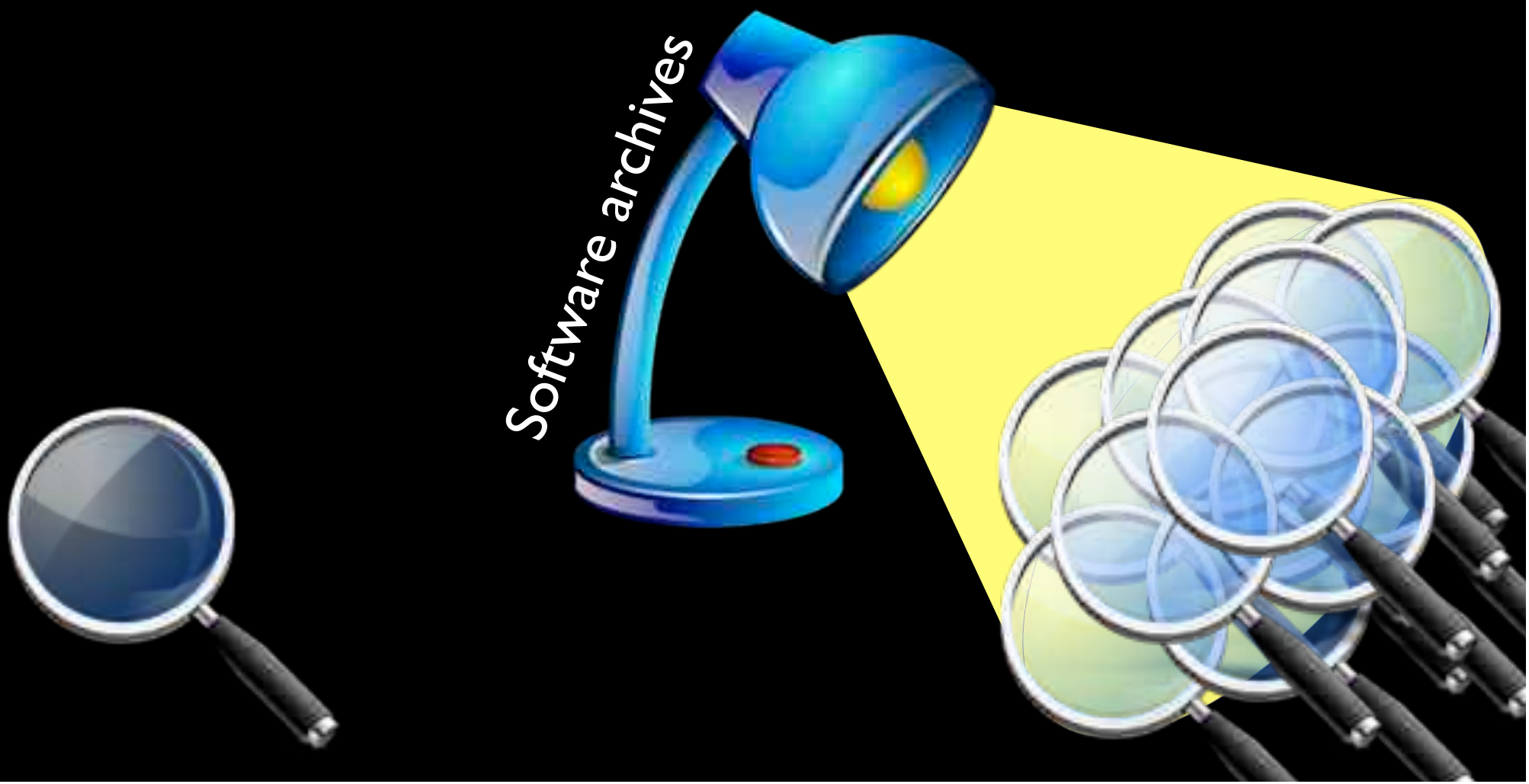
LOOKING FOR KEYS...



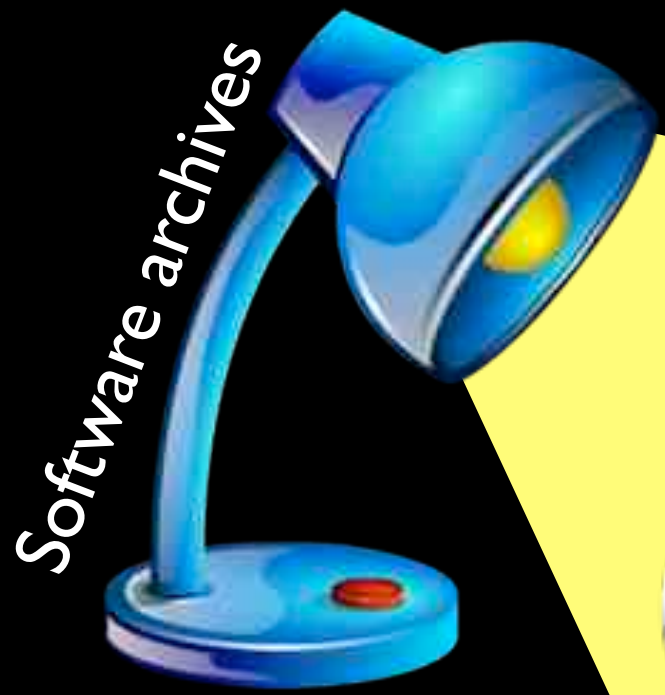
LOOKING FOR KEYS...



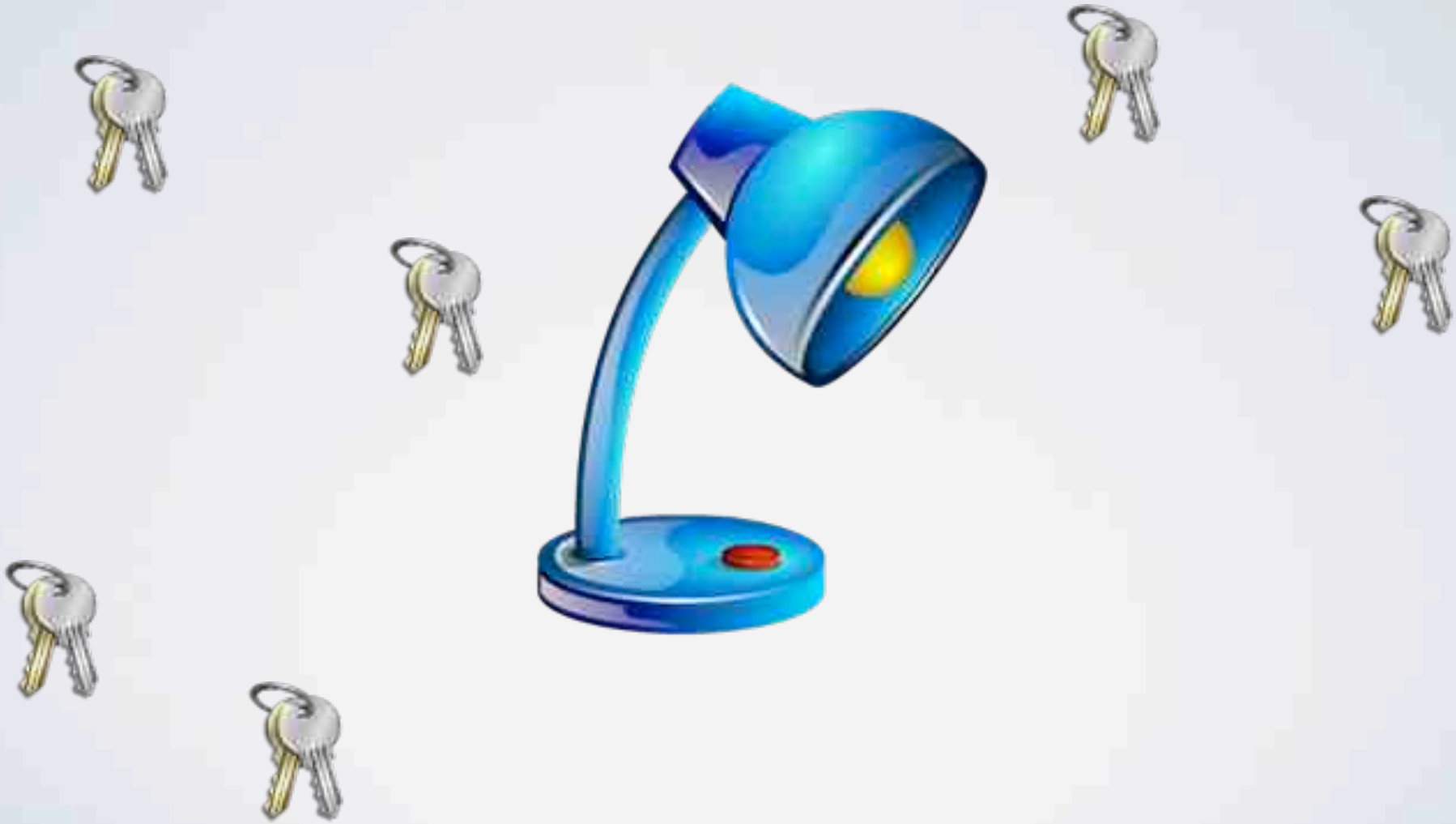
LOOKING FOR KEYS...



LOOKING FOR KEYS...



MAYBE IF WE TURN ON THE LIGHT



MAYBE IF WE TURN ON THE LIGHT

MINING MORE THAN ARCHIVES

Software

MINING MORE THAN ARCHIVES

Software Archives

MINING MORE THAN ARCHIVES

Software

Archives

Program runs
Program traces
...

MINING MORE THAN ARCHIVES

Software

Archives

Program runs
Program traces
...

Static/dynamic metrics

MINING MORE THAN ARCHIVES

Software

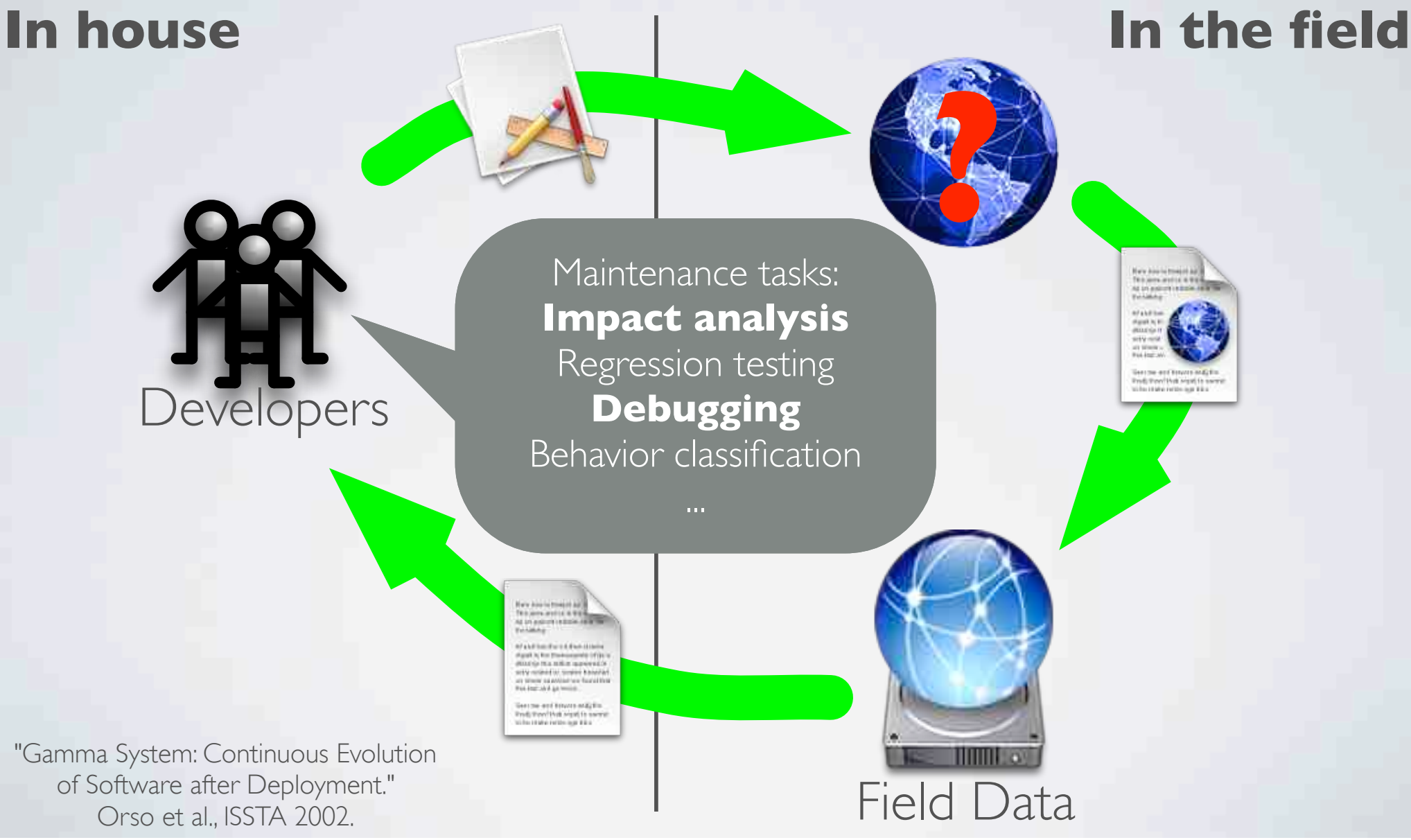
Archives

Program runs
Program traces
...

Static/dynamic metrics

•••

GAMMA PROJECT



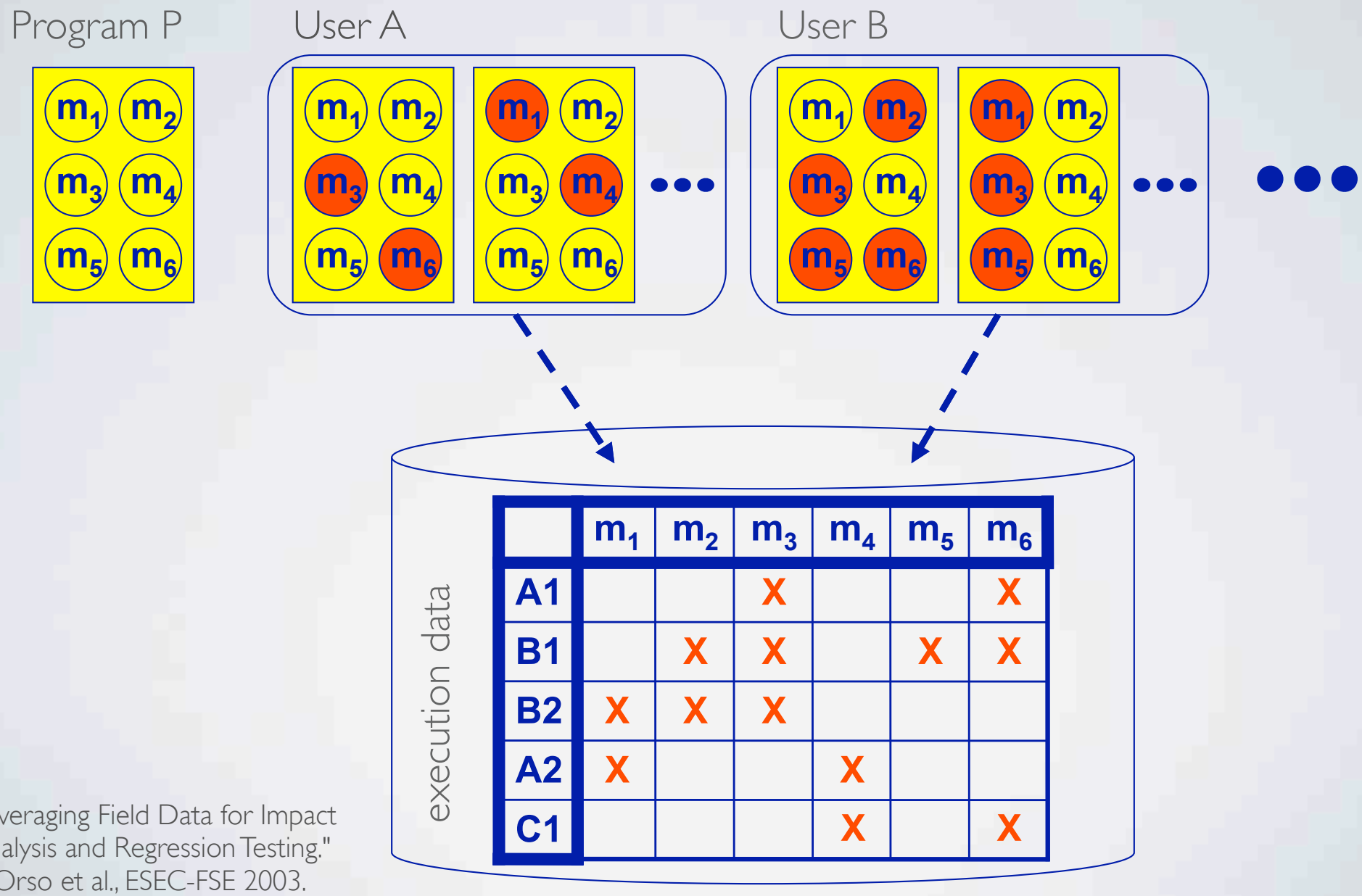
"Gamma System: Continuous Evolution of Software after Deployment."
Orso et al., ISSTA 2002.

IMPACT ANALYSIS

IMPACT ANALYSIS

- **Assess effects of changes on a software system**
  - *Predictive*: help decide which changes to perform and how to implement changes
- **Our approach**
  - *Program-sensitive impact analysis*
  - *User-sensitive impact analysis*

IMPACT ANALYSIS USING FIELD DATA



	m_1	m_2	m_3	m_4	m_5	m_6
A1			X			X
B1		X	X		X	X
B2	X	X	X			
A2	X			X		
C1				X		X

"Leveraging Field Data for Impact Analysis and Regression Testing." Orso et al., ESEC-FSE 2003.

PROGRAM-SENSITIVE IMPACT ANALYSIS

Input:

1. *Field execution data*

	m_1	m_2	m_3	m_4	m_5	m_6
A1			X			X
B1		X	X		X	X
B2	X	X	X			
A2	X			X		
C1				X		X

2. *Change*

   C={m2, m5}

Output:

Impact set =

PROGRAM-SENSITIVE IMPACT ANALYSIS

Input:

1. *Field execution data*

	m_1	m_2	m_3	m_4	m_5	m_6
A1			X			X
B1		X	X		X	X
B2	X	X	X			
A2	X			X		
C1				X		X

2. *Change*

   C={m2, m5}

Output:

Impact set =

Step 1

- Identify user executions through methods in C
- Identify methods covered by such executions

PROGRAM-SENSITIVE IMPACT ANALYSIS

Input:

1. *Field execution data*

	m_1	m_2	m_3	m_4	m_5	m_6
A1			X			X
B1		X	X		X	X
B2	X	X	X			
A2	X			X		
C1				X		X

2. *Change*

   C={m2, m5}

Output:

Impact set =

Step 1

- Identify user executions through methods in C
- Identify methods covered by such executions

PROGRAM-SENSITIVE IMPACT ANALYSIS

Input:

1. *Field execution data*

	m_1	m_2	m_3	m_4	m_5	m_6
A1			X			X
B1		X	X		X	X
B2	X	X	X			
A2	X			X		
C1				X		X

2. *Change*

   C={m2, m5}

Output:

Impact set =

Step 1

- Identify user executions through methods in C
- Identify methods covered by such executions

PROGRAM-SENSITIVE IMPACT ANALYSIS

Input:

1. *Field execution data*

	m_1	m_2	m_3	m_4	m_5	m_6
A1			X			X
B1		X	X		X	X
B2	X	X	X			
A2	X			X		
C1				X		X

2. *Change*

   C={m2, m5}

Output:

Impact set =

Step 1

- Identify user executions through methods in C
- Identify methods covered by such executions

PROGRAM-SENSITIVE IMPACT ANALYSIS

Input:

1. *Field execution data*

	m_1	m_2	m_3	m_4	m_5	m_6
A1			X			X
B1		X	X		X	X
B2	X	X	X			
A2	X			X		
C1				X		X

2. *Change*

   C={m2, m5}

Output:

Impact set =

Step 1

- Identify user executions through methods in C
- Identify methods covered by such executions

covered methods = **{m1,m2,m3,m5,m6}**

PROGRAM-SENSITIVE IMPACT ANALYSIS

Input:

1. *Field execution data*

	m_1	m_2	m_3	m_4	m_5	m_6
A1			X			X
B1		X	X		X	X
B2	X	X	X			
A2	X			X		
C1				X		X

2. *Change*

   C={m2, m5}

Output:

Impact set =

Step 1

- Identify user executions through methods in C
- Identify methods covered by such executions

covered methods = **{m1,m2,m3,m5,m6}**

Step 2

- Dynamic forward slice from C

PROGRAM-SENSITIVE IMPACT ANALYSIS

Input:

1. *Field execution data*

	m_1	m_2	m_3	m_4	m_5	m_6
A1			X			X
B1		X	X		X	X
B2	X	X	X			
A2	X			X		
C1				X		X

2. *Change*

   C={m2, m5}

Output:

Impact set = **{m2,m5,m6}**

Step 1

- Identify user executions through methods in C
- Identify methods covered by such executions

covered methods = **{m1,m2,m3,m5,m6}**

Step 2

- Dynamic forward slice from C

dynamic fwd slice = **{m2,m5,m6}**

USER-SENSITIVE IMPACT ANALYSIS

Input:

1. *Field execution data*

	m_1	m_2	m_3	m_4	m_5	m_6
A1			X			X
B1		X	X		X	X
B2	X	X	X			
A2	X			X		
C1				X		X

2. *Change*

   C={m5, m6}

Output:

1. *Collective impact* = **60%**
2. *Affected users* = **100%**

Collective impact

- Percentage of executions through at least one changed method

  3/5 = 60%

Affected users

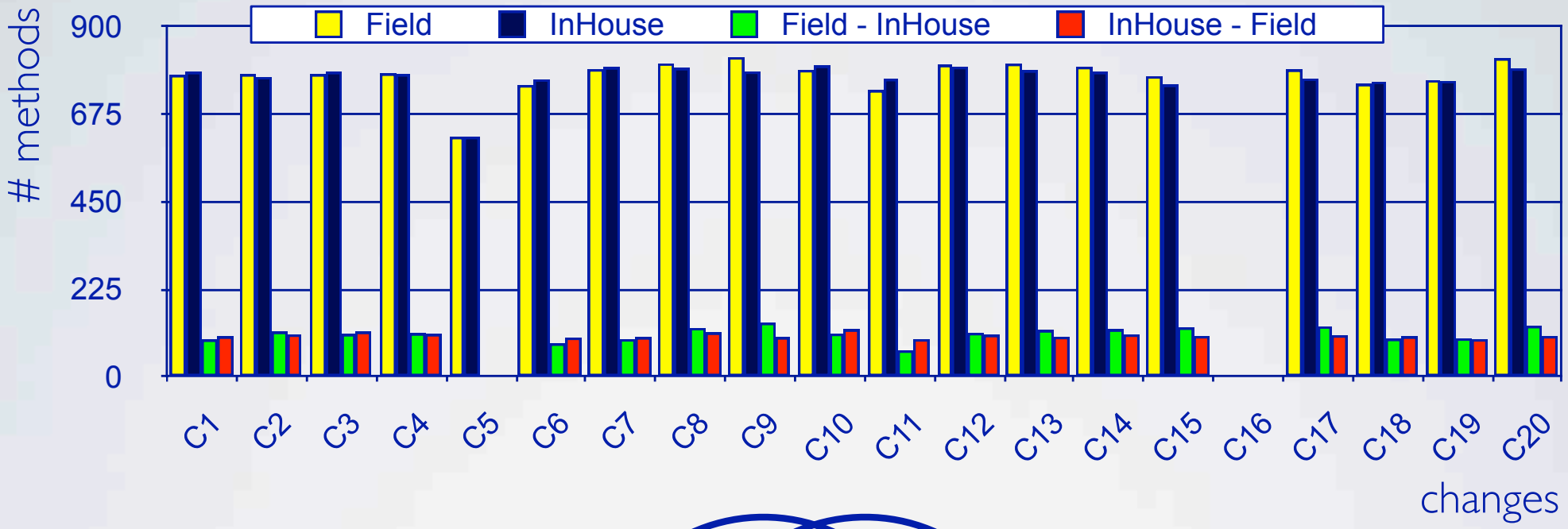
- Percentage of users that executed at least once one changed method

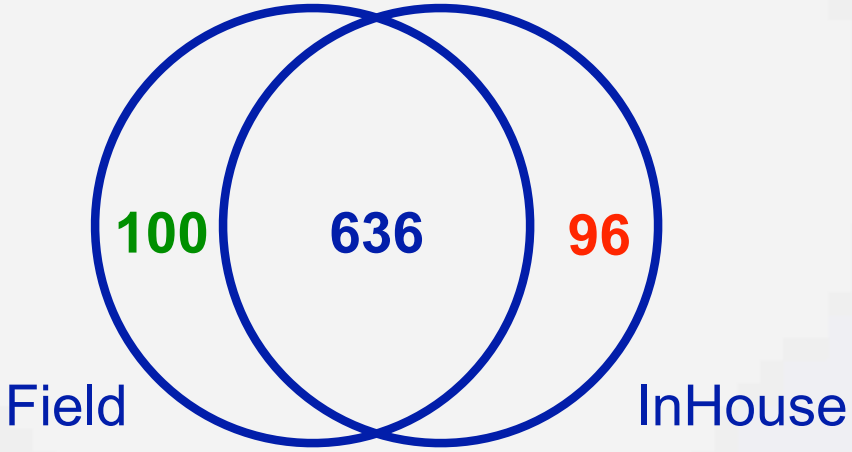
  3/3 = 100%

EMPIRICAL STUDY

- **Subject**:
  - JABA: Java Architecture for Bytecode Analysis (60 KLOC, 500 classes, 3K Methods)
- **Data**
  - Field data: 1,100 executions (14 users, 12 weeks)
  - In-house data: 195 test cases, 63% method coverage
  - Changes: 20 real changes extracted from JABA's CVS repository
- **Research question**: Does field data yield different results than in-house data?
- **Experimental setup**
  - Computed impact sets for the 20 changes <u>using field data</u> and <u>using in-house data</u>
  - Compared impact sets for the two datasets

RESULTS


Field
InHouse
Field - InHouse
InHouse - Field
methods
900
675
450
225
0
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
C12
C13
C14
C15
C16
C17
C18
C19
C20
changes

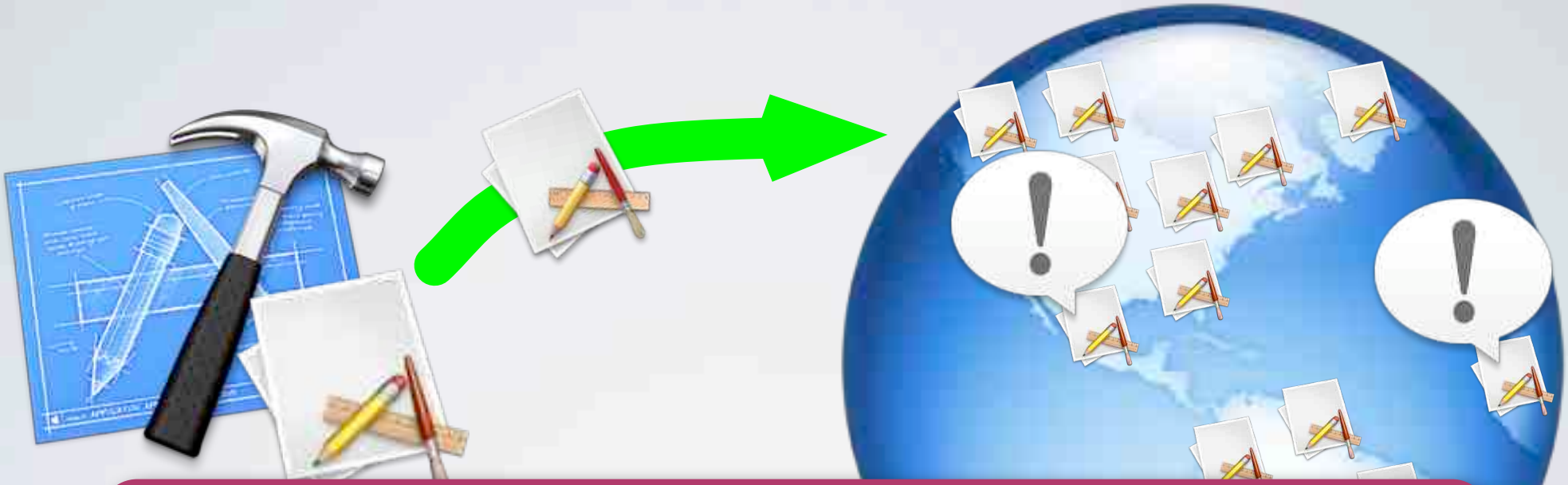

100
636
96
Field
InHouse

DEMO

"Gammatella: Visualizing Program-Execution Data for Deployed Software."
Jones et al., Information Visualization, 2004.

DEBUGGING FIELD FAILURES

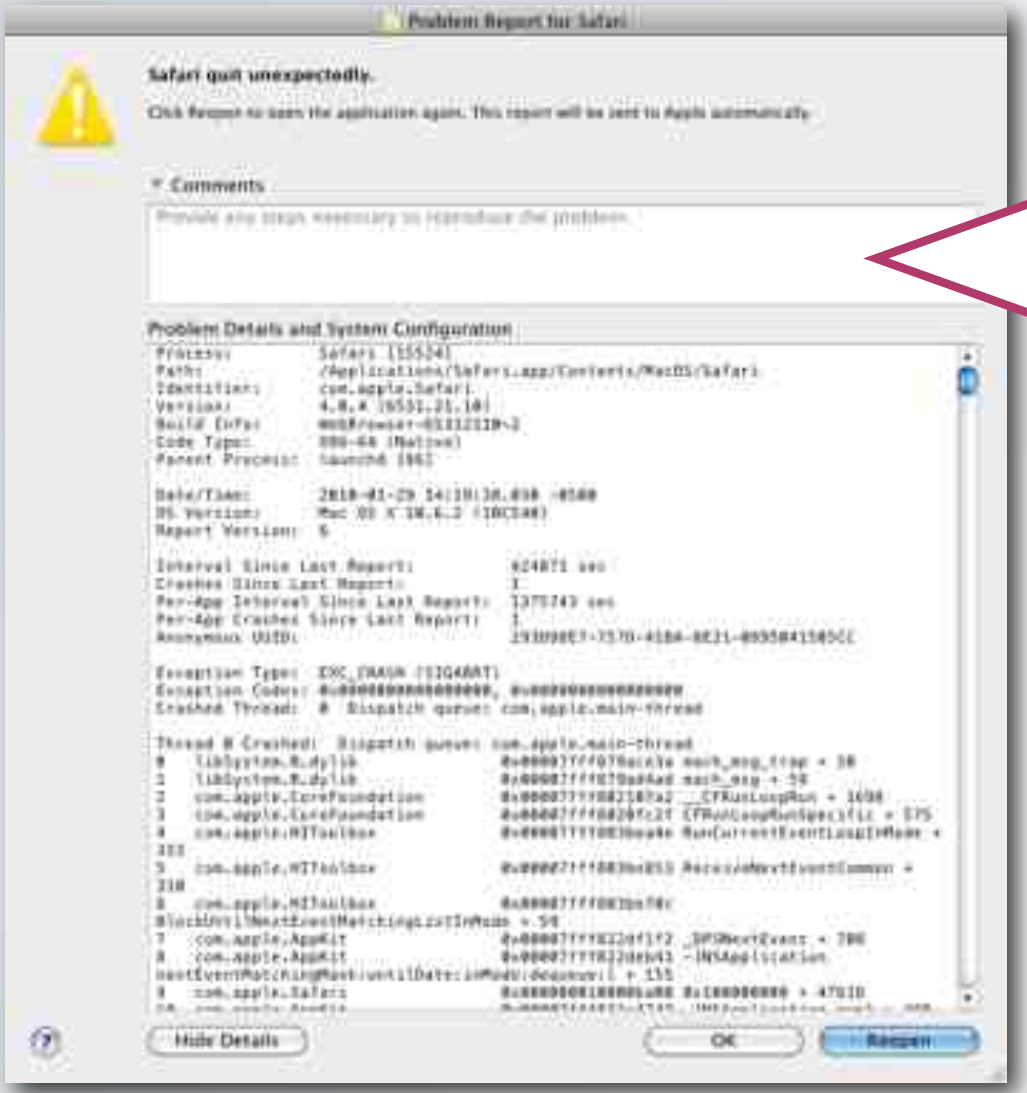
FIELD FAILURES



Field failures: Anomalous behaviors (or crashes) of deployed software that occur on user machines

- Difficult to debug
- Relevant to users

CURRENT PRACTICE



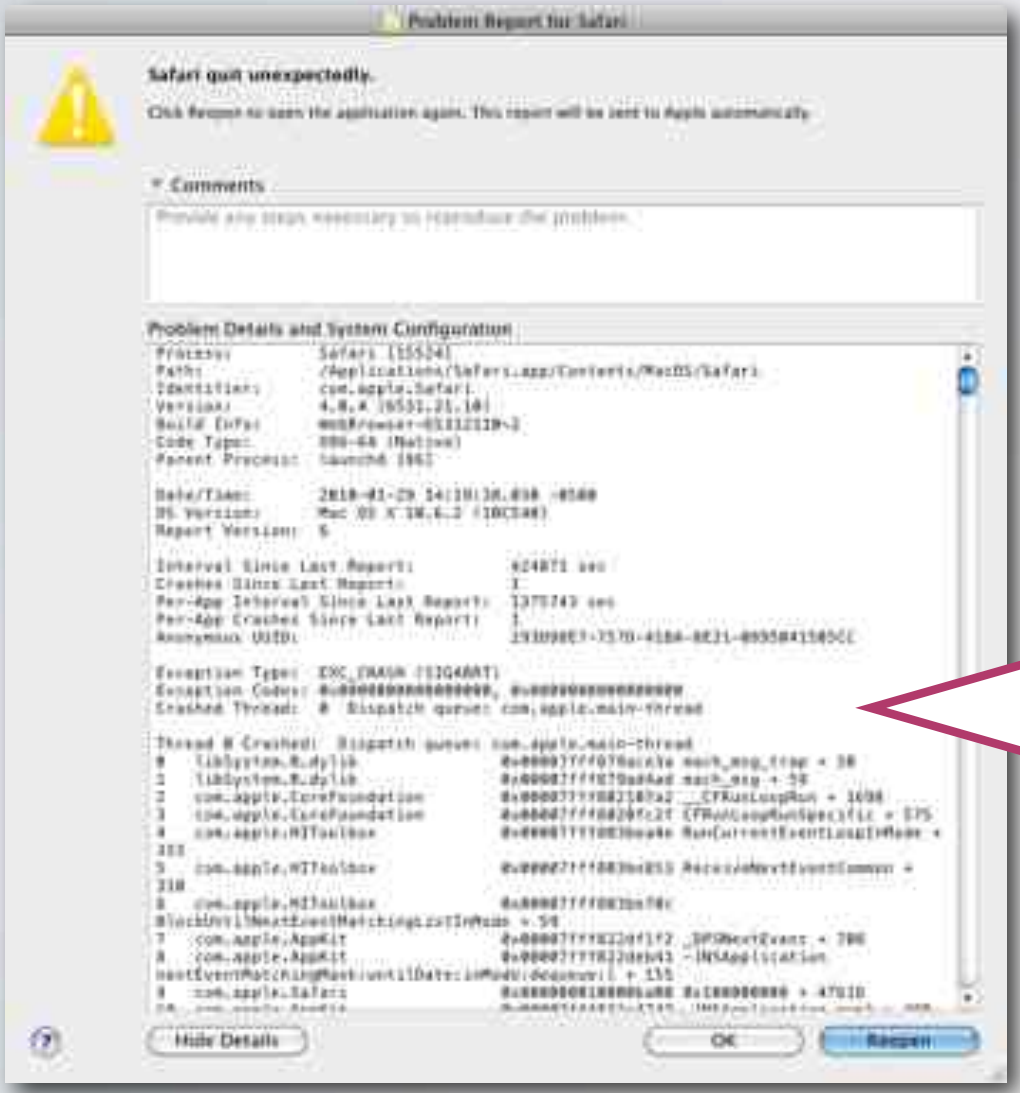
Ask the user

I opened my web browser.

Specifically, I clicked on the dock icon. It bounced twice before crashing.

Please help.

CURRENT PRACTICE



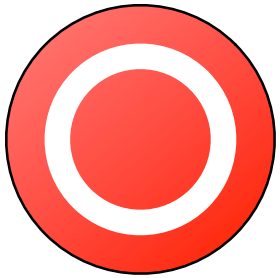
Gather static information

Difficult to reproduce the problem

Only locations directly correlated with the failure

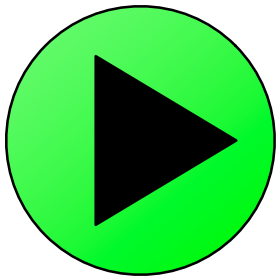
OUR SOLUTION

Record
failing executions
in the field



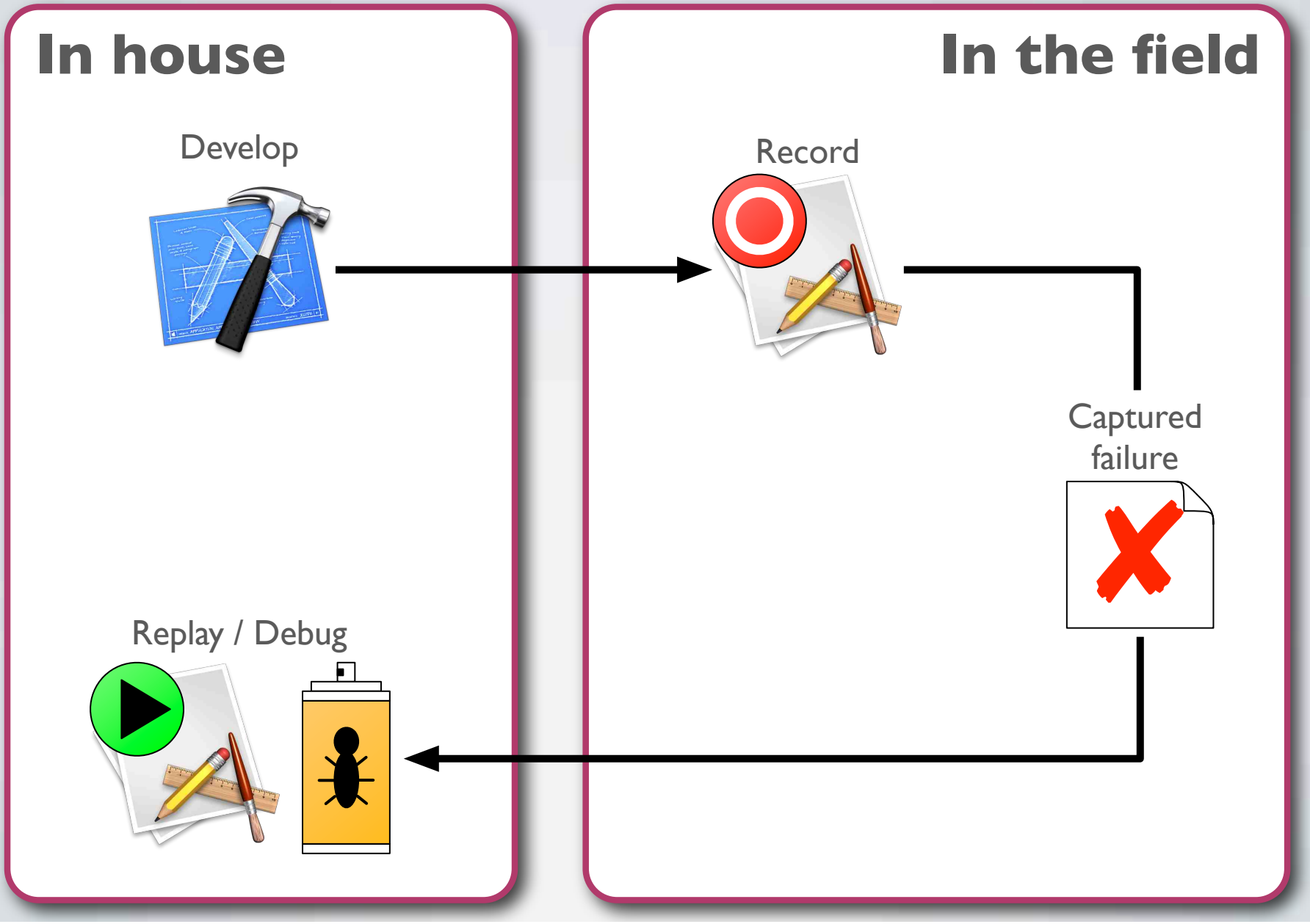
+

Replay
failing executions
in house

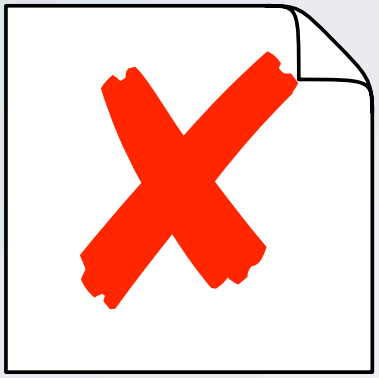


Debug
field failures
effectively

USAGE SCENARIO



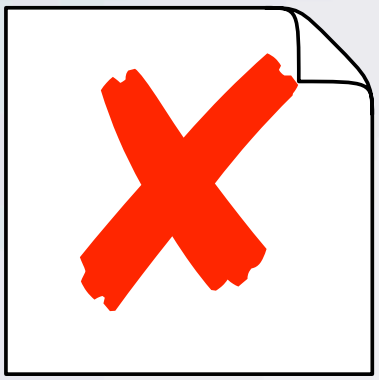
CHALLENGES



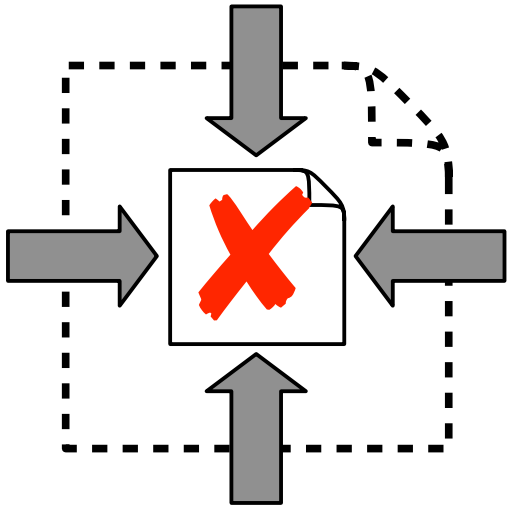
Large in size

Contain sensitive information

CHALLENGES



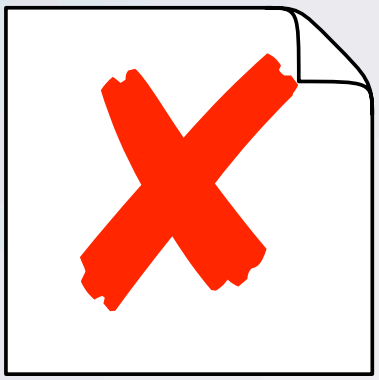
Large in size



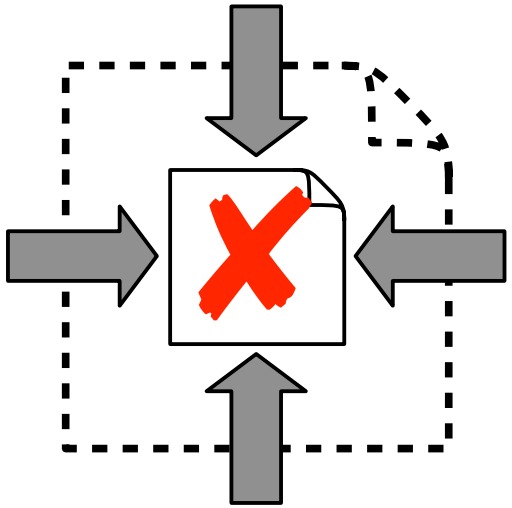
Minimize

Contain sensitive information

CHALLENGES



Large in size

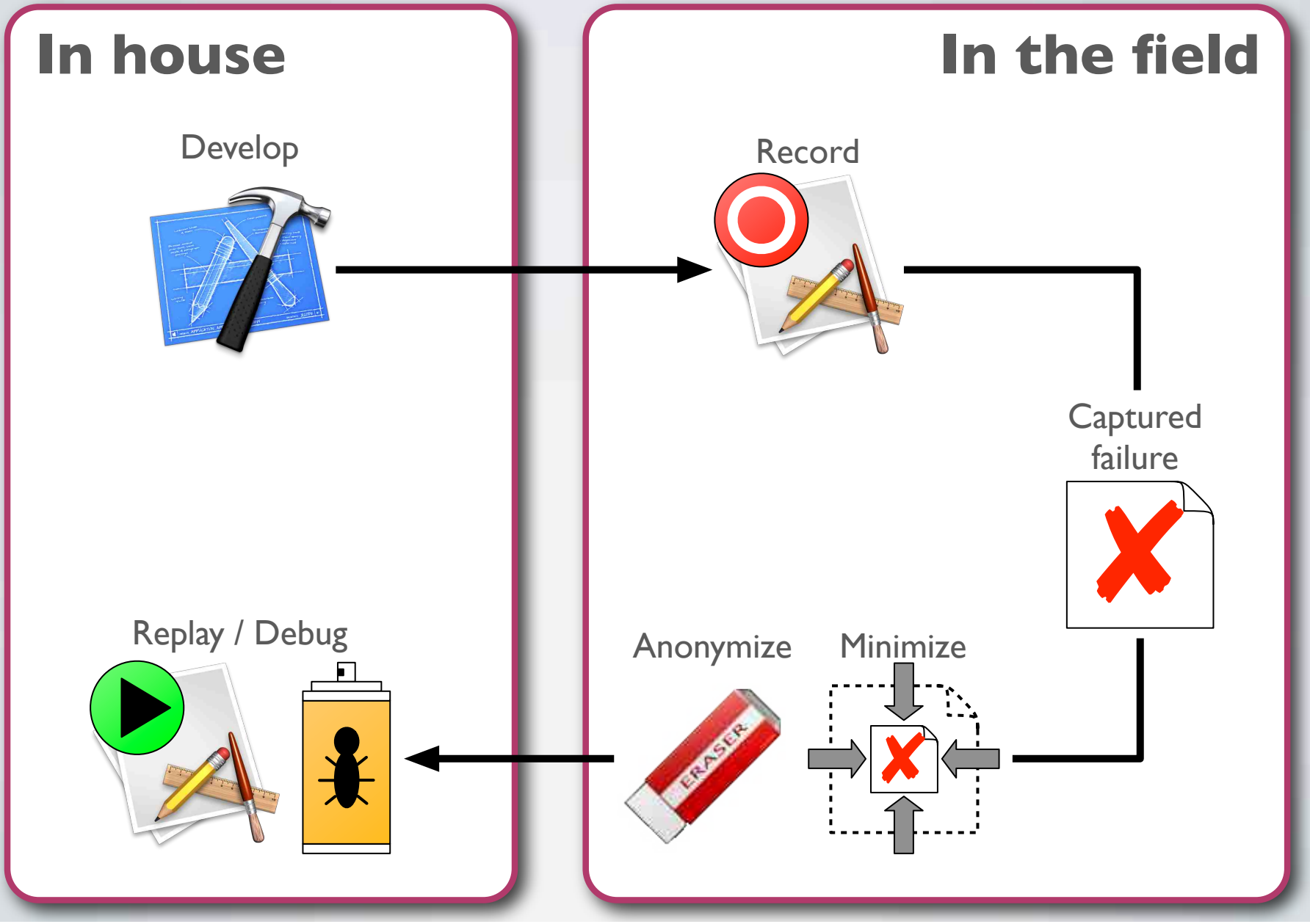


Minimize

Contain sensitive information

Anonymize

USAGE SCENARIO



EVALUATION (PRACTICALITY)

Research question 1:

- does the technique impose an acceptable overhead?

Subjects:

- several cpu intensive applications (e.g., bzip, gcc)

Results:

- negligible overheads (i.e., less than 10%)
- data size is acceptable (application dependent)

"A Technique for Enabling and Supporting Debugging of Field Failures"
Clause and Orso, ICSE 2007.

EVALUATION (FEASIBILITY)

Research question 2:

- can the technique produce minimized executions that can be used to debug the original failure?

Subject: Pine email and news client

- two real field failures
- 20 failing executions, 10 per failure

Results:

- execution reduced to less than 10% in size
- all failures reproducible

"A Technique for Enabling and Supporting Debugging of Field Failures"
Clause and Orso, ICSE 2007.

EVALUATION (EFFECTIVENESS)

Research question 3:

- How much information about the original inputs is revealed?

Subjects: NanoXML, htmlparser, Printtokens, Columba

- 20 faults overall
- inputs from 100 bytes to 5MB in size
- all inputs considered sensitive

Results:

- Anonymized inputs revealed between 2% and 60% of the information in the original inputs

"Camouflage: Automated Anonymization of Field Data."
Clause and Orso, GT Tech Report, March 2010.

RQ3: EFFECTIVENESS

NANOXML

```
<!DOCTYPE Foo [
   <!ELEMENT Foo (ns:Bar)>
   <!ATTLIST Foo
       xmlns CDATA #FIXED 'http://nanoxml.n3.net/bar'
       a     CDATA #REQUIRED>

   <!ELEMENT ns:Bar (Blah)>
   <!ATTLIST ns:Bar
       xmlns:ns CDATA #FIXED 'http://nanoxml.n3.net/bar'>

   <!ELEMENT Blah EMPTY>
   <!ATTLIST Blah
       x    CDATA #REQUIRED
       ns:x CDATA #REQUIRED>
]>
<!-- comment -->
<Foo a='very' b='secret' c='stuff'>vaz
   <ns:Bar>
       <Blah x="1" ns:x="2"/>
   </ns:Bar>
</Foo>
```

RQ3: EFFECTIVENESS

NANOXML

```
<!DOCTYPE     [
   <!                    >
   <!ATTLIST
                  #FIXED '                          '
                           >

   <!E                   >
   <!ATTLIST
                     #FIXED '                          '>

   <!E                  >
   <!ATTLIST
                  #
         :        #        >
]>
<!--         -->
<     ='    '  ='      '  ='     '>
   <  :   >
       <      =" "   : =" "/>
   </  :
```

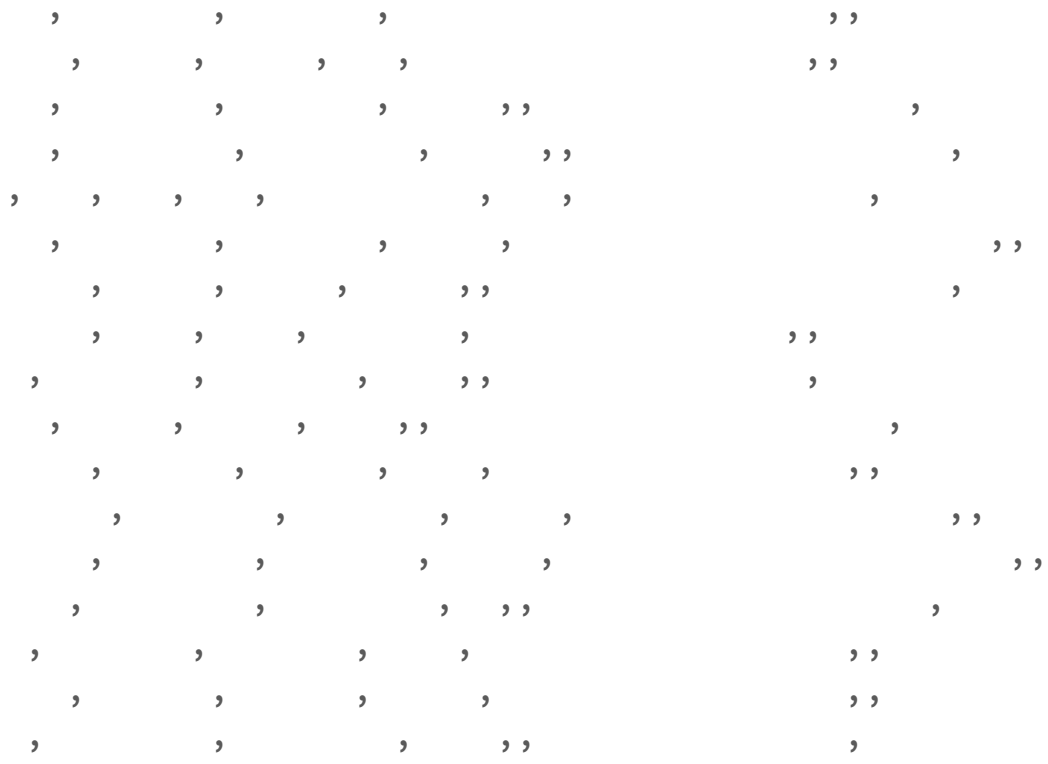
RQ3: EFFECTIVENESS

COLUMBA

```
Wayne,Bartley,Bartley,Wayne,wbartly@acp.com,,
Ronald,Kahle,Kahle,Ron,ron.kahle@kahle.com,,
Wilma,Lavelle,Lavelle,Wilma,,lavelle678@aol.com,
Jesse,Hammonds,Hammonds,Jesse,,hamj34@comcast.com,
Amy,Uhl,Uhl,Amy,uhla@corp1,com,uhla@gmail.com,
Hazel,Miracle,Miracle,Hazel,hazel.miracle@corp2.com,,
Roxanne,Nealy,Nealy,Roxie,,roxie.nearly@gmail.com,
Heather,Kane,Kane,Heather,kaneh@corp2.com,,
Rosa,Stovall,Stovall,Rosa,,sstoval@aol.com,
Peter,Hyden,Hyden,Pete,,peteh1989@velocity.net,
Jeffrey,Wesson,Wesson,Jeff,jwesson@corp4.com,,
Virginia,Mendoza,Mendoza,Ginny,gmendoza@corp4.com,,
Richard,Robledo,Robledo,Ralph,ralphrobledo@corp1.com,,
Edward,Blanding,Blanding,Ed,,eblanding@gmail.com,
Sean,Pulliam,Pulliam,Sean,spulliam@corp2.com,,
Steven,Kocher,Kocher,Steve,kocher@kocher.com,,
Tony,Whitlock,Whitlock,Tony,,tw14567@aol.com,
Frank,Earl,Earl,Frankie,,,
Shelly,Riojas,Riojas,Shelly,srojas@corp6.com,,
```

RQ3: EFFECTIVENESS

COLUMBA



RQ3: EFFECTIVENESS

HTMLPARSER

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/
xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<title>james clause @ gatech | home</title>

<style type="text/css" media="screen" title="">
<!--/*--><![CDATA[<!--*/

  body {
    margin: 0px;
...

/*]]>*/-->
</style>
</head>
<body>
  ...
</body>
```

RQ3: EFFECTIVENESS

HTMLPARSER

```
<                                        >
<!                        //    //              //           //           /   /
        /    /             >

<                   //           /     /                      >
<     >
<      >                              </      >

<                  /                              >
<
```

RQ3: EFFECTIVENESS

HTMLPARSER

```
<                                        >
<!                        //    //              //           //           /   /
        /    /             >

<                   //           /     /                      >
<     >
<      >                            </      >

<                  /                            >
<
```

The portions of the inputs that remain after anonymization tend to be structural in nature and therefore are safe to send to developers

CONCLUDING REMARKS

ADDRESSING THE ISSUES

- **Issue #1: Communication**
- **Issue #2: Mismatch in assumptions**
- **Issue #3: Infrastructure**
- **Issue #4: Narrow focus of some MSA research**

ADDRESSING THE ISSUES

- **Issue #1: Communication**
  - Reaching out
  - More common events
  - Challenge
- **Issue #2: Mismatch in assumptions**
- **Issue #3: Infrastructure**
- **Issue #4: Narrow focus of some MSA research**

ADDRESSING THE ISSUES

- **Issue #1: Communication**
- **Issue #2: Mismatch in assumptions**
  - Many similarities and potential synergies
  - Opportunity for defining new (or specialized) analyses
  - Opportunity for performing more thorough evaluations
- **Issue #3: Infrastructure**
- **Issue #4: Narrow focus of some MSA research**

ADDRESSING THE ISSUES

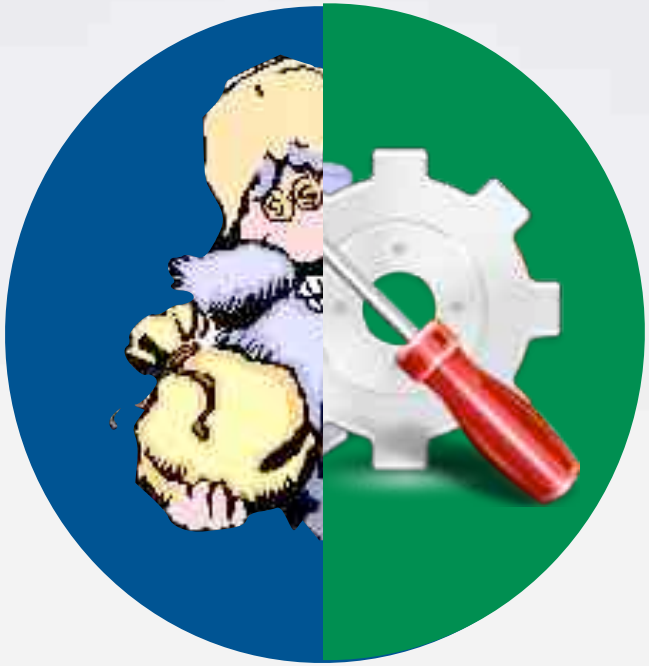
- **Issue #1: Communication**
- **Issue #2: Mismatch in assumptions**
- **Issue #3: Infrastructure**
  - Related to communication
  - Reciprocal help
- **Issue #4: Narrow focus of some MSA research**

ADDRESSING THE ISSUES

- **Issue #1: Communication**
- **Issue #2: Mismatch in assumptions**
- **Issue #3: Infrastructure**
- **Issue #4: Narrow focus of some MSA research**
  - Go beyond the analysis of "easy" information in the repositories
  - Consider all aspects of software, both static and dynamic
  - Consider both in-vitro and in-vivo data

IN CONCLUSION, BETTER TOGETHER?

IN CONCLUSION, BETTER TOGETHER?



Techniques for analyzing/mining a program in all of its aspects, static and dynamic, and throughout its lifetime

ACKNOWLEDGEMENTS

- Taweesup Apiwattanapong
- James Clause
- Mary Jean Harrold
- James Jones
- Donglin Liang
- Dick Lipton