Estimating the Accuracy of Dynamic Change-impact Analysis using Sensitivity Analysis

Haipeng Cai♦

Raul Santelices ♦

Tianyu Xu*

♦ University of Notre Dame, USA

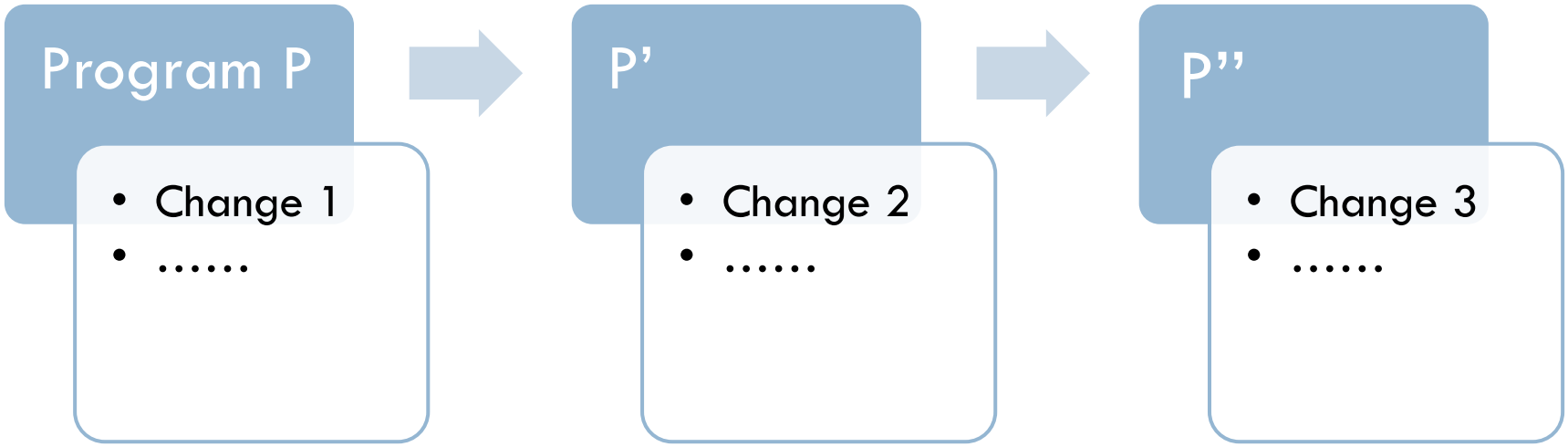
* Fudan University, China

UNIVERSITY OF NOTRE DAME
College of Engineering

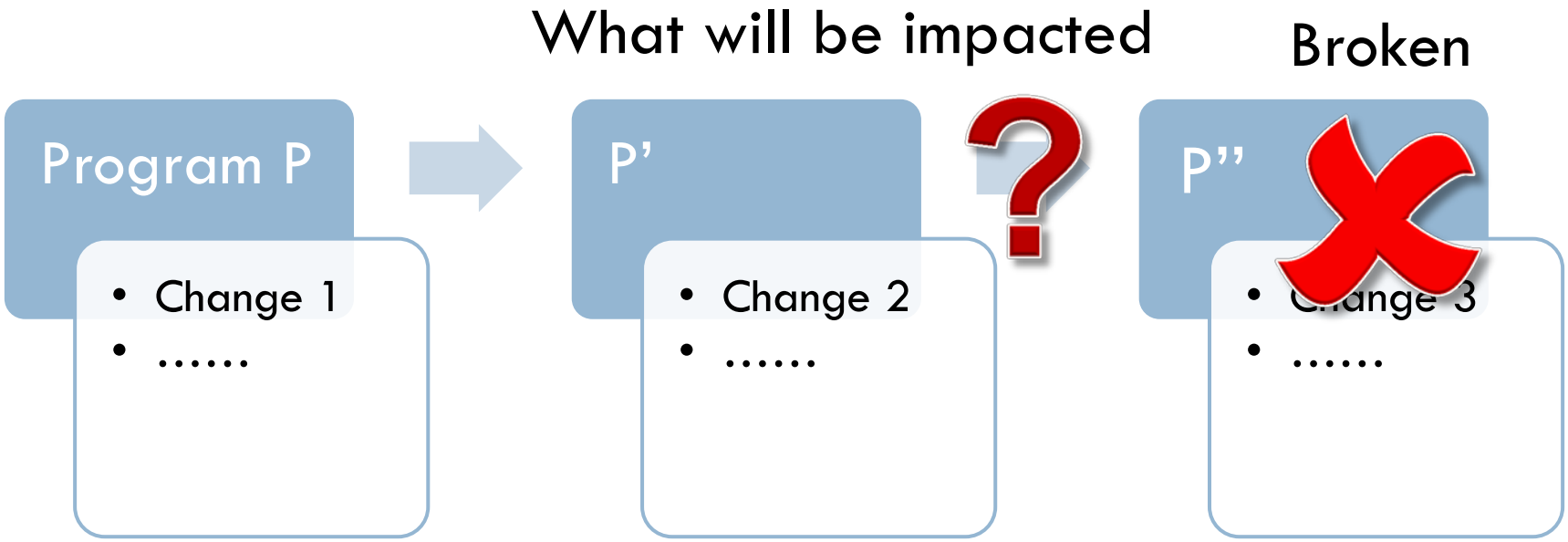
SERE 2014

Supported by ONR Award N000141410037

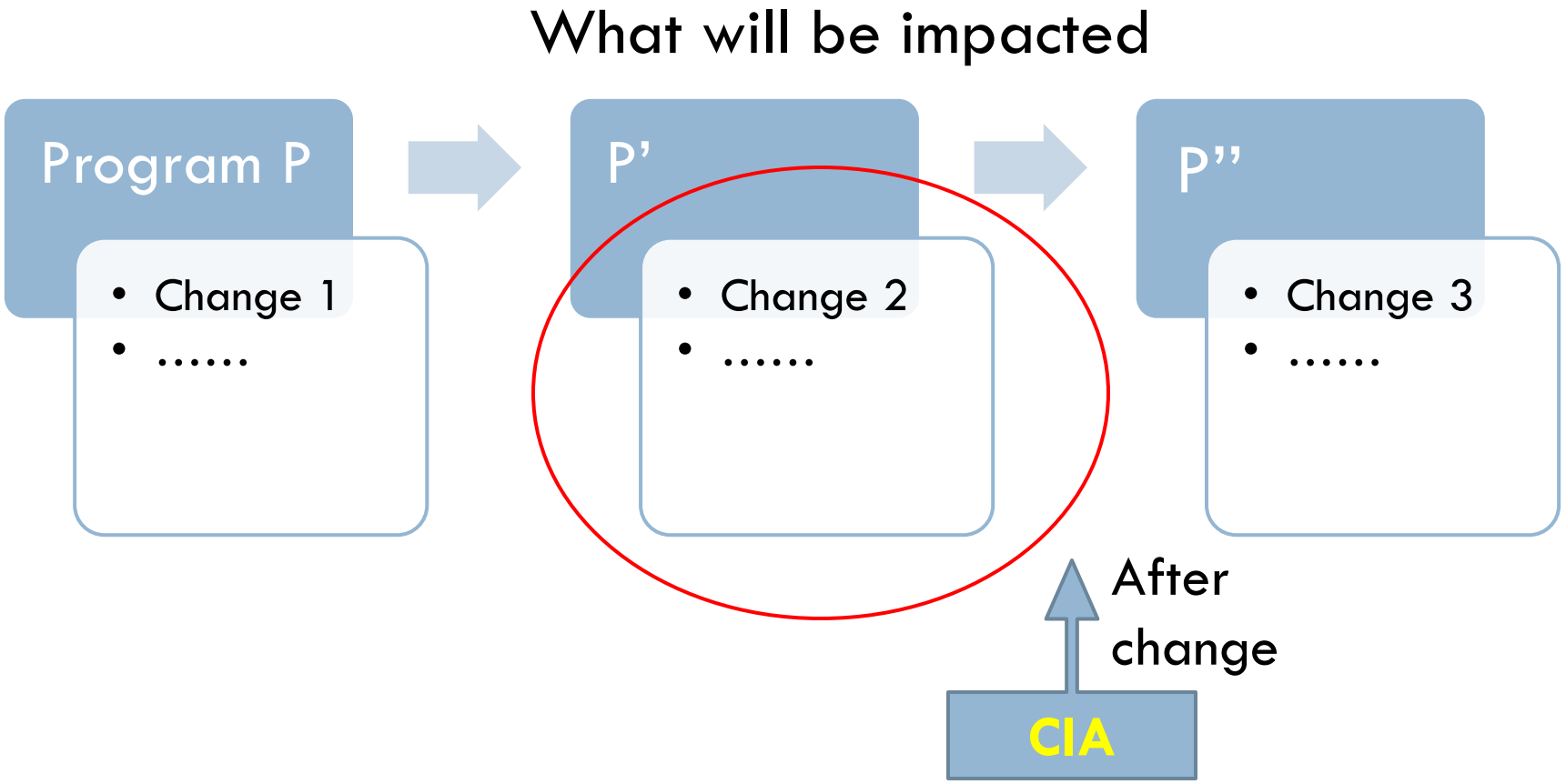
Software keeps changing



Software keeps changing

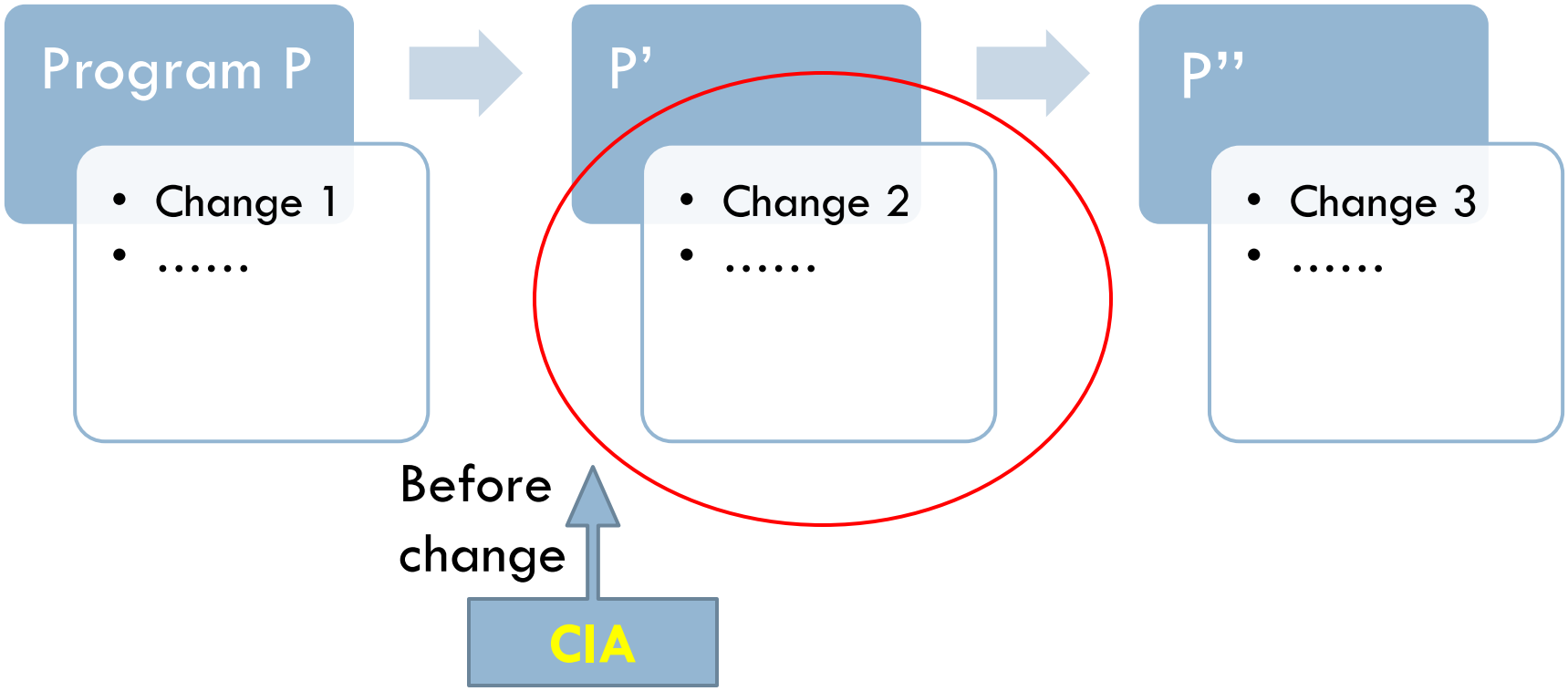


Change impacts need be analyzed



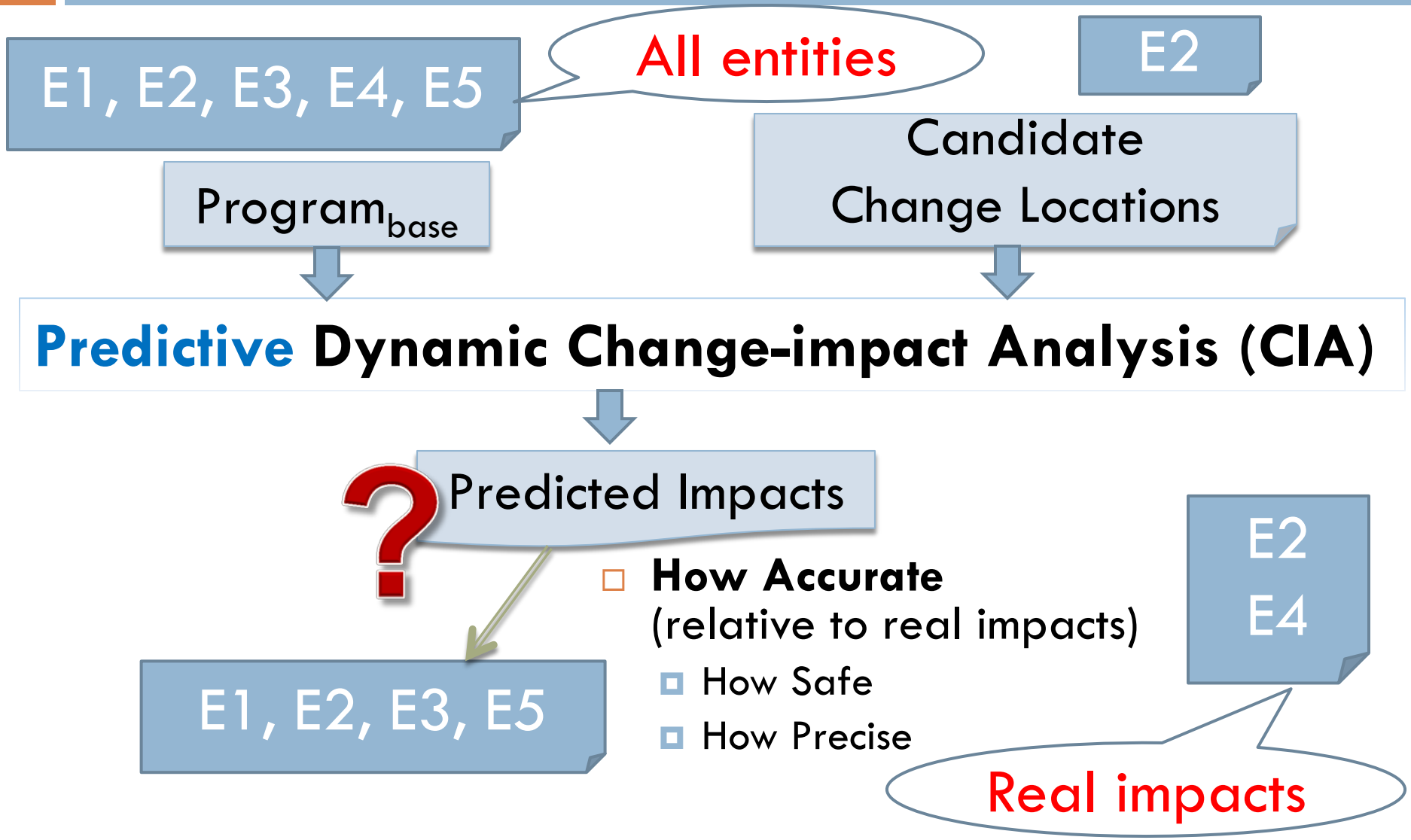
Dynamic Change-impact Analysis (CIA)

Change impacts need be analyzed

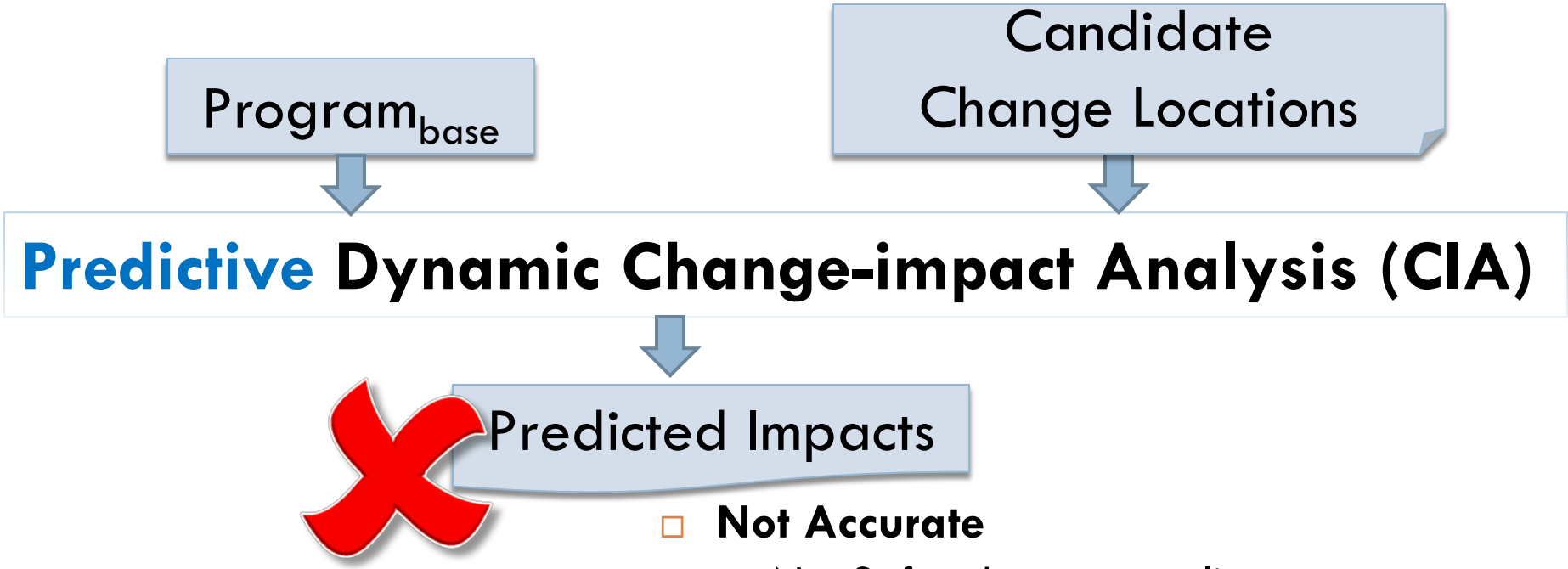


Predictive Dynamic Change-impact Analysis (CIA)

How accurate is it ?



Why study CIA accuracy ?



- **Not Accurate**
  - Not Safe : threaten quality
  - Not Precise : waste time
- (If not accurate) **Motivate developing better techniques**

Which techniques to study ?

- Our target
  - Method level
  - The most cost-effective

Predictive Dynamic Change-impact Analysis (CIA)

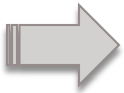
PI/EAS

- PathImpact (PI)
  [Law & Rothermel *ICSE'03*]
- CoverageImpact (CI)
  [Orso & Apiwattanapong & Harrold *FSE'03*]
- Execute-After-Sequences (EAS)
  [Apiwattanapong & Orso & Harrold *ICSE'05*]
- InfluenceDynamic (InfDyn)
  [Breech & Tegtmeyer & Pollock *ICSM'06*]

Much less precise

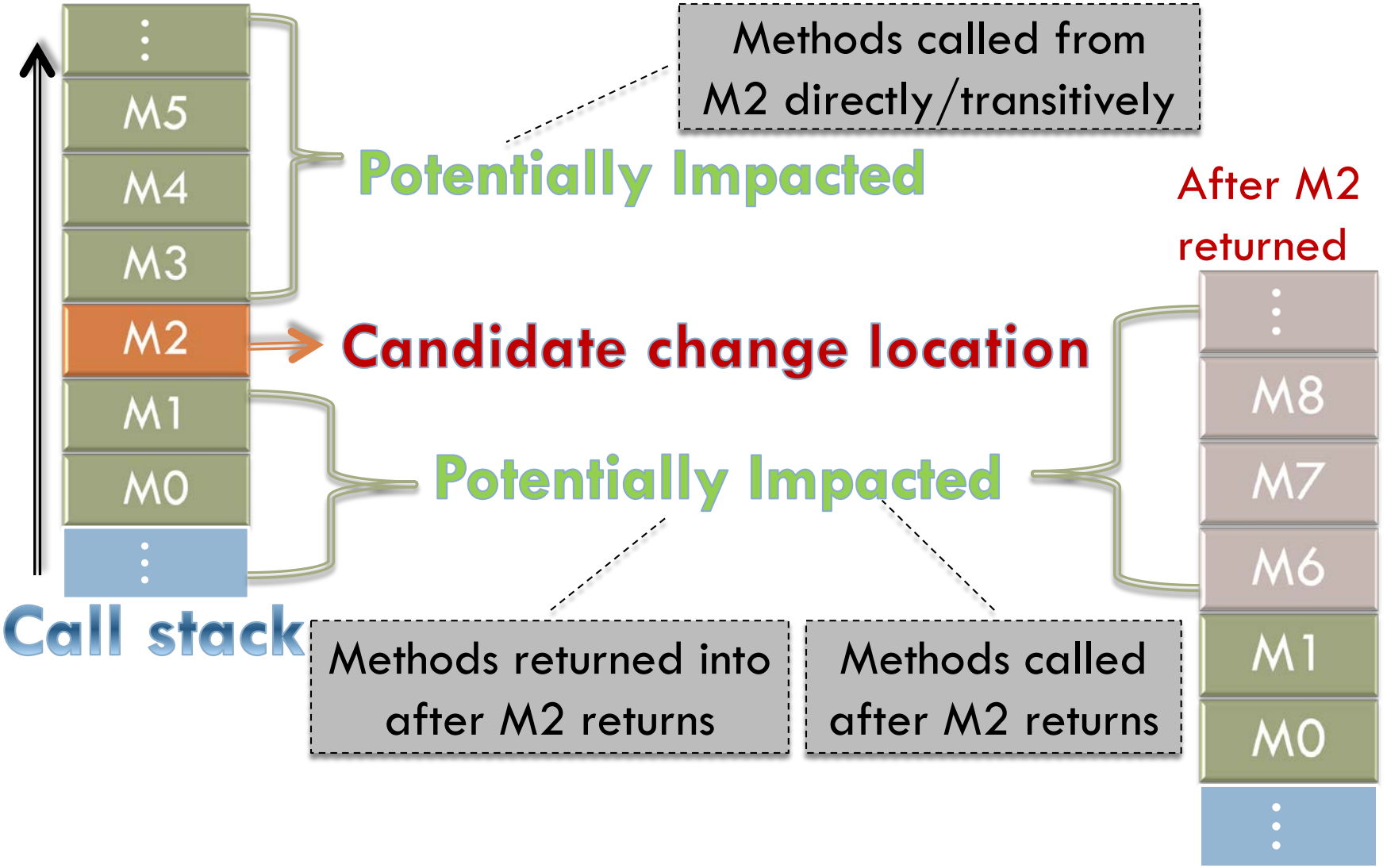
Little more precise
much more expensive

Outline



- Background – PI/EAS
- Methodology
  - Sensitivity Analysis
  - Execution Differencing
- Results
- Contributions

How PI/EAS works



How PI/EAS works

- Example code

```
1  public class A {
2      static int M1(int f, int z) {
3          M2(f+z);
4          return new B().M3(f,1); }
5      void M2(int m) {
6          if (m > 0)
7              C.M5(); }}
```

```
8  public class B {
9      public static int t=0;
10     int M3(int a, int b) {
11         int n = b*b - a;
12         return n; }
13     static void M4() {
14         t = 10; }}
```

```
15 public class C {
16     public static boolean M5() {
17         return B.t > 10; }
18
19     public static void M0() {
20         if (A.M1(4,-3) > 0)
21             B.M4(); }}
```

- Example execution

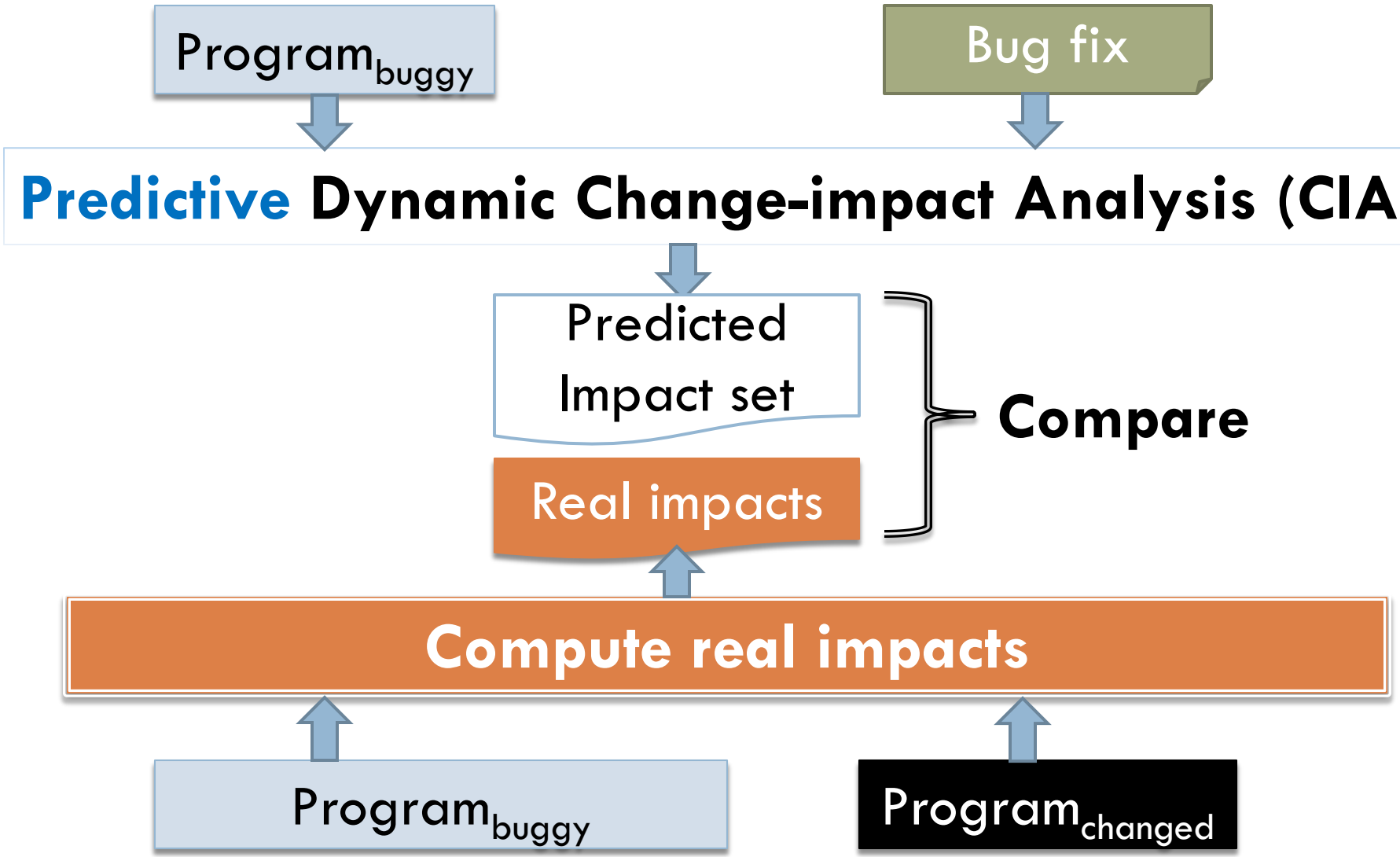
- Suppose changes will be in method M2
  - M5, M3 potentially impacted: entered after M2 entered
  - M1, M0 potentially impacted: returned into after M2 returned
  - Impact set = {M0,M1,M2,M3,M5}

Outline

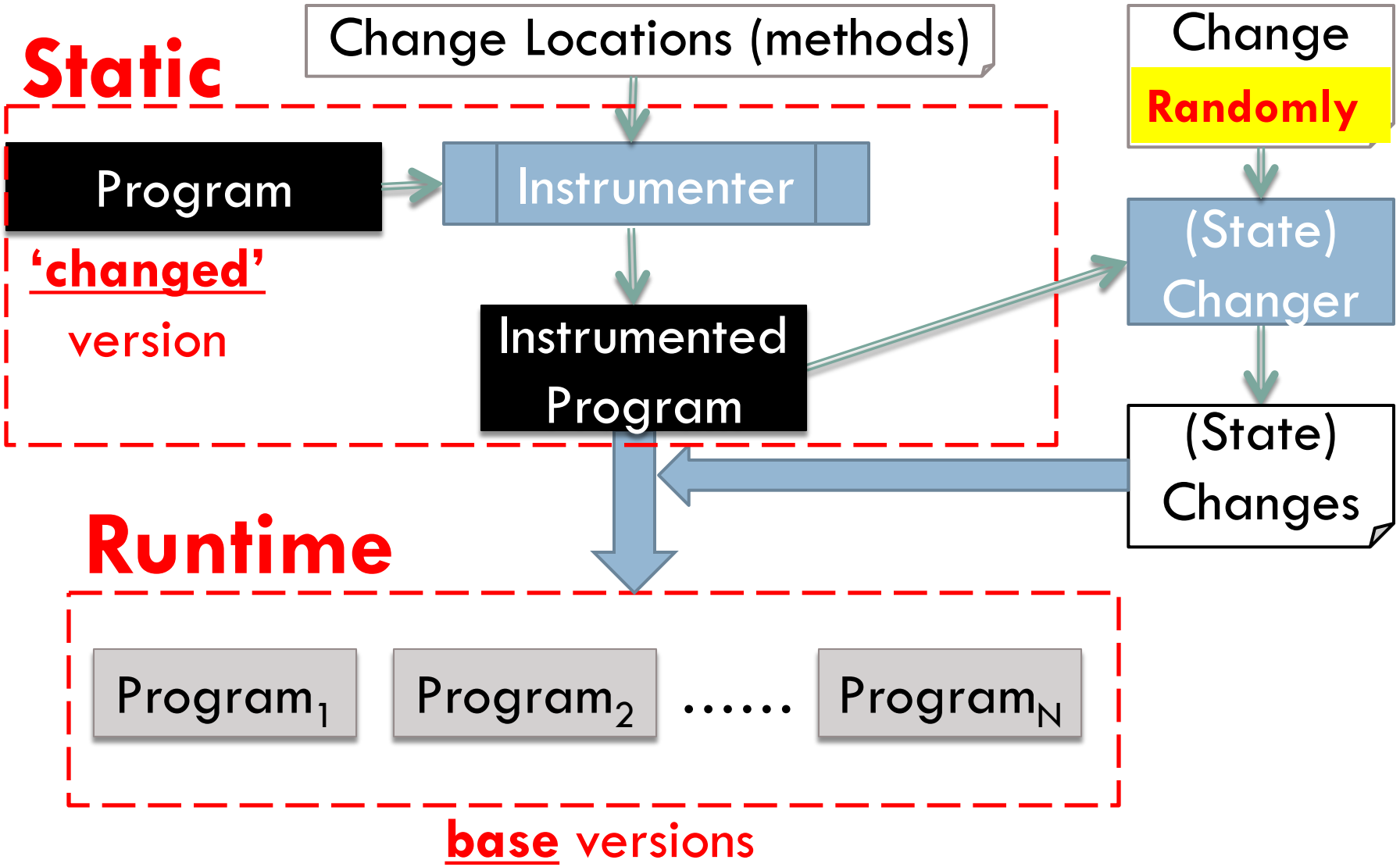
- Background: PI/EAS
- Methodology
  - Sensitivity Analysis
  - Execution Differencing
- Results
- Contributions

Study as many changes as possible

Bug fix is a common type of change



Efficient Sensitivity Analysis



Execution Differencing


```
1 public class A {
2     static int M1(int f, int z) {
3         M2(f+z);
4         return new B().M3(f,1); }
5     void M2(int m) {
6         if (m > 0)      // m < 0
7             C.M5(); }}
```

```
8 public class B {
9     public static int t=0;
10    int M3(int a, int b) {
11        int n = b*b - a;
12        return n; }
13    static void M4() {
14        t = 10; }}
```

```
15 public class C {
16    public static boolean M5() {
17        return B.t > 10; }
18
19    public static void M0() {
20        if (A.M1(4,-3) > 0)
21            B.M4(); }}
```

Execution History

Base version

Line no.	Value
20	False
6	True
11	-3
12	-3
7	
17	False
4	-3

Results (statements):
6
7
17

Changed version

Line no.	Value
20	False
6	False
11	-3
12	-3
-	
-	
4	-3

Execution Differencing


```
1 public class A {
2     static int M1(int f, int z) {
3         M2(f+z);
4         return new B().M3(f,1); }
5     void M2(int m) {
6         if (m > 0)      // m < 0
7             C.M5(); }}
```

```
8 public class B {
9     public static int t=0;
10    int M3(int a, int b) {
11        int n = b*b - a;
12        return n; }
13    static void M4() {
14        t = 10; }}
```

```
15 public class C {
16    public static boolean M5() {
17        return B.t > 10; }
18
19    public static void M0() {
20        if (A.M1(4,-3) > 0)
21            B.M4(); }}
```

Execution History

Base version

Line no.	Value
20	False
6	True
11	-3
12	-3
7	
17	False

Results (statements):
6
7
17

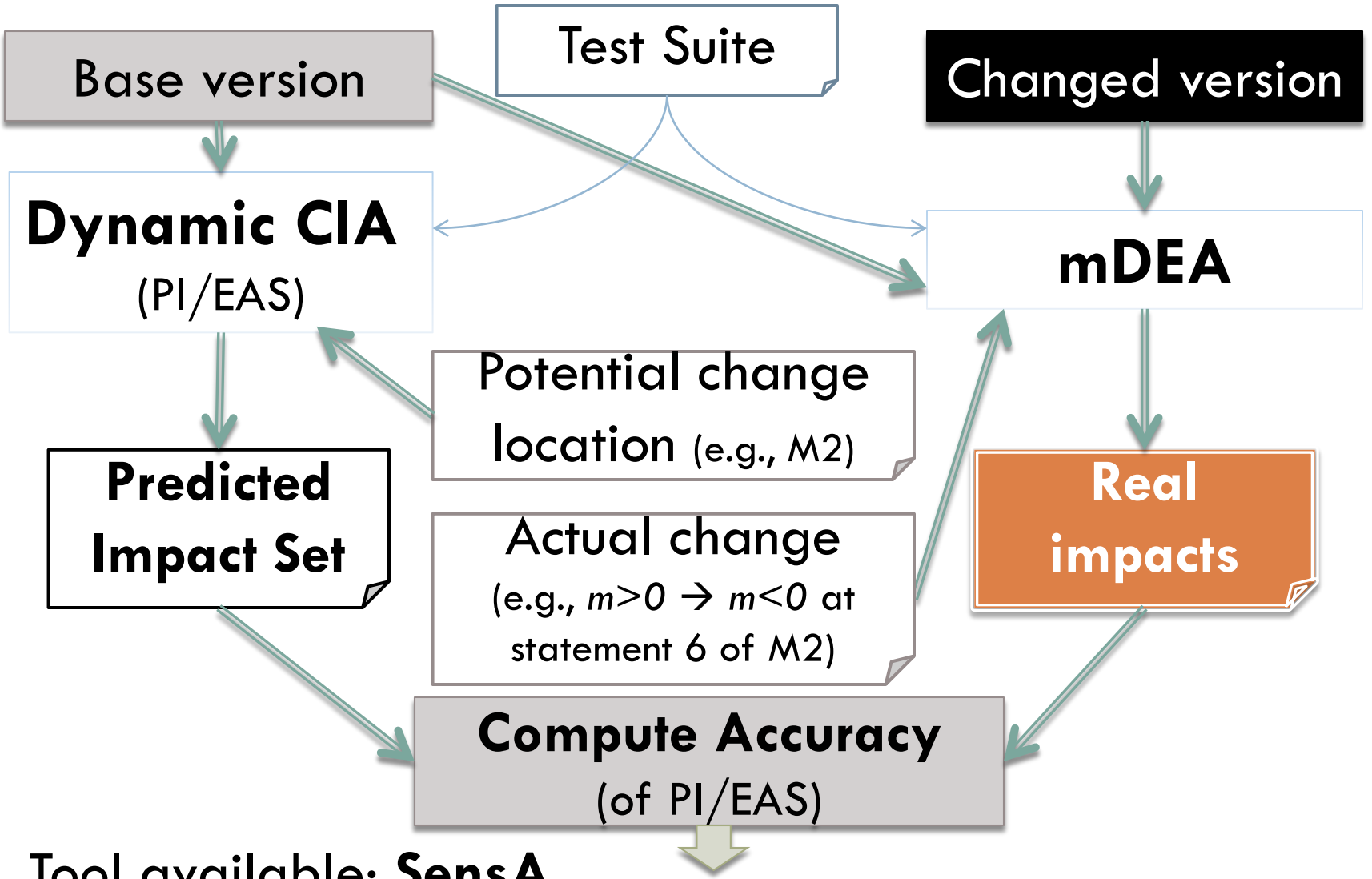
Results (methods):
M2
M5

Changed version

Line no.	Value
20	False
6	False
11	-3
12	-3
-	
-	

Method-level Differential Execution Analysis (**mDEA**)

Accuracy estimation



Tool available: **SensA**

Outline

- Background: PI/EAS
- Methodology
  - Sensitivity Analysis
  - Execution Differencing
- Results
- Contributions

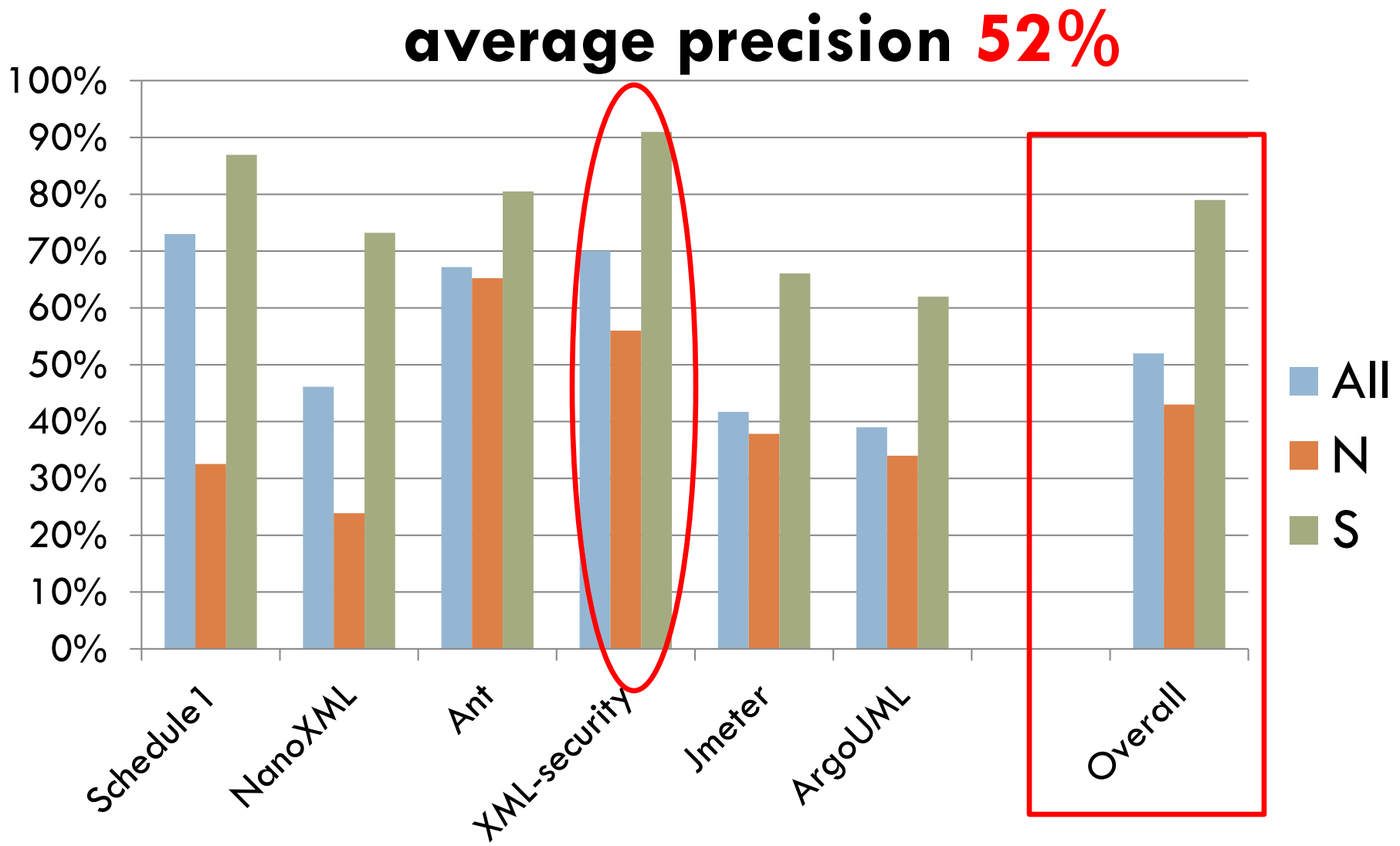
Subject programs and statistics

Subject	Description	Lines of Code	Methods	Tests
Schedule1	Priority Scheduler	290	24	2,650
NanoXML	XML parser	3,521	282	214
Ant	Java project build tool	18,830	1,863	112
XML-Security	Encryption library	22,361	1,928	92
JMeter	Performance monitor	35,547	3,054	79
ArgoUML	UML Modeling tool	102,400	8,856	211

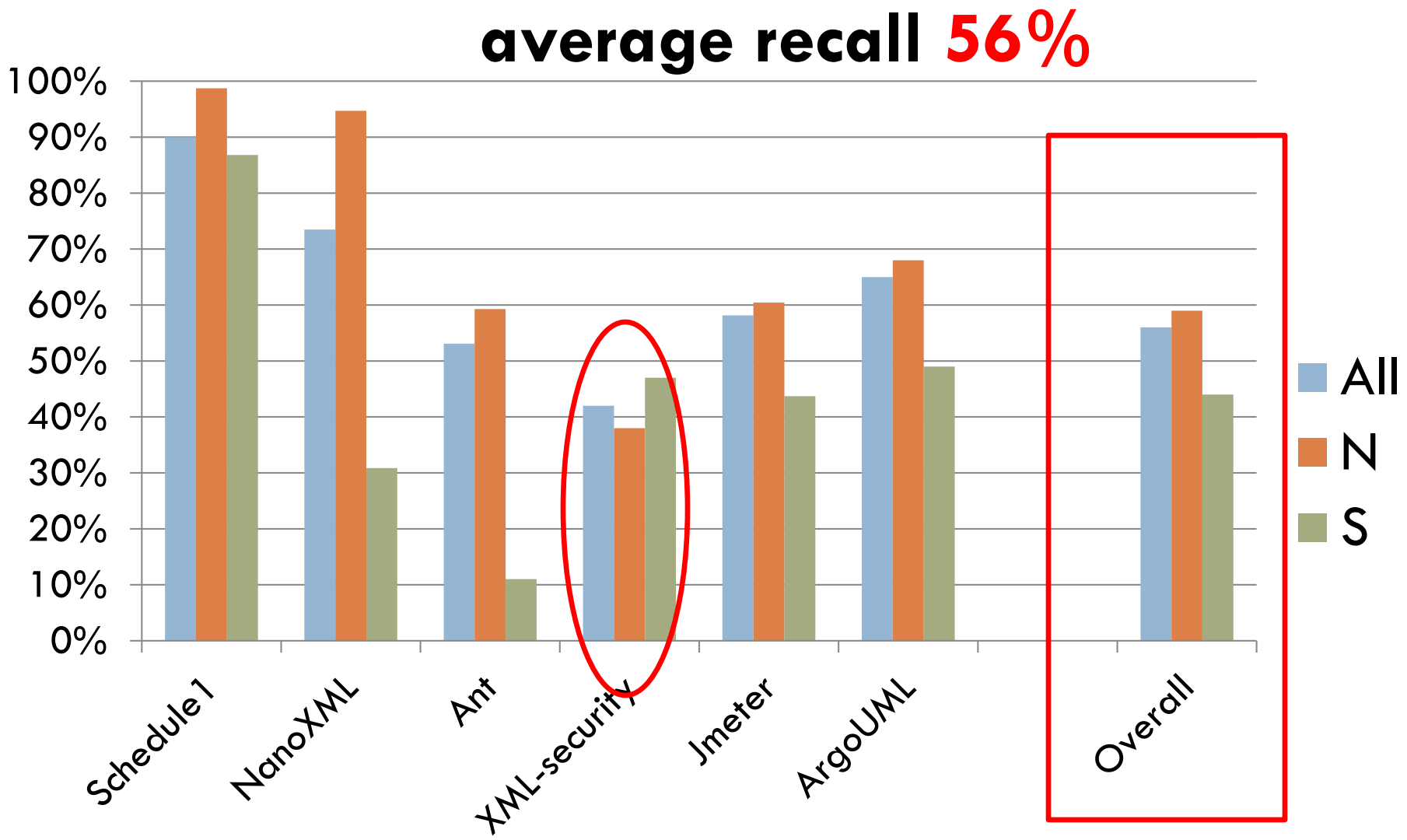
Metrics

- Impact sets (predicted and real)
- Number of false positives (FP) & false negatives (FN)
- Accuracy (F1)
  - Precision
  - Recall
  - F1 = 2x(precision x recall)/(precision + recall)
- Result classification
  - ***All**: both S and N*
    - ***S**hortening* (base execution is over 50% shorter)
    - ***N**ormal* (otherwise)

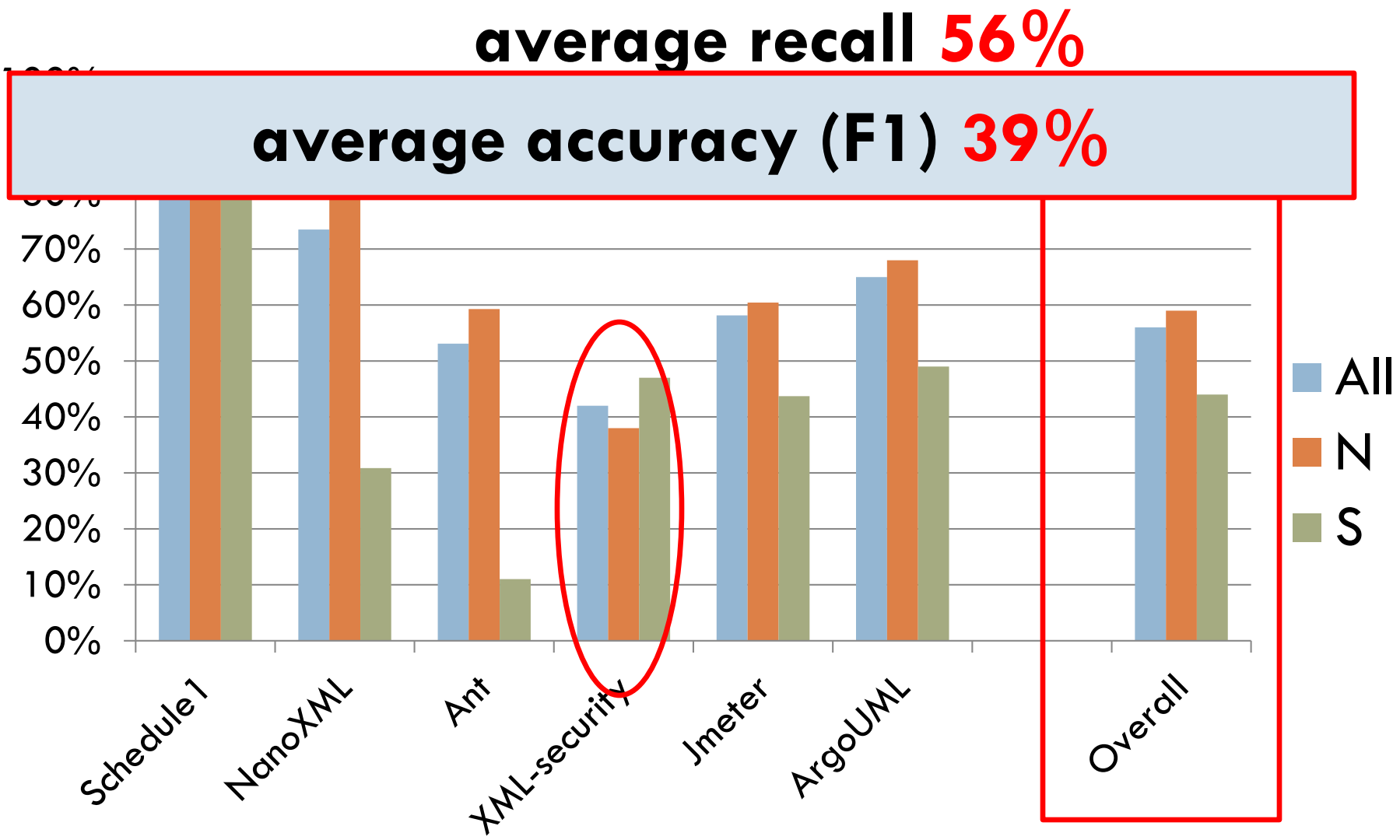
Precision


average precision 52%
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Schedule1
NanoXML
Ant
XML-security
Jmeter
ArgoUML
Overall
All
N
S

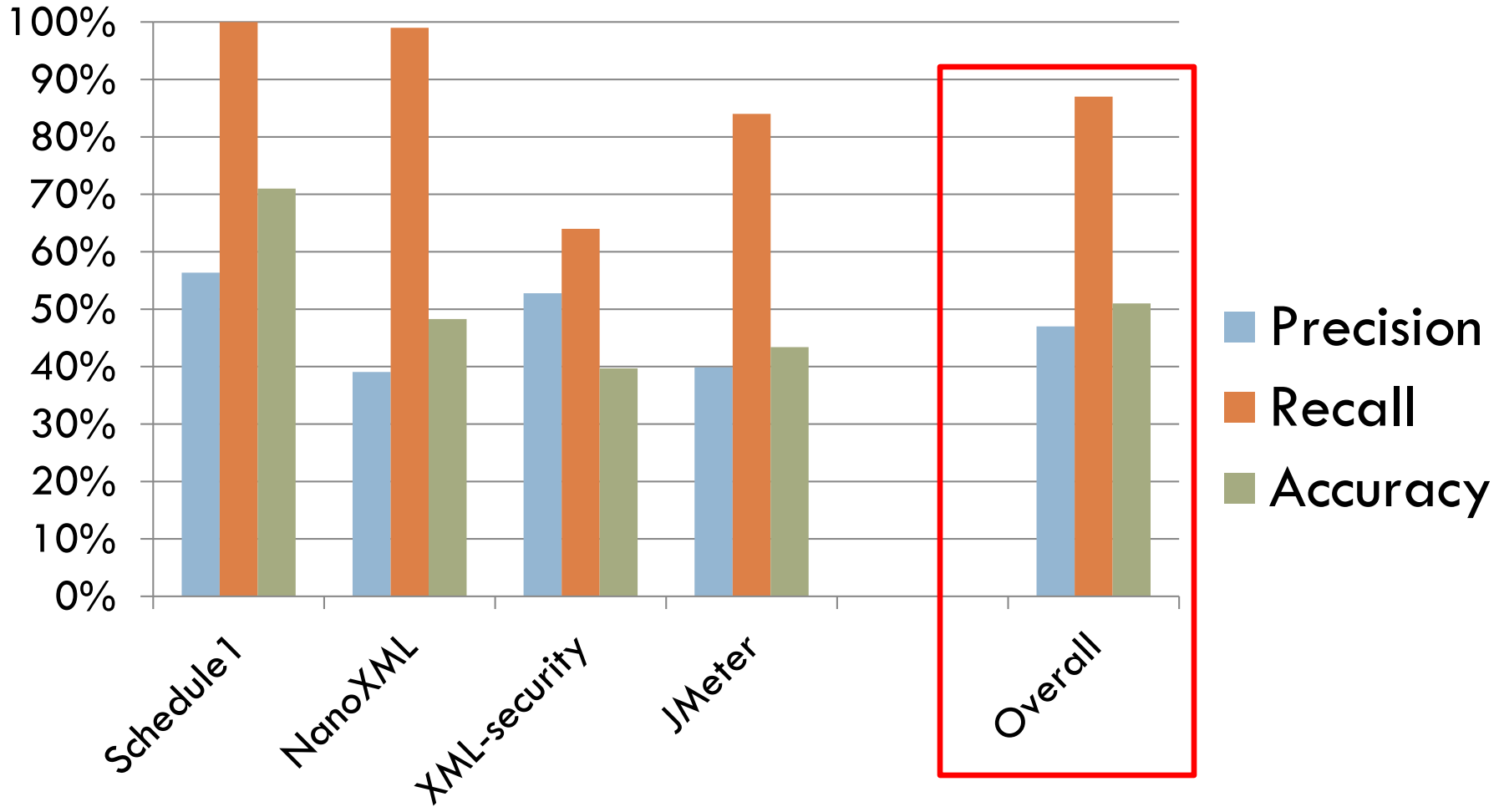
Recall


average recall 56%
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Schedule1
NanoXML
Ant
XML-security
Jmeter
ArgoUML
Overall
All
N
S

Recall


average recall 56%
average accuracy (F1) 39%
70%
60%
50%
40%
30%
20%
10%
0%
All
N
S
Schedule1
NanoXML
Ant
XML-security
Jmeter
ArgoUML
Overall

Results for SIR changes (bug fixes)



New Results for real changes

Future work

- Enhance the experimentation framework to support extended study for
  - More subjects
  - Other dynamic impact analyses
- Develop more precise technique for dynamic impact prediction

Outline

- Background: PI/EAS
- Methodology
  - Sensitivity Analysis
  - Execution Differencing
- Results
- Contributions

Contributions

- A methodology for estimating the accuracy of dynamic impact analyses
- The first empirical study of the predictive accuracy of dynamic impact analysis
- Insights to the effectiveness of predictive dynamic impact analysis
  - Current dynamic impact analysis can be surprisingly imprecise
    - Precision 52% for random changes, 47% for SIR changes
  - Moreover, existing dynamic impact analysis can be also quite ***unsafe***
    - Recall 56% for random changes, 87% for SIR changes