How Well do Test Case Prioritization Techniques Support Statistical Fault Localization

- An empirical study on the effectiveness issue of continuous integration

Bo Jiang, ***Zhenyu Zhang***, *T.H. Tse*, and *T.Y. Chen*

The University of Hong Kong
Hong Kong

Swinburne University of Technology
Melbourne, Australia

33rd Annual IEEE International Computer Software and Applications Conference
Jul 21th, 2009

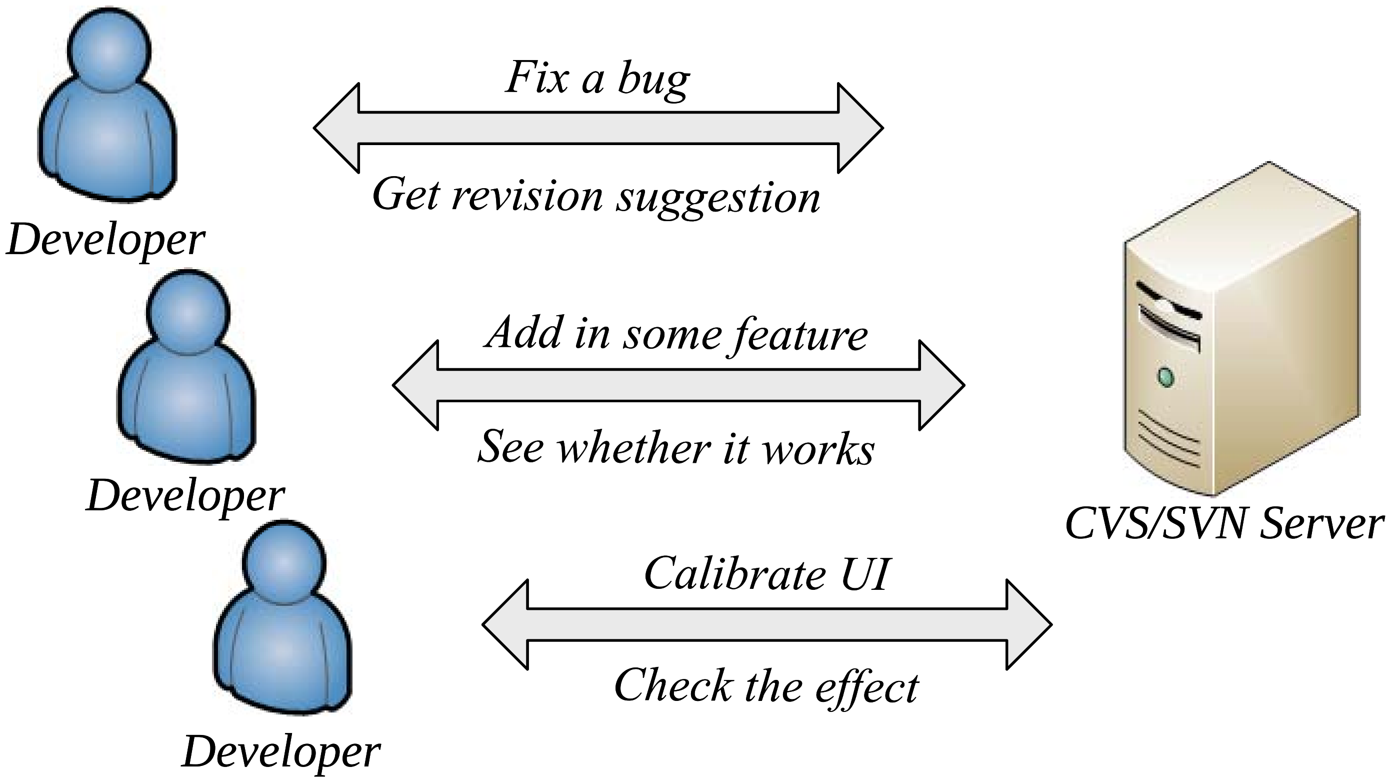
Contents

- Introduction and background
- Problems and research questions
- The empirical study
- Results and analysis
- Related work
- Conclusion

Contents

Motivating Example

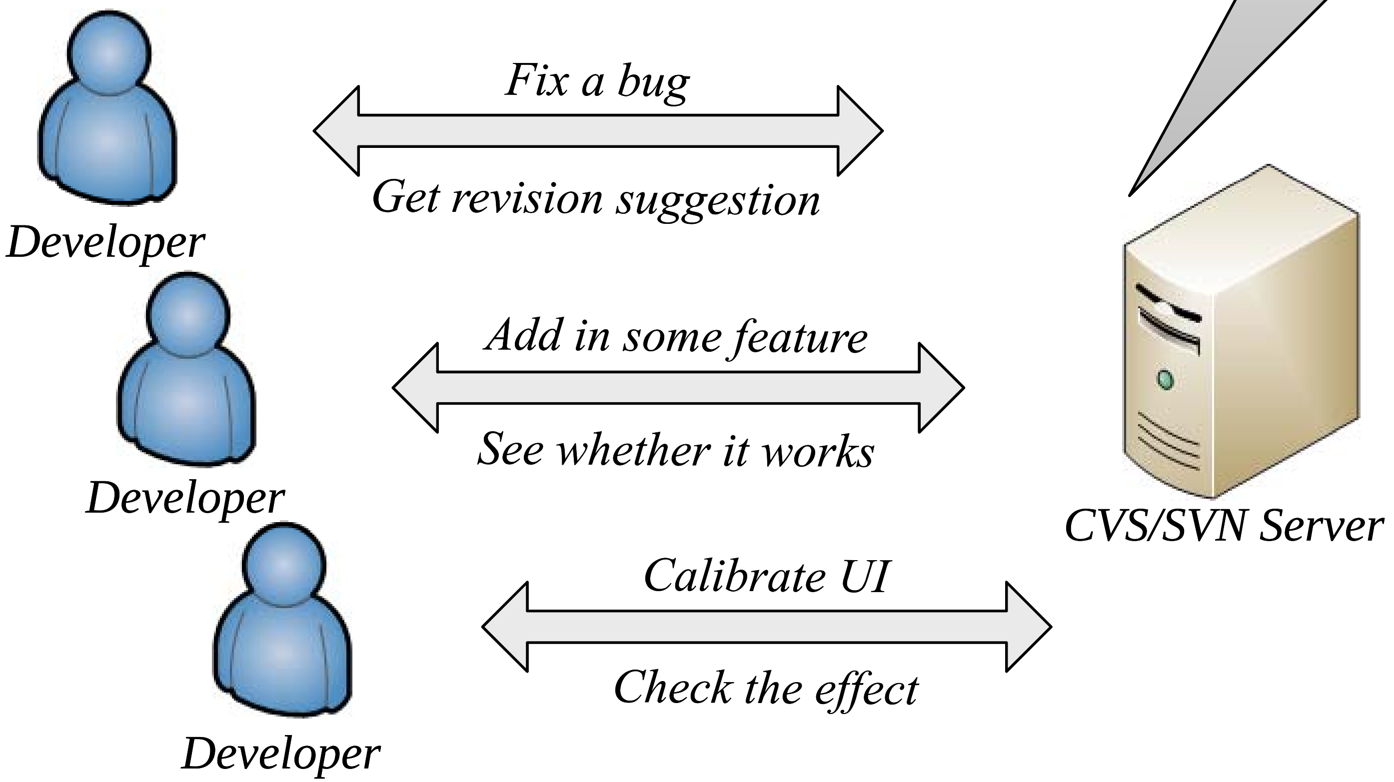
- A previous software development scenario
  - Source code version control



Motivating Example

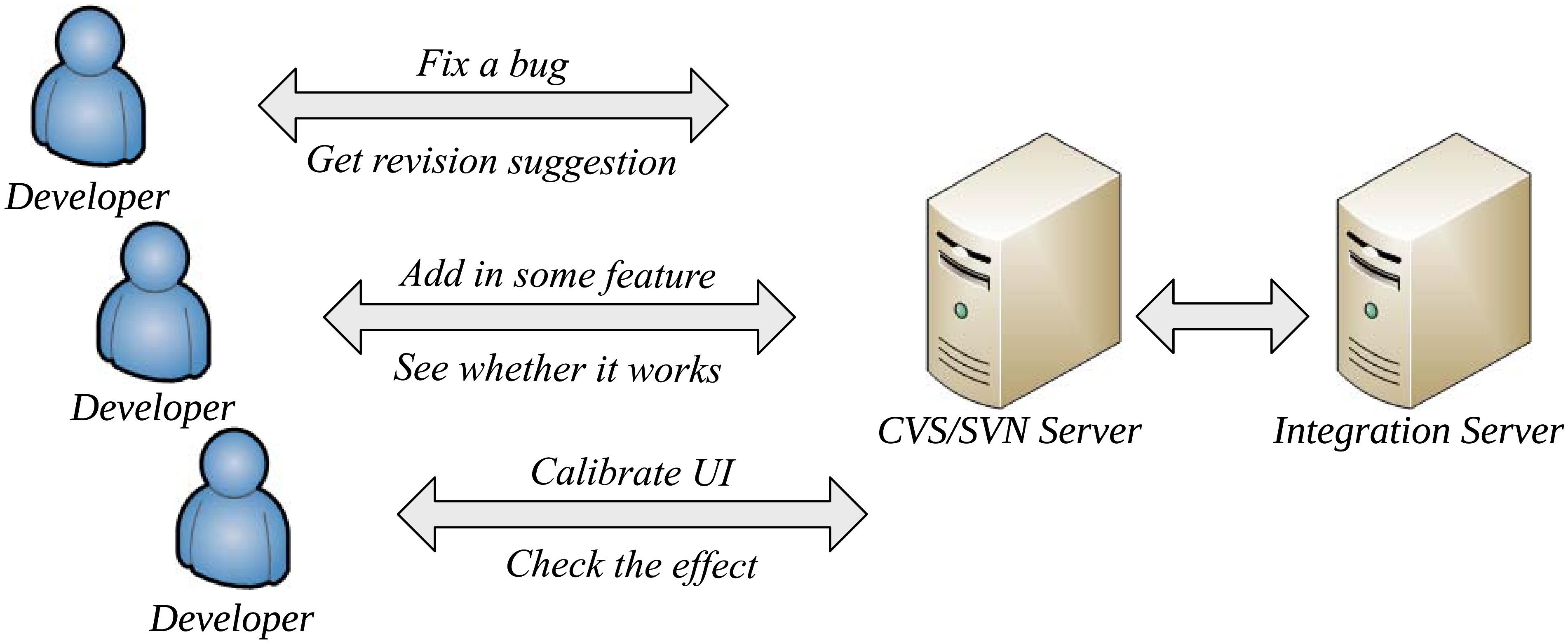
Fixed. Compiles.
But any draw back?
All crucial functionality?
Smoke test?

- A previous software development scenario
  - Source code version control



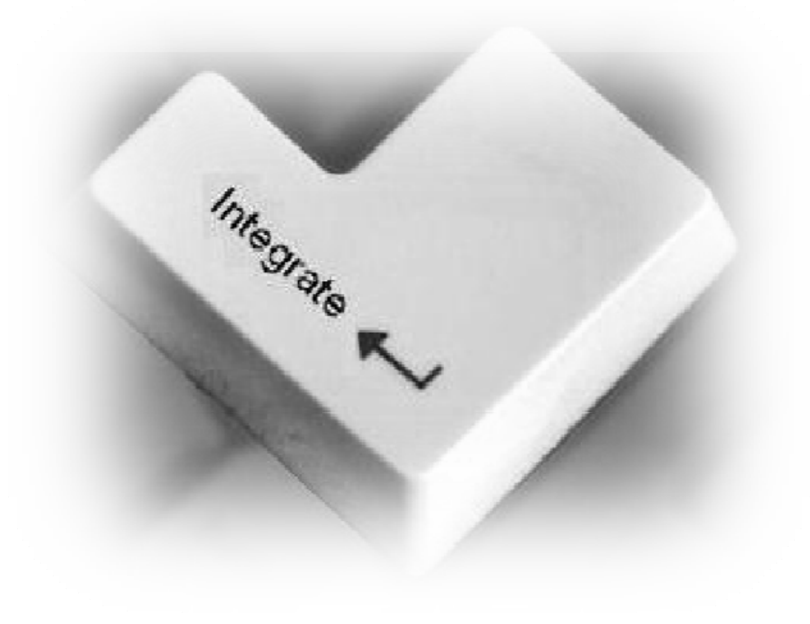
Motivating Example

- A modern software development scenario
  - More than a source code version control

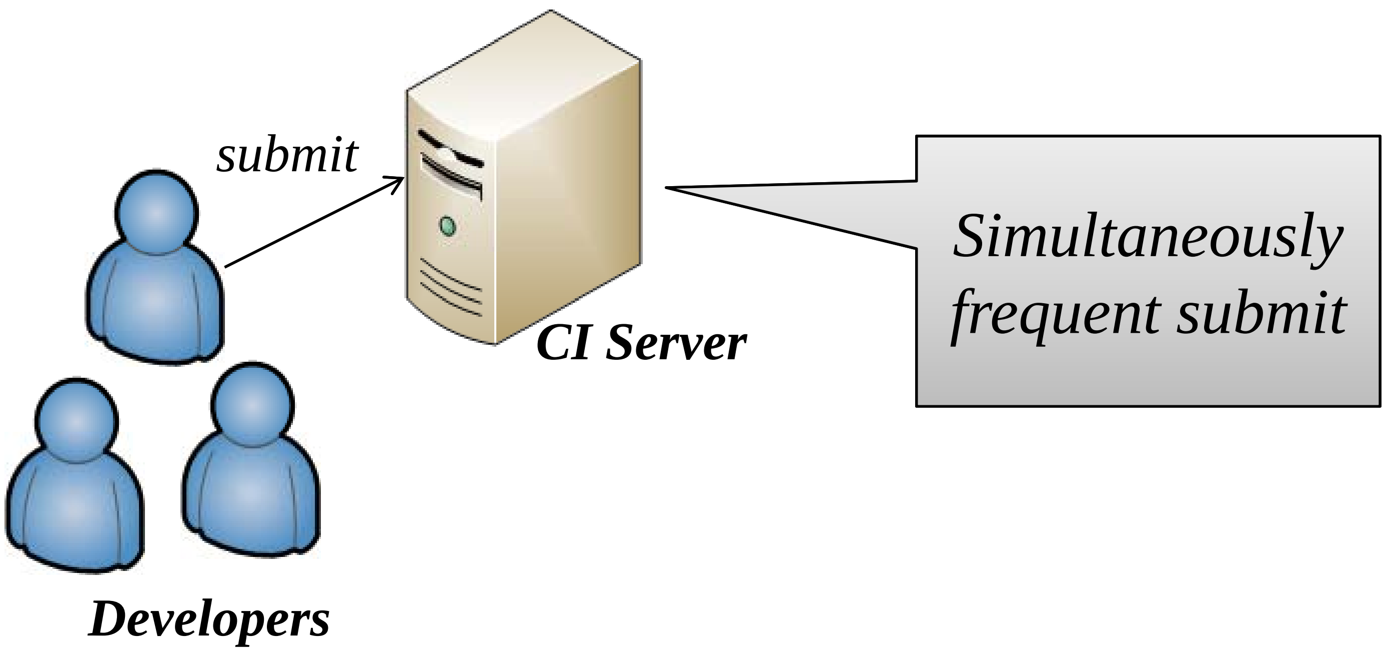


A Practice: Continuous Integration

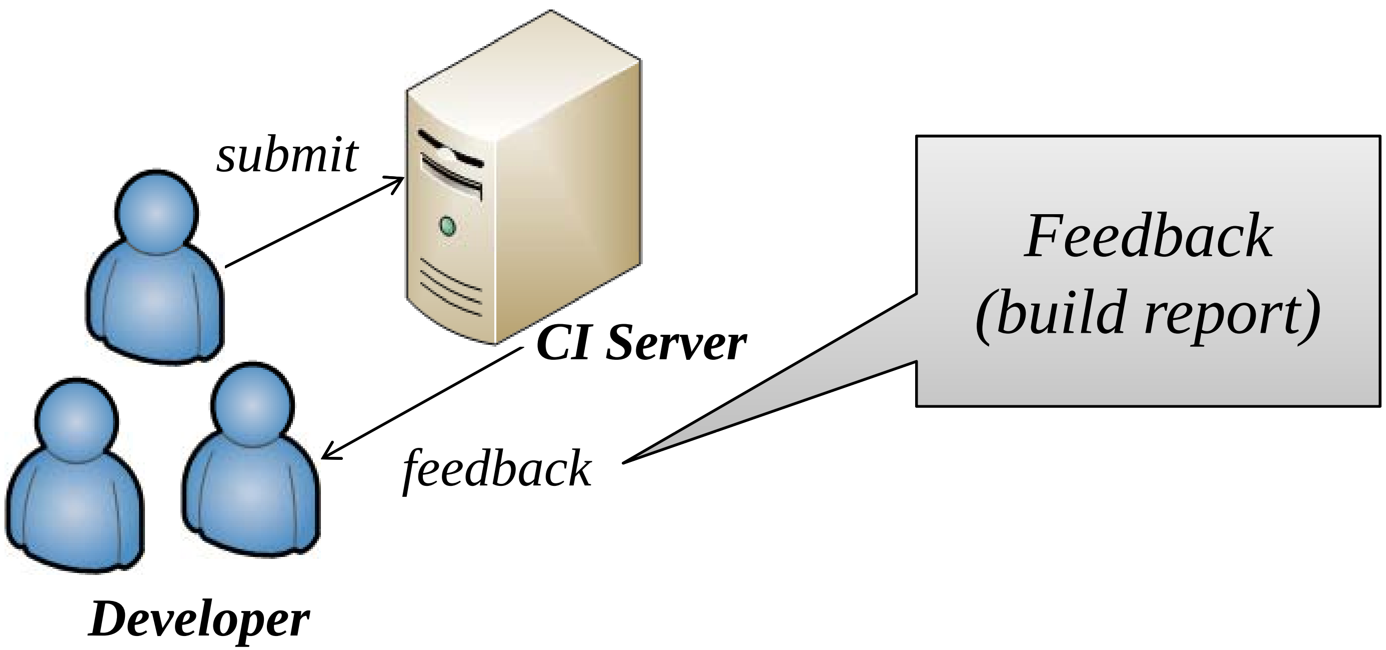
- ***Continuous Integration*** (CI) [1] is such a practice.



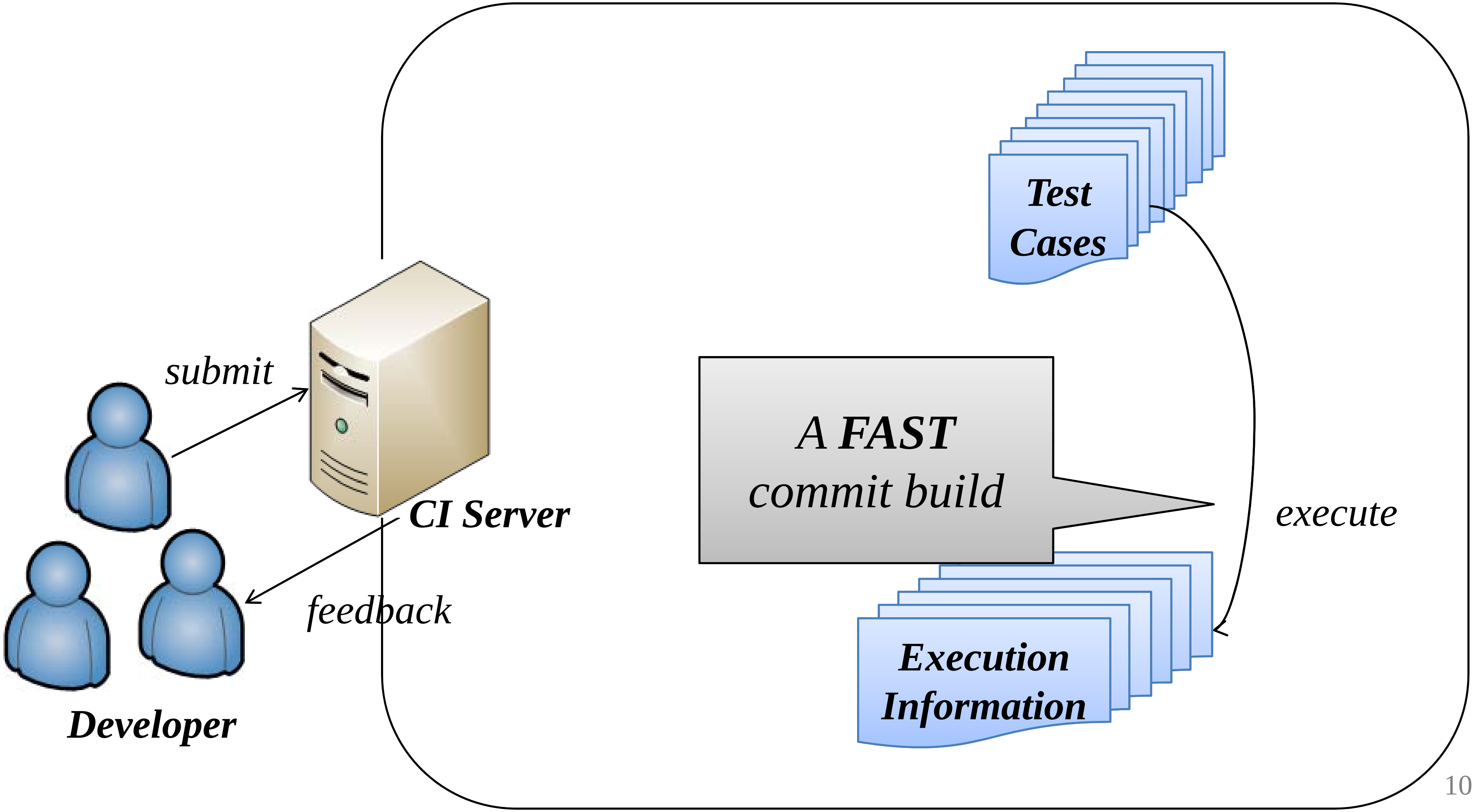
A CI framework



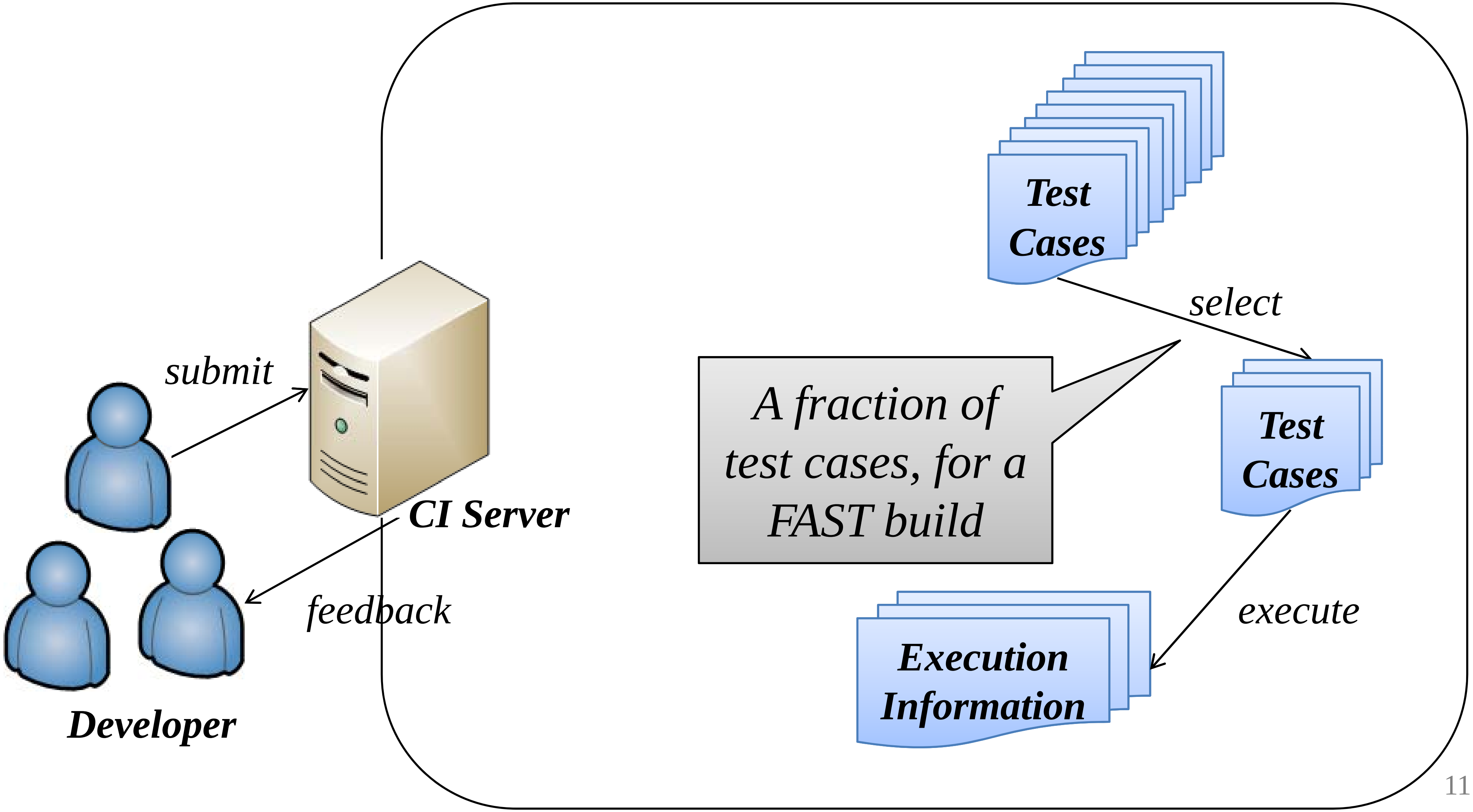
A CI framework



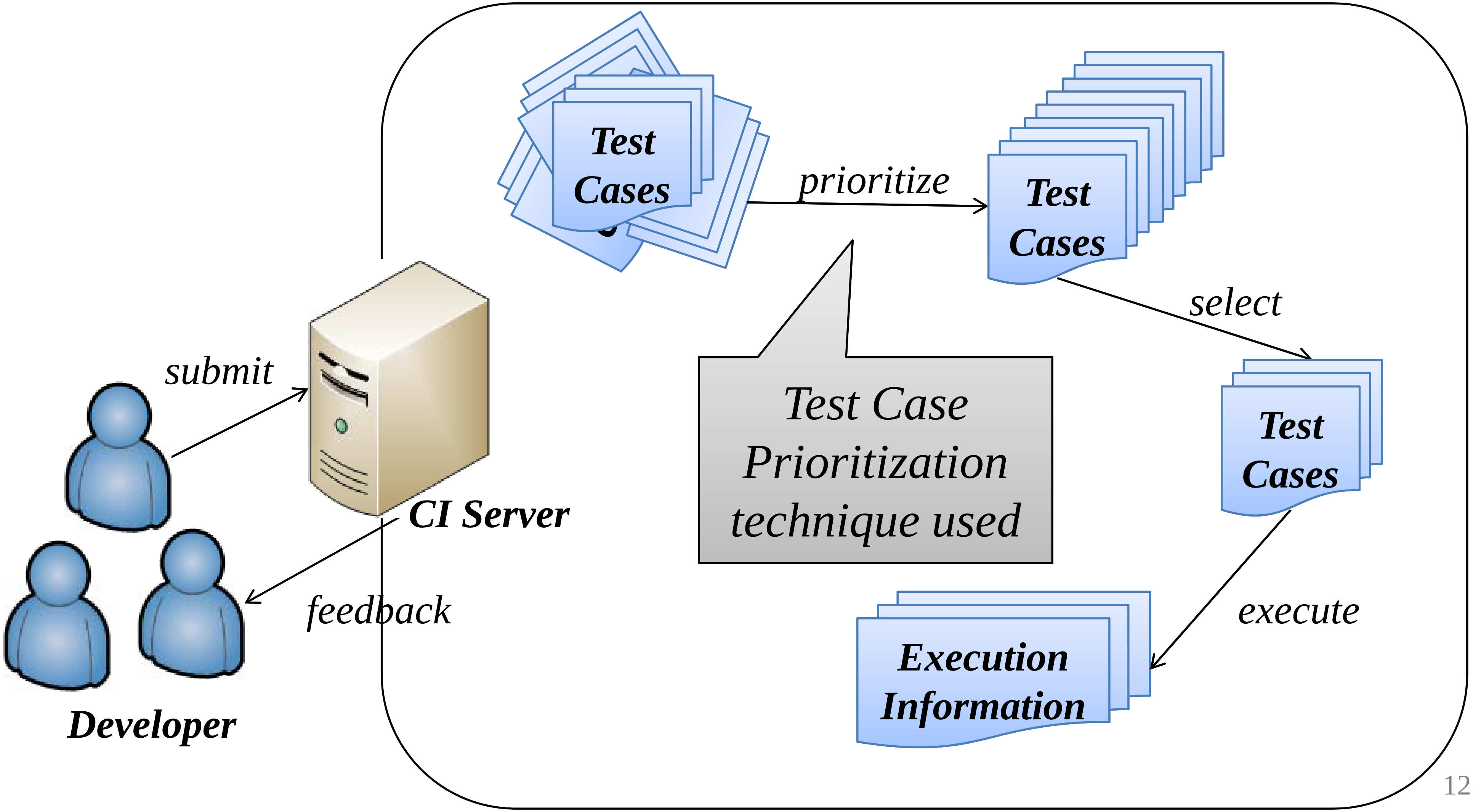
A CI framework



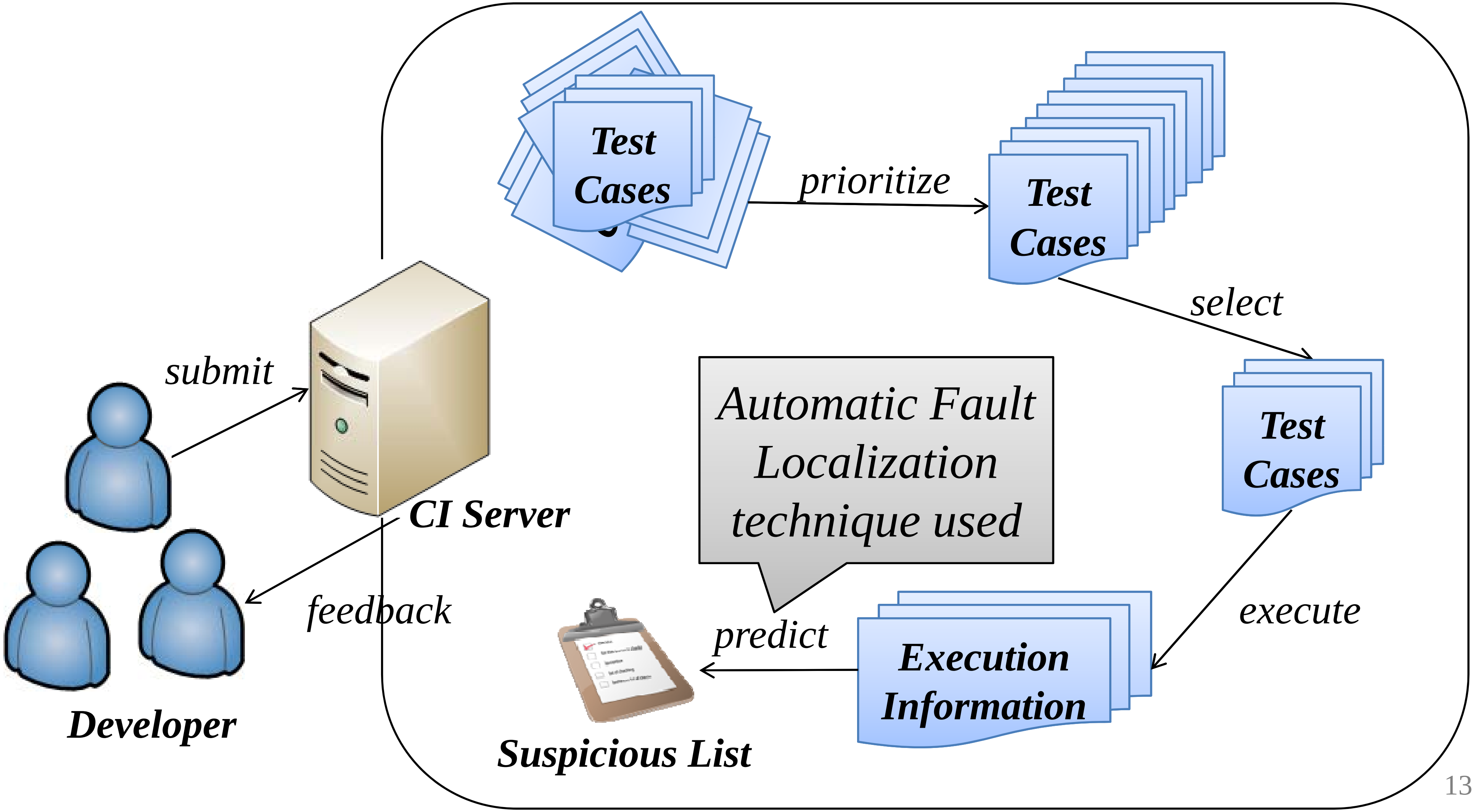
A CI framework



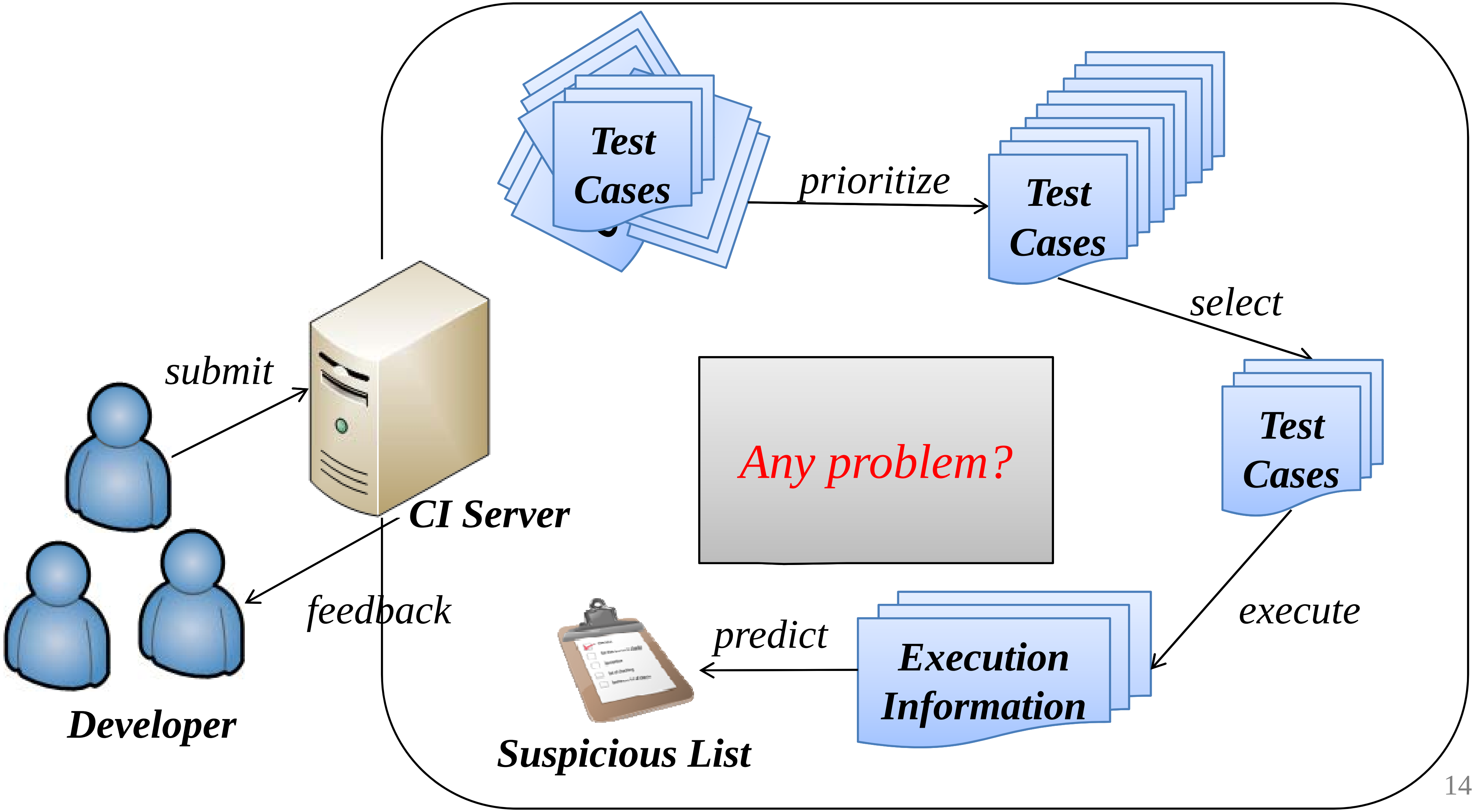
A CI framework



A CI framework



A CI framework



Contents

- Introduction and background
- **Problems and research questions**
- The empirical study
- Results and analysis
- Related work
- Conclusion

A Dilemma Existing in CI:

- Fast commit build needs
  - **Less** test cases (shorten the response time of CI cycle)
  - The less, the more efficient and fast

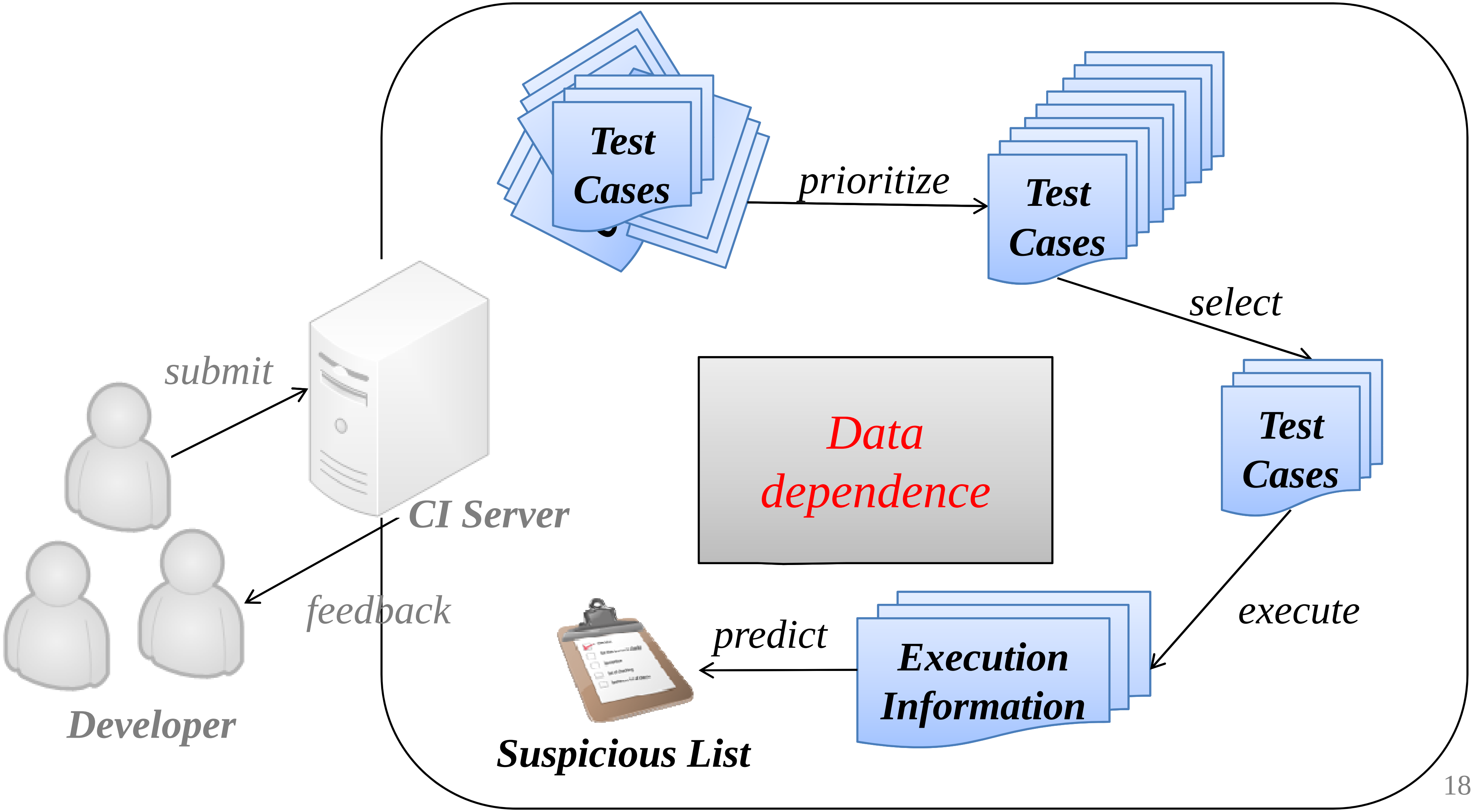
- Automatic Fault localization
  - **More** test cases (provide more information)
  - The more, the more effective and accurate

Research Questions:

- RQ1: To what extent will a fault localization technique be affected, if only a fraction of high-priority test cases are used as input?

- Any other question?

A CI framework



More Related Issues of CI:

- Different aims
  - Test Case Prioritization (TCP) techniques
    - E.g., to increase the rate of failure detection
  - Fault Localization (FL) techniques
    - E.g., to predict suspicious program location

- Since TCP affects FL, can TCP both detect failures earlier and effectively support FL?

More Research Questions:

- RQ1: To what extent will a fault localization technique be affected if only a fraction of high-priority test cases are used as input?
- RQ2: With a view to fasten the localization of fault, is there any particularly outstanding strategy for TCP?
- RQ3: Is random TCP an acceptable cost-effect TCP strategy?

Contents

Subject Programs

- The Siemens suite programs
  - The Software-artifact Infrastructure Repository (SIR)

	# of faulty versions	Executable LoC	Test Pool Size
tcas	41	133–137	1608
schedule	9	291–294	2650
schedule2	10	261–263	2710
tot_info	23	272–274	1052
print_tokens	7	341–342	4130
print_tokens2	10	350–354	4115
replace	32	508–515	5542

Experiment Setup

S0: For each faulty program
(121 faulty versions in total)

S1: Randomly select test cases from test pool to form a test suite
(Iterate different test suite sizes (50/100/200/300/400/500))

S2: Use a TCP technique to prioritize the test cases in the test suite
(Iterate 9 TCP techniques)

S3: Execute program over a fraction of high-priority test cases
(Iterate different fractions (10%/30%/50%/70%/90%/100%))

S4: Apply a FL technique, generate a suspicous list, and evaluate its effectiveness
(Iterate 4 FL techniques)

S5: Repeat **S1** to **S4** for 100 times to reduce the affection of noise

Selection of Techniques

- 4 FL techniques
  - Tarantula [2]

$$suspiciousness_T(s) = \frac{\%failed(s)}{\%passed(s) + \%failed(s)}$$

  - SBI [2]

$$suspiciousness_S(p) = \frac{failed(p)}{passed(p) + failed(p)}$$

  - Jaccard [3]

  - Ochiai [3]

$$suspiciousness_O(s) = \frac{failed(s)}{sqrt(totalfailed * (failed(s)+passed(s)))}$$

Selection of Techniques

- 9 TCP techniques
  - Granularity – at what level of unit

    Statement-level [4] or *Function-level* [4]
  - Information used – based on what information

    Coverage-based [5] or *Distribution-based* [5]
  - Strategy – what concrete strategy

    Total [5], *Additional* [5], *Count Metric* [6], or *Proportional Binary Metric*[6]

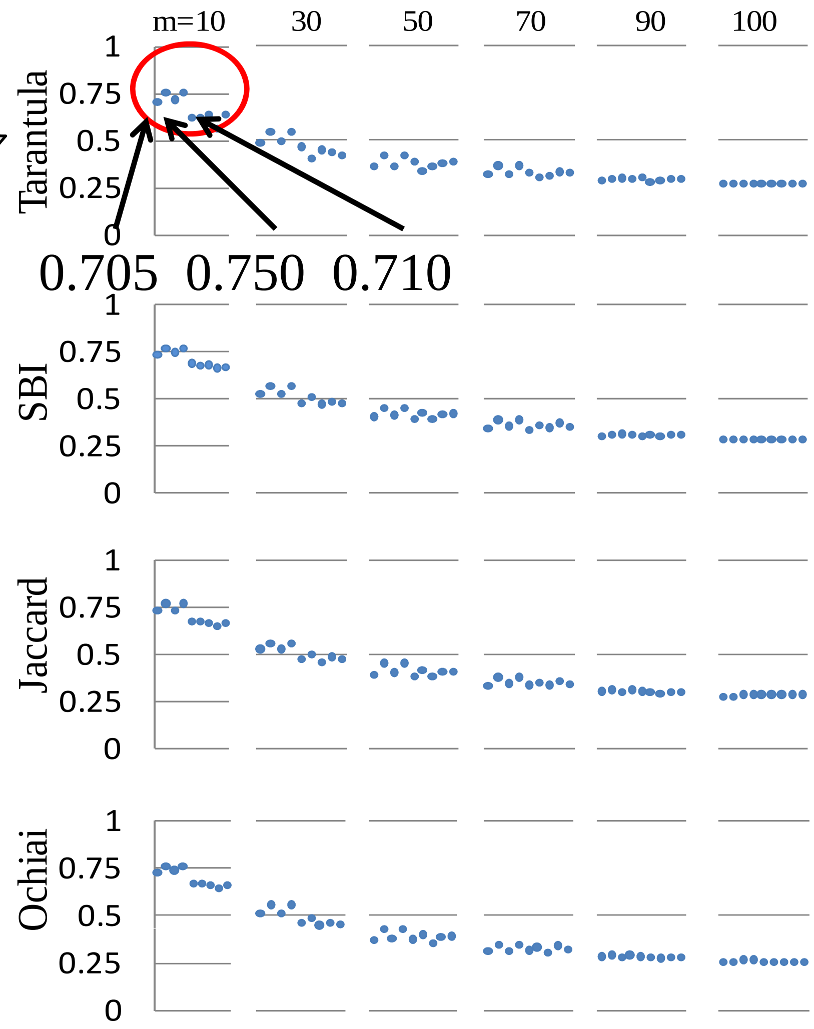
		Statement-level	Function-level
Coverage-based	"Total" strategy	TS	TF
	"Additional" strategy	AS	AF
Distribution-based	"Count Metric" distance	CS	CF
	"Proportional Binary metric" distance	PBS	PBF
	Random strategy	R	

Contents

- Introduction and background
- Problems and research questions
- The empirical study
- **Results and analysis**
- Related work
- Conclusion

Observations – Answering RQ1

The nine points stand for effectiveness w.r.t. CF, PBF, CS, PBS, AF, AS, R, TF, and TS.

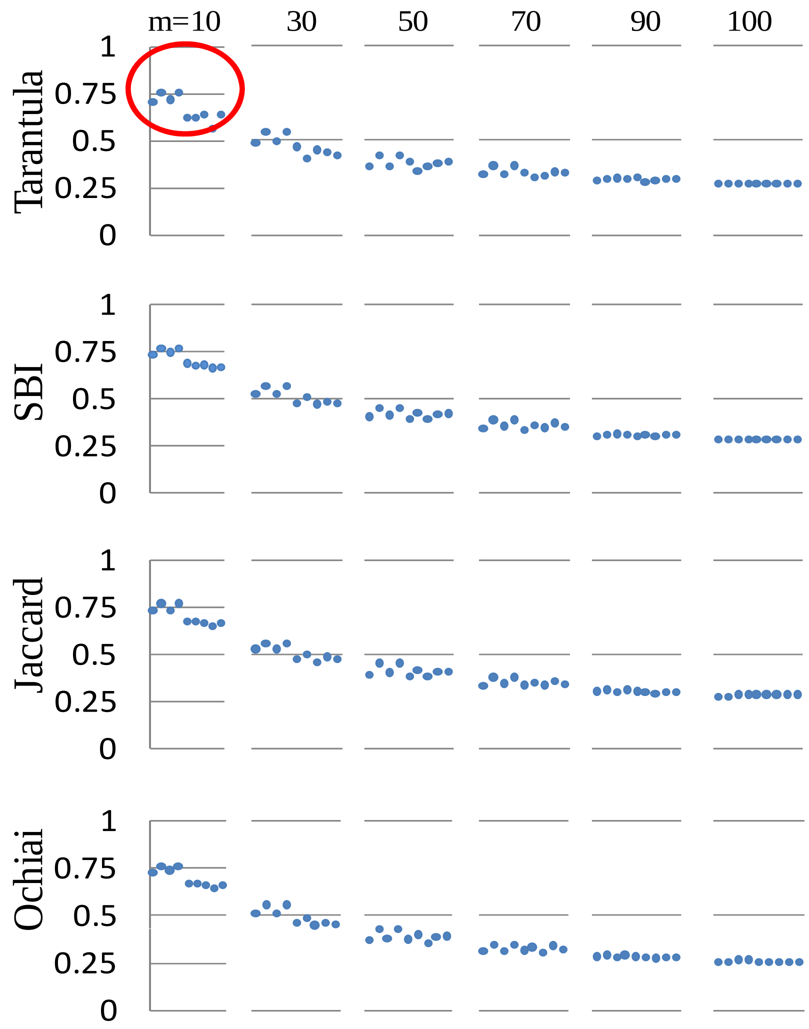


Observations – Answering RQ1

Expense

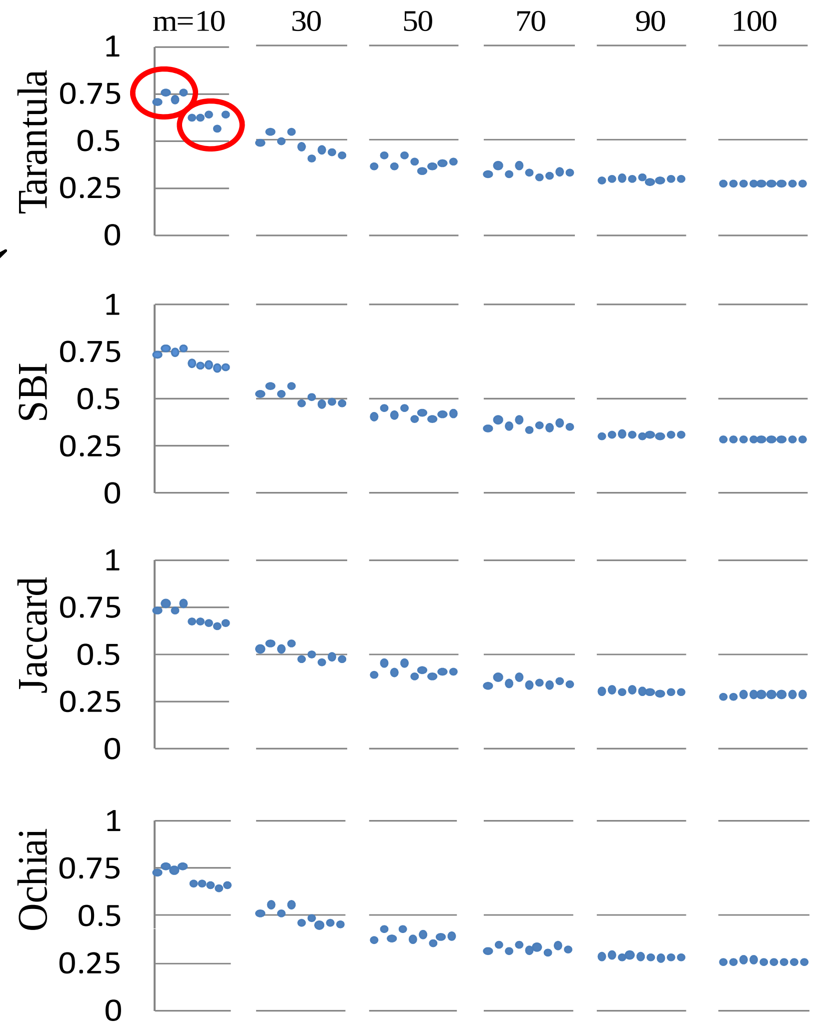
$$= \frac{\text{rank of faulty statements}}{\text{number of all statements}} \text{ [6]}$$

- *The lower the better*



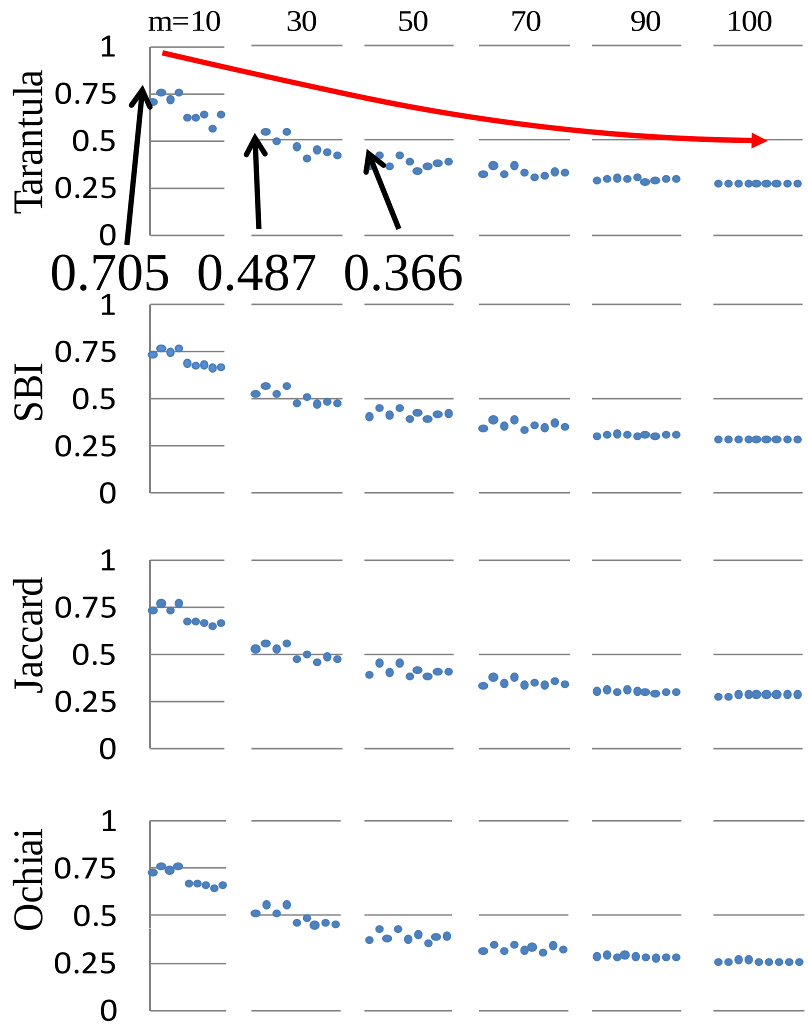
- *Example*: A program consists of statements $s_1 - s_{50}$, where s_{20} is faulty:
- *For list* <s_{13}, s_{31}, s_{25}, s_4, s_{20}, …>
  - Expense = 5/50 = 0.1

Observations – Answering RQ1



Different TCP techniques have different impact on FL techniques.

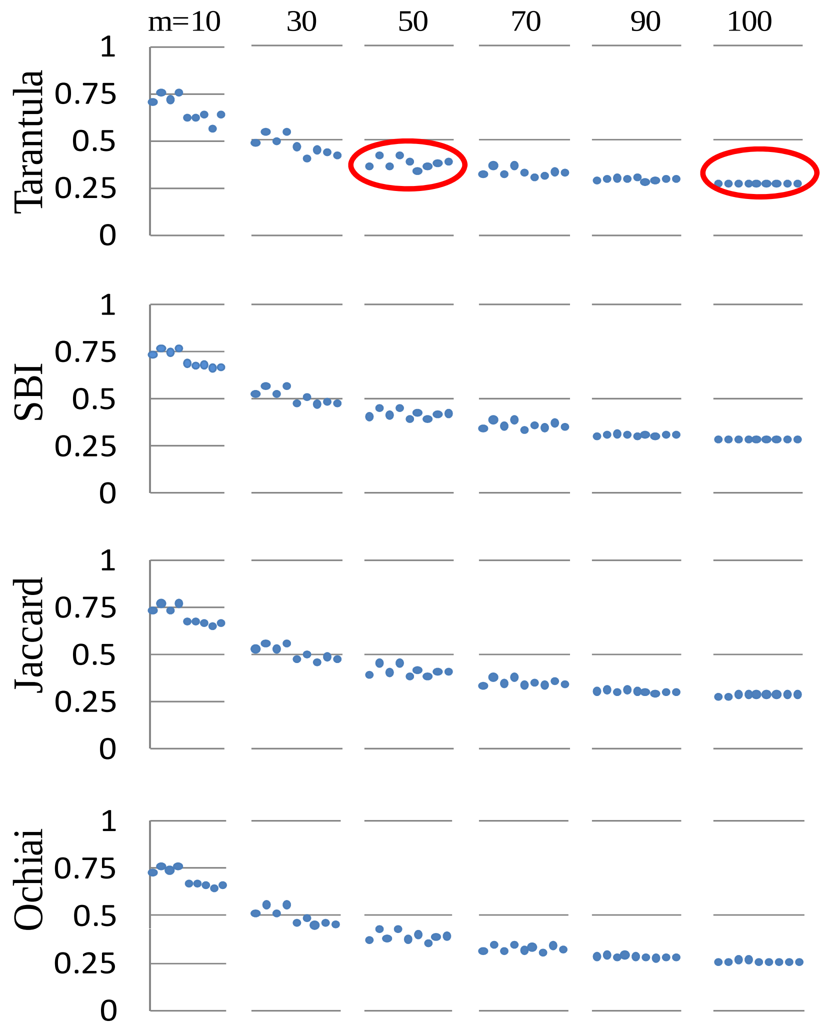
Observations – Answering RQ1



Large fraction, more effective.

Observations – Answering RQ1

No significant differences for effectiveness w.r.t. 50% fraction and 100% fraction.



Observations – Answering RQ2

	m	10	30	50	70	90	100
Distribution-based	CF	164%	86%	39%	20%	7%	0%
	PBF	177%	102%	59%	33%	9%	0%
	CS	164%	86%	41%	21%	7%	0%
	PBS	176%	102%	59%	33%	9%	0%
Random	R	140%	70%	39%	19%	7%	0%
Coverage-based	AF	140%	73%	43%	22%	7%	0%
	AS	**139%**	**64%**	**36%**	**17%**	**5%**	**0%**
	TF	128%	70%	45%	26%	8%	0%
	TS	139%	66%	45%	22%	8%	0%

Expense=26.8%

Expense=25%

(26.8-25)/25=7%

Relative Additional Expense

$$= \frac{Expense(m) - Expense(100)}{Expense(100)}$$

- *The lower the better*

Observations – Answering RQ2

	m	10	30	50	70	90	100
Distribution-based	CF	164%	86%	39%	20%	7%	0%
	PBF	177%	102%	59%	33%	9%	0%
	CS	164%	86%	41%	21%	7%	0%
	PBS	176%	102%	59%	33%	9%	0%
Random	R	140%	70%	39%	19%	7%	0%
Coverage-based	AF	140%	73%	43%	22%	7%	0%
	AS	**139%**	**64%**	**36%**	**17%**	**5%**	**0%**
	TF	128%	70%	45%	26%	8%	0%
	TS	139%	66%	45%	22%	8%	0%

PBF and PBS are most affected by fraction.
(low robustness)
(more sensitive)

Observations – Answering RQ2

	m	10	30	50	70	90	100
Distribution-based	CF	164%	86%	39%	20%	7%	0%
	PBF	177%	102%	59%	33%	9%	0%
	CS	164%	86%	41%	21%	7%	0%
	PBS	176%	102%	59%	33%	9%	0%
Random	R	140%	70%	39%	19%	7%	0%
Coverage-based	AF	140%	73%	43%	22%	7%	0%
	AS	**139%**	**64%**	**36%**	**17%**	**5%**	**0%**
	TF	128%	70%	45%	26%	8%	0%
	TS	139%	66%	45%	22%	8%	0%

AS is least affected by fraction.
(high robustness)
(less sensitive)

Observations – Answering RQ3

	m	10	30	50	70	90	100
Distribution-based	CF	1.168	1.222	0.986	1.025	0.913	1.000
	PBF	1.258	1.450	1.502	1.715	1.252	1.000
	CS	1.172	1.232	1.029	1.073	0.947	1.000
	PBS	1.255	1.449	1.501	1.713	1.250	1.000
Random	R	1.000	1.000	1.000	1.000	1.000	1.000
Coverage-based	AF	0.997	1.043	1.084	1.126	0.887	1.000
	AS	0.994	0.911	0.908	0.907	0.624	1.000
	TF	0.913	0.996	1.129	1.326	1.048	1.000
	TS	0.990	0.947	1.133	1.149	1.077	1.000

Expense=33.8%

Expense=27%

33.8/25=1.250

Relative Expense

$$= \frac{Expense}{Expense\ of\ Random}$$

- *The lower the better*

Observations – Answering RQ3

	m	10	30	50	70	90	100
Distribution-based	CF	1.168	1.222	0.986	1.025	0.913	1.000
	PBF	1.258	1.450	1.502	1.715	1.252	1.000
	CS	1.172	1.232	1.029	1.073	0.947	1.000
	PBS	1.255	1.449	1.501	1.713	1.250	1.000
Random	R	1.000	1.000	1.000	1.000	1.000	1.000
Coverage-based	AF	0.997	1.043	1.084	1.126	0.887	1.000
	AS	0.994	0.911	0.908	0.907	0.624	1.000
	TF	0.913	0.996	1.129	1.326	1.048	1.000
	TS	0.990	0.947	1.133	1.149	1.077	1.000

Coverage-based techniques perform better than Distribution-based techniques.

Observations – Answering RQ3

	m	10	30	50	70	90	100
Distribution-based	CF	1.168	1.222	0.986	1.025	0.913	1.000
	PBF	1.258	1.450	1.502	1.715	1.252	1.000
	CS	1.172	1.232	1.029	1.073	0.947	1.000
	PBS	1.255	1.449	1.501	1.713	1.250	1.000
Random	R	1.000	1.000	1.000	1.000	1.000	1.000
Coverage-based	AF	0.997	1.043	1.084	1.126	0.887	1.000
	AS	0.994	0.911	0.908	0.907	0.624	1.000
	TF	0.913	0.996	1.129	1.326	1.048	1.000
	TS	0.990	0.947	1.133	1.149	1.077	1.000

R is a cost-effective choice.

Contents

- Introduction and background
- Problems and research questions
- The empirical study
- Results and analysis
- **Related work**
- Conclusion

Related Work

[1] P. M. Duvall, S. Matyas, and A. Glover. Continuous Integration: Improving Software Quality and Reducing Risk. Addison Wesley, 2007.

- A general introduction of CI

[2] Y. Yu, J. A. Jones, and M. J. Harrold. An Empirical Study of the Effects of Test-suite Reduction on Fault Localization. ICSE 2008.

- FL techniques

[3] R. Abreu, P. Zoeteweij, and A. J. C. van Gemund. On the Accuracy of Spectrum-based Fault Localization. TAICPART-MUTATION 2007.

- FL techniques

Related Work

[4] S. G. Elbaum, A. G. Malishevsky, and G. Rothermel. Test case prioritization: a family of empirical studies. TSE, 2002.

- TCP techniques

[5] D. Leon and A. Podgurski. A Comparison of Coverage-based and Distribution-based Techniques for Filtering and Prioritizing Test Cases. ISSRE 2003.

- TCP techniques

[6] D. Jeffrey, N. Gupta, and R. Gupta. Fault Localization Using Value Replacement. ISSTA 2008.

- Effectiveness Metric

Contents

- Introduction and background
- Problems and research questions
- The empirical study
- Results and analysis
- Related work
- **Conclusion**

Conclusion

- The effectiveness of a statistical FL technique is, on average, not much different between applying the entire test suite and applying the first half of the test suite as the FL's inputs.

- Statistical FL + Coverage-based TCP is less sensitive (more robust) than Statistical FL + Distribution-based TCP.
  - Least sensitive TCP: *Additional-Statement* (*AS*)
  - No other TCP is less sensitive than the *Random* (*R*) in all sizes.

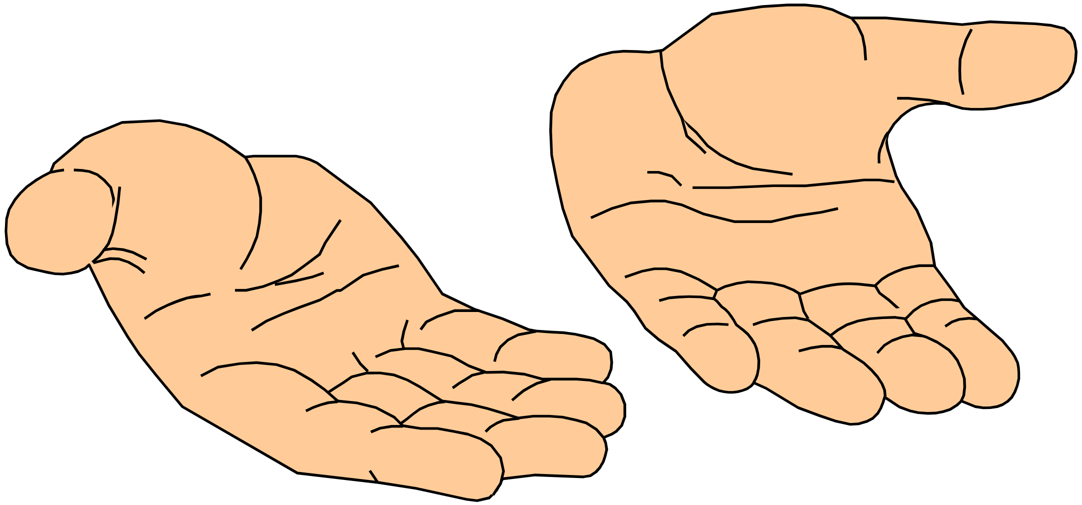
- The *R* prioritization can be a cost-effective choice, in supporting fault localization in a test budget limited environment.

Future Work

- For what kind of program spectra in executions, will *AS* be a best TCP strategy, in supporting FL?   Proof?
  - The process of a AS prioritization
    - The highest-priority test case is the one with maximum coverage
    - The 2nd test case is the one taking in maximum additional coverage
    - When all statements have been covered, clear all flag and repeat
  - One Clue: *AS* maximizes the minimum number of times a statement is covered
  - …

  - Such a future work scientifically supports using AS in CI.

Your Comments are Welcome



Thank you!