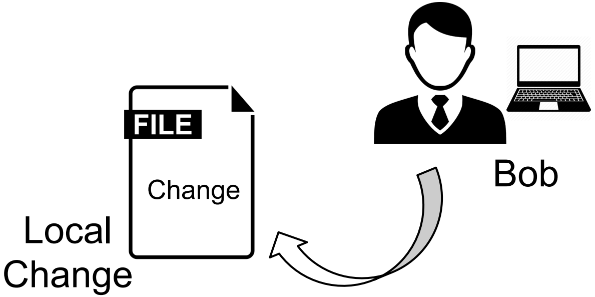
Towards Refactoring-Aware Regression Test Selection

Kaiyuan Wang, Chenguang Zhu, Ahmet Celik, Jongwook Kim, Don Batory, Milos Gligoric

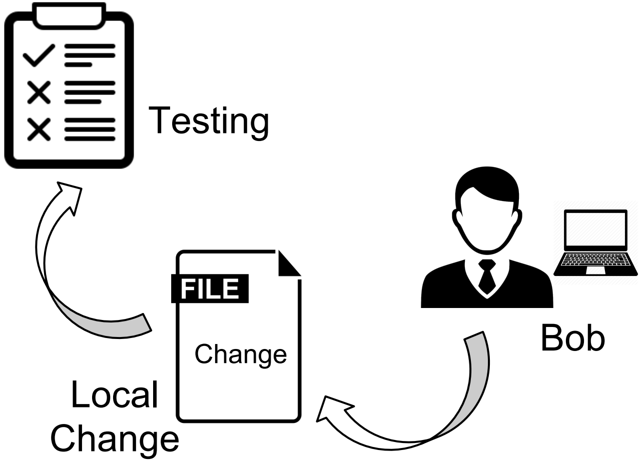
TEXAS
The University of Texas at Austin

Funded in part by the US National Science Foundation and a Google Faculty Research Award

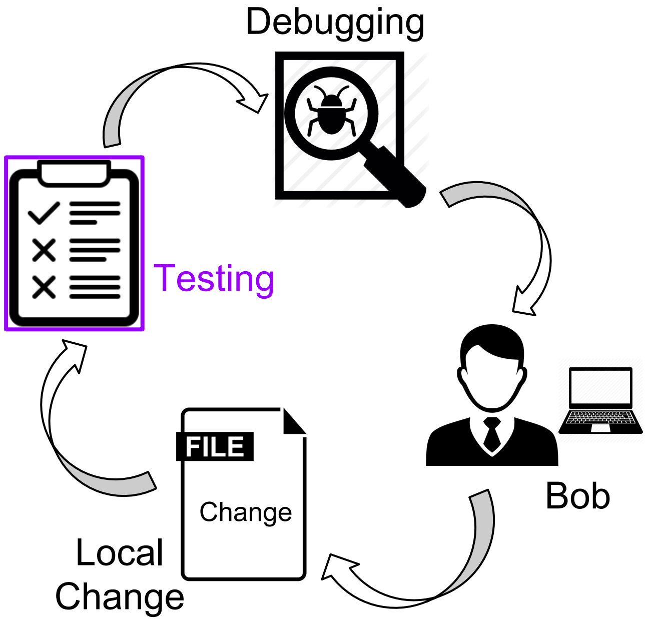
Regression Testing Is Important



Regression Testing Is Important



Regression Testing Is Important



Regression Testing is Costly

- Many companies report high cost
- Google Test Automation Platform (TAP) handles 800k builds and runs 150 million tests per day
- Microsoft’s CloudBuild is used by >4k developers and handles 20k builds per day

Regression Test Selection (RTS)

- Optimizes regression testing by skipping tests that are unaffected by recent code changes
- Maintains mapping from tests to code elements (statements, methods, classes)
- When code changes, uses mapping to find tests to run
- Many techniques developed over last 30 years, e.g.
    - TestTube (mapping from tests to functions)
    - FaultTracer (mapping from tests to methods)
    - Ekstazi (mapping from tests to classes)
    - HyRTS

Ekstazi Illustrated

- https://github.com/raphw/byte-buddy (Pull-up)

35da279

```
abstract class AbstractBase {
} ...
class ForLoadedExecutable extends AbstractBase {
 @Override
 protected ParameterList
  wrap(List<ParameterDescription> values) {
  return new Explicit(values);
 }
} ...
```

f1dfb66

```
abstract class AbstractBase {
 @Override
 protected ParameterList
  wrap(List<ParameterDescription> values) {
  return new Explicit(values);
 }
} ...
class ForLoadedExecutable extends AbstractBase {
} ...
```

Ekstazi Illustrated

- https://github.com/raphw/byte-buddy (Pull-up)

Ekstazi runs tests

35da279

```
abstract class AbstractBase {
} ...
class ForLoadedExecutable extends AbstractBase {
 @Override
 protected ParameterList
  wrap(List<ParameterDescription> values) {
  return new Explicit(values);
 }
} ...
```

```
JavaInstanceMethodTypeTest
 (AbstractBase, 7263), (ForLoadedExecutable, 4267), ...
JavaInstanceMethodHandleTest
 (AbstractBase, 7263), (ForLoadedExecutable, 4267), ...
```

f1dfb66

```
abstract class AbstractBase {
 @Override
 protected ParameterList
  wrap(List<ParameterDescription> values) {
  return new Explicit(values);
 }
} ...
class ForLoadedExecutable extends AbstractBase {
} ...
```

```
JavaInstanceMethodTypeTest
 (AbstractBase, 1076), (ForLoadedExecutable, 1291), ...
JavaInstanceMethodHandleTest
 (AbstractBase, 1076), (ForLoadedExecutable, 1291), ...
```

How to Improve RTS for Refactorings

- Refactorings, i.e. behavior preserving changes, do not impact the test outcome
- Existing RTS techniques do not reason about semantics of changes and thus run all tests affected by refactorings, e.g., rename method
- Recent work has shown that refactorings do happen in practice [Silva+FSE'16,Tsantalis+ICSE'17]
- How can we improve RTS for behavior preserving changes?

Reks: Refactoring-Aware RTS

- Integrate with refactoring engines and file tracking systems
- Keep track of the files affected by refactoring changes and files affected by non-refactoring changes from the last commit
- Defines rules to update dependencies for each test without running any test

Reks Illustrated

- https://github.com/raphw/byte-buddy (Pull-up)

Reks skips tests

35da279	f1dfb66

abstract class AbstractBase {
} ...
class ForLoadedExecutable extends AbstractBase {
@Override
protected ParameterList
wrap(List<ParameterDescription> values) {
return new Explicit(values);
}
} ...
	abstract class AbstractBase {
	@Override
	protected ParameterList
	wrap(List<ParameterDescription> values) {
	return new Explicit(values);
	}
	} ...
	class ForLoadedExecutable extends AbstractBase {
	} ...

JavaInstanceMethodTypeTest (AbstractBase, 7263), (ForLoadedExecutable, 4267), … JavaInstanceMethodHandleTest (AbstractBase, 7263), (ForLoadedExecutable, 4267), …	JavaInstanceMethodTypeTest (AbstractBase, 1076), (ForLoadedExecutable, 1291), … JavaInstanceMethodHandleTest (AbstractBase, 1076), (ForLoadedExecutable, 1291), …

Reks Update Rules

- Defined three update rules to update dependencies
- Supports all (27) refactorings available in Eclipse
- Rules
  - Modify class
  - Replace class
  - Move elements
- More formally defined in the paper

Reks Update Rule (1): Modify Class

- https://github.com/apache/commons-compress (Rename Field)

eee4f61

```
public class GzipCompressorOutputStream {
 private final byte[] buffer = new byte[512];
 private void deflate() throws IOException {
  int length = deflater.deflate(buffer, 0,
   buffer.length);
  if (length > 0) {
   out.write(buffer, 0, length);
  }}}
```

3e45dc8

```
public class GzipCompressorOutputStream {
 private final byte[] deflateBuffer = new byte[512];
 private void deflate() throws IOException {
  int length = deflater.deflate(deflateBuffer, 0,
   deflateBuffer.length);
  if (length > 0) {
   out.write(deflateBuffer, 0, length);
  }}}
```

Reks Update Rule (1): Modify Class

- https://github.com/apache/commons-compress (Rename Field)

Reks skips tests

eee4f61	3e45dc8

public class GzipCompressorOutputStream {
private final byte[] buffer = new byte[512];
private void deflate() throws IOException {
int length = deflater.deflate(buffer, 0,
buffer.length);
if (length > 0) {
out.write(buffer, 0, length);
}}}
	public class GzipCompressorOutputStream {
	private final byte[] deflateBuffer = new byte[512];
	private void deflate() throws IOException {
	int length = deflater.deflate(deflateBuffer, 0,
	deflateBuffer.length);
	if (length > 0) {
	out.write(deflateBuffer, 0, length);
	}}}

GZipTestCase (GzipCompressorOutputStream, 5482), …	GZipTestCase (GzipCompressorOutputStream, 9402), …

Reks Update Rule (2): Replace Class

- https://github.com/google/auto (Rename Class)

26eaf2f

```
class JavaTokenizer {…}
final class AbstractMethodExtractor {
 ImmutableListMultimap<String, String>
   abstractMethods(JavaTokenizer
     tokenizer, String packageName) {…}
}
…
```

75a9cee

```
class EclipseHackTokenizer {…}
final class AbstractMethodExtractor {
 ImmutableListMultimap<String, String>
   abstractMethods(EclipseHackTokenizer
     tokenizer, String packageName) {…}
}
…
```

Reks Update Rule (2): Replace Class

- https://github.com/google/auto (Rename Class)

Reks skips tests

26eaf2f	75a9cee

class JavaTokenizer {…}
final class AbstractMethodExtractor {
  ImmutableListMultimap<String, String>
    abstractMethods(JavaTokenizer
      tokenizer, String packageName) {…}
}
…
	class EclipseHackTokenizer {…}
	final class AbstractMethodExtractor {
	  ImmutableListMultimap<String, String>
	    abstractMethods(EclipseHackTokenizer
	      tokenizer, String packageName) {…}
	}
	…

AbstractMethodExtractorTest (JavaTokenizer, 2839), …	AbstractMethodExtractorTest (EclipseHackTokenizer, 8347), …

Reks Update Rule (3): Move Element

- https://github.com/apache/commons-crypto (Move Field)

eee4f61

```
public class CryptoCipherFactory {
 ...}
public class ConfigurationKeys {
 public static final String
   CIPHER_CLASSES_DEFAULT =
   OpensslCipher.class.getName();
 …}
```

3e45dc8

```
public class CryptoCipherFactory {
 public static final String
   CIPHER_CLASSES_DEFAULT =
   OpensslCipher.class.getName();
}
public class ConfigurationKeys {
 …}
```

Reks Update Rule (3): Move Element

- https://github.com/apache/commons-crypto (Move Field)

Reks skips tests

eee4f61	3e45dc8

public class CryptoCipherFactory {
…}
public class ConfigurationKeys {
public static final String
CIPHER_CLASSES_DEFAULT =
OpensslCipher.class.getName();
…}
	public class CryptoCipherFactory {
	public static final String
	CIPHER_CLASSES_DEFAULT =
	OpensslCipher.class.getName();
	}
	public class ConfigurationKeys {
	…}

CryptoCipherFactoryTest (CryptoCipherFactory, 4901), (ConfigurationKeys, 3782), …	CryptoCipherFactoryTest (CryptoCipherFactory, 2170), (ConfigurationKeys, 7492), …

Evaluation: Research Questions

- **RQ1:** How many tests would have been skipped by Reks had it been used by open-source developers?

Evaluation: Research Questions

- **RQ1:** How many tests would have been skipped by Reks had it been used by open-source developers?
- **RQ2:** How many tests does Reks skip on average if refactorings are systematically performed?

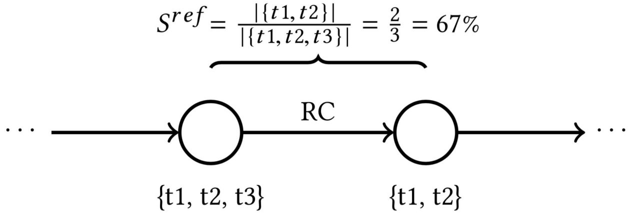
Evaluation: Research Questions

- **RQ1:** How many tests would have been skipped by Reks had it been used by open-source developers?
- **RQ2:** How many tests does Reks skip on average if refactorings are systematically performed?
- **RQ3:** How many tests does Reks skip on average for various refactoring types (e.g. move method) if refactorings are systematically performed?

Evaluation: Research Questions

- **RQ1:** How many tests would have been skipped by Reks had it been used by open-source developers?
- **RQ2:** How many tests does Reks skip on average if refactorings are systematically performed?
- **RQ3:** How many tests does Reks skip on average for various refactoring types (e.g. move method) if refactorings are systematically performed?
- **RQ4:** What is the cost of Reks update rules and how does this cost compare to the test execution time?

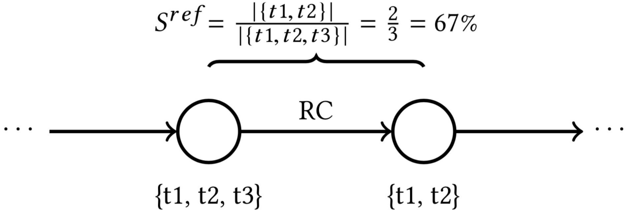
RQ1 How many tests would have been skipped by Reks had it been used by open-source developers?



$$S^{ref} = \frac{|\{t1, t2\}|}{|\{t1, t2, t3\}|} = \frac{2}{3} = 67\%$$

$$S^{ref} = |T^{ref}| / |T^{available}| \times 100$$

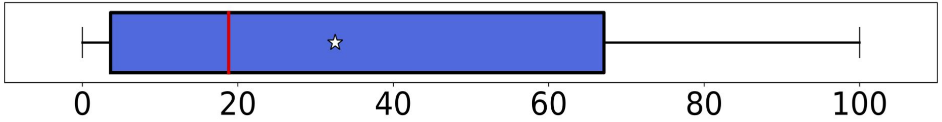
RQ1 How many tests would have been skipped by Reks had it been used by open-source developers?



$$S^{ref} = \frac{|\{t1, t2\}|}{|\{t1, t2, t3\}|} = \frac{2}{3} = 67\%$$

$$S^{ref} = |T^{ref}| \,/\, |T^{available}| \times 100$$

- 100 real refactorings from 37 projects
  - Reks saves 33% tests on average
  - 92 pure refactoring changes
    - Save [0%, 100%] tests, 34% on average
  - 8 mixed changes (refactoring + non-refactoring)
    - Save [0%, 64%] tests, 11% on average



RQ2 How many tests does Reks skip on average if refactorings are systematically performed?

Systematically perform 27 refactoring types on 10 open source projects using Eclipse refactoring engine. Total of 302k LOC & 74,160 refactorings.

Project	LOC	#Test Class	Coverage (%)		
			instruction	method	class
Coll	60251	159	83	77	95
Config	64341	163	87	83	98
DBCP	20547	30	45	56	95
IO	29159	100	86	82	100
JClassmate	6797	34	94	89	100
JObjectDiff	9976	61	89	84	94
Lang	69014	134	94	89	100
Net	26928	42	32	26	38
Pebble	13375	30	86	79	96
Stateless4J	1702	9	53	43	64
Avg	30.2k	76.1	74.9	70.8	88.5

RQ2 How many tests does Reks skip on average if refactorings are systematically performed?

% of Tests Reks skips per project

Project	Max	Med	Avg	Std
Coll	48.4	1.3	3.3	6.5
Config	72.4	4.3	13.8	18.0
DBCP	89.7	20.7	26.6	22.3
IO	63.0	9.0	10.9	10.1
JClassmate	85.3	32.4	37.1	25.5
JObjectDiff	85.3	8.2	21.4	23.5
Lang	52.2	1.5	4.6	7.1
Net	38.1	2.4	4.3	7.4
Pebble	96.7	96.7	67.3	40.4
Stateless4J	77.8	11.1	22.8	19.9
Max	96.7	96.7	67.3	40.4
Avg	70.9	-	21.2	-

RQ3 How many tests does Reks skip on average for various refactoring types if refactorings are systematically performed?

% of Tests Reks skips per refactoring type

Refactoring	Max	Avg	Std
Rename Field	96.7	13.0	20.4
Rename Method	96.7	22.3	27.1
Rename Local	96.7	0.7	2.9
Move Method	96.7	13.7	13.9
Change Signature	96.7	23.7	29.1
Extract Method	96.7	19.3	24.1
Extract Local	96.7	18.0	22.5
Extract Constant	96.7	15.0	21.2
Inline Constant	96.7	9.2	15.7
Inline Method	96.7	12.0	20.9
Inline Local	96.7	18.5	28.2
Convert Local to Field	96.7	18.1	28.0
Convert Anonymous	96.7	22.2	28.8
Move Type to New File	72.1	9.2	14.7
Extract Superclass	96.7	16.9	24.4
Extract Interface	96.7	9.6	18.6
Use Supertype	96.7	2.5	11.9
Push down	96.7	2.2	10.7
Pull up	96.7	21.1	23.6
Extract Class	96.7	19.0	26.3
Introduce Param Obj.	96.7	27.0	32.4
Introduce Indirection	96.7	23.6	27.7
Introduce Factory	96.7	15.0	25.9
Introduce Parameter	96.7	9.3	18.3
Encapsulate Field	96.7	12.0	21.6
Generalize Type	96.7	15.8	25.2
Infer Generic Type Args	12.6	0.1	0.6
Max	96.7	27.0	32.4

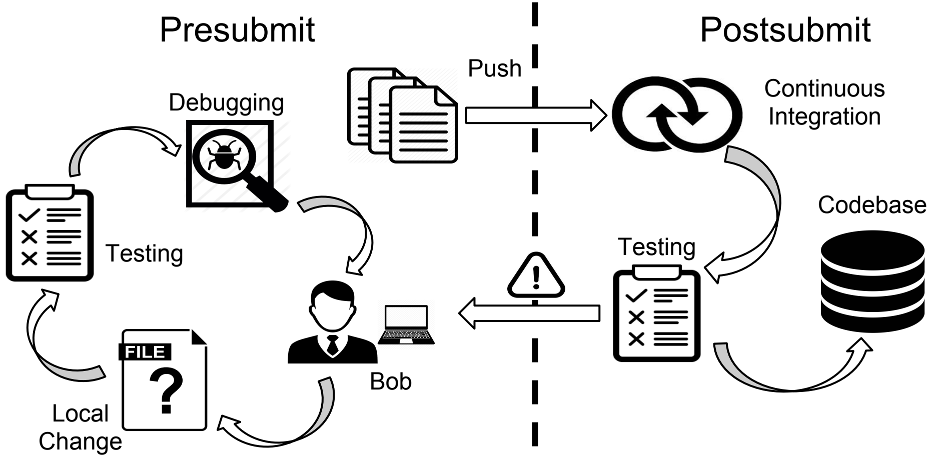
RQ4 What is the cost of Reks update rules and how does this cost compare to the test execution time?

Execution Time for Reks, RetestAll and Ekstazi

Project	Reks [s]	All [s]	Ekstazi [s]	R/A [%]
Coll	23.0	59.0	34.6	38.9
Config	14.3	54.1	33.8	26.5
DBCP	6.9	86.7	35.8	7.9
IO	4.3	132.7	11.9	3.3
JClassmate	2.9	3.5	4.2	82.8
JObjectDiff	19.9	35.6	35.7	56.0
Lang	14.9	43.5	17.9	34.2
Net	3.9	63.1	4.9	6.1
Pebble	1.2	6.2	5.9	19.4
Stateless4J	1.8	2.2	2.8	79.0
Max	23.0	132.7	35.8	82.8
Avg	9.3	48.7	18.7	35.4

Reks Assumptions

- Refactoring changes are behavior-preserving
- Any bugs introduced by the refactoring engine is captured by test runs in the post-submit phase
- Users perform refactorings using refactoring engine



Conclusion

- Reks is the first refactoring-aware RTS technique
- Reks saves 33% of tests for real refactoring relevant changes, 16% of tests for artificial refactorings
- Reks saves 64.6% of test time compared to RetestAll
- http://cozy.ece.utexas.edu/reks/

Kaiyuan Wang (kaiyuanw@utexas.edu)
Chenguang Zhu (cgzhu@utexas.edu)
Ahmet Celik (ahmetcelik@utexas.edu)
Jongwook Kim (jongwook@utexas.edu)
Don Batory (dsb@cs.utexas.edu)
Milos Gligoric (gligoric@utexas.edu)

TEXAS
The University of Texas at Austin

Q & A