

Do Automated Program Repair Techniques Repair Hard and Important Bugs?

Manish Motwani

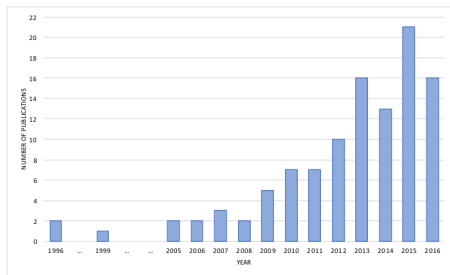
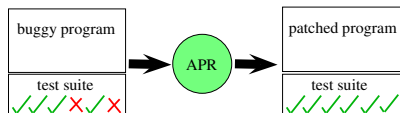
Sandhya Sankarnarayanan

René Just

Yuriy Brun

University of Massachusetts Amherst

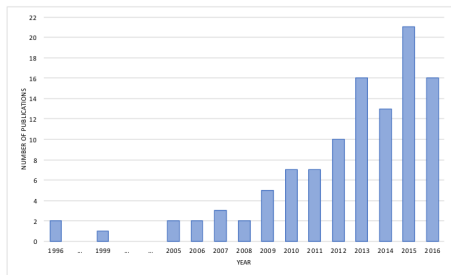
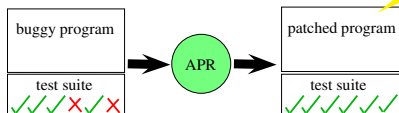
Automatic Program Repair: An Active Research Area



Automated program repair publications per year [1]

Automatic Program Repair: An Active Research Area

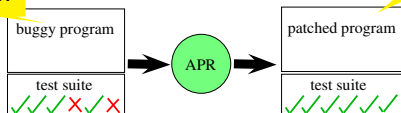
Is the patched program correct?



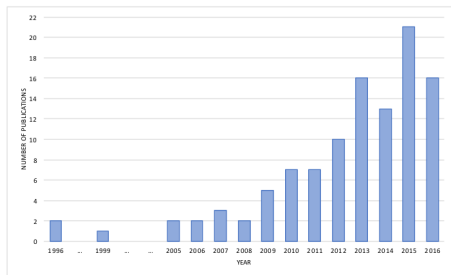
Automated program repair publications per year [1]

Automatic Program Repair: An Active Research Area

Is the bug hard to fix?



Is the patched program correct?

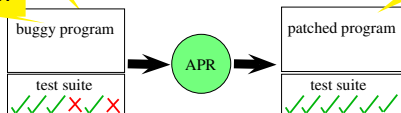


Automated program repair publications per year [1]

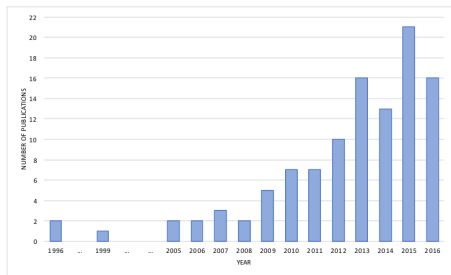
Automatic Program Repair: An Active Research Area

Is the bug important to fix?

Is the bug hard to fix?



Is the patched program correct?



Automated program repair publications per year [1]

Motivation

Prior evaluations of automated repair have focused on:

- ▶ Fraction of defects repaired [1,2]
- ▶ Computational resources required to repair defects [3,4]
- ▶ Correctness and quality of generated patches [5,6,7]
- ▶ Patch maintainability [8]
- ▶ Repair acceptability [9,10]

-
- [1] Ke et al. Repairing programs with semantic code search. ASE. 2015.
- [2] Qi et al. An analysis of patch plausibility and correctness for G&V patch generation systems. ISSTA. 2015.
- [3] Le Goues et al. The ManyBugs and IntroClass benchmarks for automated repair of C programs. TSE. 2015
- [4] Weimer et al. Leveraging program equivalence for adaptive program repair: models and first results. ASE. 2013
- [5] (DBGBench) Boehme, et al. Where is the bug and how is it fixed? an experiment with practitioners. FSE. 2017.
- [6] Smith et al. Is the cure worse than the disease? Overfitting in automated program repair. FSE. 2015.
- [7] Pei et al. Automated fixing of programs with contracts. TSE. 2014.
- [8] Fry et al. A human study of patch maintainability. ISSTA. 2012.
- [9] Durieux et al. Automatic repair of real bugs: An experience report on the Defects4J dataset. 2015.
- [10] Kim et al. Automatic patch generation learned from human-written patches. ICSE. 2013.

Motivation

YetAnotherFix
fixes 60% of the defects

Defect-1	patched
Defect-2	patched
Defect-3	not patched
Defect-4	patched
Defect-5	patched
Defect-6	not patched
Defect-7	patched
Defect-8	patched
Defect-9	not patched
Defect-10	not patched

ThisNeverEndsFix
fixes 30% of the defects

Defect-1	not patched
Defect-2	not patched
Defect-3	patched
Defect-4	not patched
Defect-5	not patched
Defect-6	not patched
Defect-7	not patched
Defect-8	not patched
Defect-9	patched
Defect-10	patched

Which automated program repair technique is better?

Motivation

YetAnotherFix
fixes 60% of the defects

Defect-1	patched
Defect-2	patched
Defect-3	not patched
Defect-4	patched
Defect-5	patched
Defect-6	not patched
Defect-7	patched
Defect-8	patched
Defect-9	not patched
Defect-10	not patched

Hard to fix
defects

ThisNeverEndsFix
fixes 30% of the defects

Defect-1	not patched
Defect-2	not patched
Defect-3	patched
Defect-4	not patched
Defect-5	not patched
Defect-6	not patched
Defect-7	not patched
Defect-8	not patched
Defect-9	patched
Defect-10	patched

Which automated program repair technique is better?

How about now?

Which is harder to fix?

Invalid error message



The screenshot shows a bug report page for Bug #53898. The title is "Bug #53898 PHAR reports invalid error message, when the directory does not exist". The report was created on 2011-02-01 09:55 UTC and modified on 2011-02-01 12:01 UTC. The reporter is nightstorm at tien dot pl, and the assignee is ilan (profika). The status is Closed, and the package is PHAR related. The PHP version is 5.3.5, the OS is Arch Linux x86_64, and the CVE-ID is None. The description states: "When we try to create a PHAR archive, and the directory, where we want to save it does not exist, PHAR complains about invalid file extension, instead of informing that there is a problem with accessing the requested location. This is a bit confusing."

Bug #53898 PHAR reports invalid error message, when the directory does not exist

Created: 2011-02-01 09:55 UTC Modified: 2011-02-01 12:01 UTC

From: nightstorm at tien dot pl Assigned: ilan (profika)

Status: Closed Package: PHAR related

PHP Version: 5.3.5 OS: Arch Linux x86_64

Private report: No CVE-ID: None

[View](#) [Add Comment](#) [Developer](#) [Edit](#)

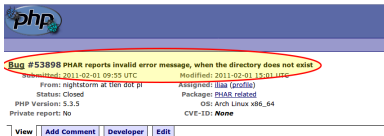
[2011-02-01 09:55 UTC] nightstorm at tien dot pl

Description:

When we try to create a PHAR archive, and the directory, where we want to save it does not exist, PHAR complains about invalid file extension, instead of informing that there is a problem with accessing the requested location. This is a bit confusing.

Which is harder to fix?

Invalid error message



Bug #53898 PHAR reports invalid error message, when the directory does not exist

Reported: 2011-02-01 09:55 UTC Modified: 2011-02-01 15:01 UTC
From: nightstorm at tlen dot pl Assigned: ilaa (profka)
Status: Closed Package: PHAR related
PHP Version: 5.3.5 OS: Arch Linux x86_64
Private report: No CVE-ID: None

[View](#) [Add Comment](#) [Developer](#) [Edit](#)

[2011-02-01 09:55 UTC] nightstorm at tlen dot pl

Description:

When we try to create a PHAR archive, and the directory, where we want to save it does not exist, PHAR complains about invalid file extension, instead of informing that there is a problem with accessing the requested location. This is a bit confusing.

Invalid memory access (Application crash)



Bug #5273 ISAPI crash on shutdown

Reported: 2000-06-29 18:06 UTC Modified: 2000-08-20 02:39 UTC
From: chris-php-bugs at digitaria dot com Assigned:
Status: Closed Package: Reproducible Crash
PHP Version: 4.0.0 Release OS: NT 4.0 SP6a
Private report: No CVE-ID: None

[View](#) [Add Comment](#) [Developer](#) [Edit](#)


[2000-06-29 18:06 UTC] chris-php-bugs at digitaria dot com

(The bug is actually in 4.0.1 release. bugs.php.net doesn't have that option)

With PHP configured using ISAPI, a PHP page which generates HTML, which references two images generated by PHP scripts

Which is harder to fix? Which is more important to fix?

Invalid error message



Bug #53898 PHAR reports invalid error message, when the directory does not exist
Modified: 2011-02-01 09:55 UTC
From: nightstorm at tlen dot pl
Assigned: ilaa (ilaa)
Status: Closed
Package: PHAR related
OS: Arch Linux x86_64
Private report: No
CVE-ID: None

[View](#) [Add Comment](#) [Developer](#) [Edit](#)

[2011-02-01 09:55 UTC] nightstorm at tlen dot pl
Description:

When we try to create a PHAR archive, and the directory, where we want to save it does not exist, PHAR complains about invalid file extension, instead of informing that there is a problem with accessing the requested location. This is a bit confusing.

Invalid memory access (Application crash)



Bug #5273 ISAPI crash on shutdown
Modified: 2000-06-29 18:06 UTC
From: chris-php-bugs at digitaria dot com
Assigned:
Status: Closed
Package: Reproducible Crash
OS: NT 4.0 SP6a
Private report: No
CVE-ID: None

[View](#) [Add Comment](#) [Developer](#) [Edit](#)

[2000-06-29 18:06 UTC] chris-php-bugs at digitaria dot com
(The bug is actually in 4.0.1 release. bugs.php.net doesn't have that option)
With PHP configured using ISAPI, a PHP page which generates HTML, which references two images generated by PHP scripts

Which is harder to fix? Which is more important to fix?

Invalid error message



Bug #53898 PHAR reports invalid error message, when the directory does not exist
Reported: 2011-02-01 09:55 UTC Modified: 2011-02-01 15:01 UTC
From: nightstorm at tien dot pt assigned: ilias (stefka)
Status: Closed Package: PHAR related
PHP Version: 5.3.5 OS: Arch Linux x86_64
Private report: No CVE-ID: None

[View](#) [Add Comment](#) [Developer](#) [Edit](#)

[2011-02-01 09:55 UTC] nightstorm at tien dot pt

Description:

When we try to create a PHAR archive, and the directory, where we want to save it does not exist, PHAR complains about invalid file extension, instead of informing that there is a problem with accessing the requested location. This is a bit confusing.

Easy and less important

Invalid memory access (Application crash)



Bug #5273 ISAPI crash on shutdown
Reported: 2000-06-29 18:06 UTC Modified: 2000-06-20 02:39 UTC
From: chris-php-bugs at digitaria dot com Assigned:
Status: Closed Package: Reproducible Crash
PHP Version: 4.0.0 Release OS: NT 4.0 SP6a
Private report: No CVE-ID: None

[View](#) [Add Comment](#) [Developer](#) [Edit](#)

[2000-06-29 18:06 UTC] chris-php-bugs at digitaria dot com

(The bug is actually in 4.0.1 release. bugs.php.net doesn't have that option)

With PHP configured using ISAPI, a PHP page which generates HTML, which references two images generated by PHP scripts

Hard and more important

How do we measure hardness and importance of a defect?

Goals of this study

A methodology for measuring a defect's hardness and importance.

An evaluation of whether automated program repair techniques repair hard and important defects.

Measuring hardness and importance of a defect

The screenshot shows a JIRA issue page for 'NumberUtils does not handle Long Hex numbers' in the Commons Lang project. The issue is a 'Major' bug with a 'Fixed' status, resolved in version 3.2. Annotations include:

- Priority:** A yellow box labeled 'Priority' with a red arrow pointing to the 'Major' priority indicator.
- Versions:** A yellow box labeled 'Versions' with red arrows pointing to the 'Affects Version/s: None' and 'Fix Version/s: 3.2' fields.
- Time to fix:** A yellow box labeled 'Time to fix' with a red arrow pointing to the 'Created' date (23/Aug/11 12:50).

Details:
Type: Bug
Priority: Major
Affects Version/s: None
Component/s: lang.math.*
Labels: None

Status: Fixed
Resolution: Fixed
Fix Version/s: 3.2

People:
Assignee: Unassigned
Reporter: Setb
Votes: 0
Watchers: 0

Dates:
Created: 23/Aug/11 12:50
Updated: 15/Oct/13 02:33
Resolved: 26/Oct/13 01:06

Description:
NumberUtils.createLong() does not handle hex numbers, but createInteger() handles hex numbers. This seems odd.
NumberUtils.createNumber() assumes that hex numbers can only be Integer.
Again, why not handle bigger Hex numbers?

bug report

Measuring hardness and importance of a defect

The screenshot shows a Jira issue page for 'NumberUtils does not handle Long Hex numbers'. The issue is categorized as a 'Bug' with a 'Major' priority. The status is 'Resolved' with a 'Fixed' resolution. The fix version is '3.2'. The reporter is 'Unassigned' and the assignee is 'Setb'. The issue was created on 23/Aug/11 12:50 and resolved on 26/Oct/13 01:06. The description states: 'NumberUtils.createLong() does not handle hex numbers, but createInteger() handles hex numbers. This seems odd. NumberUtils.createNumber() assumes that hex numbers can only be Integer. Again, why not handle bigger Hex numbers?' Annotations include 'Priority' pointing to the priority field, 'Versions' pointing to the fix version, and 'Time to fix' pointing to the resolution date.

bug report

```
diff --git a/source/org/jfree/data/DefaultKeyedValues.java b/source/org/jfree/data/DefaultKeyedValues.java
@@ -105,14 +105,7 @@ public class ShapeList extends AbstractObjectList {
     if (!(obj instanceof ShapeList)) {
         return false;
     }
     ShapeList that = (ShapeList) obj;
     int listSize = size();
     for (int i = 0; i < listSize; i++) {
         if (!ShapeUtil.equals((Shape) get(i), (Shape) that.get(i)))
             return false;
     }
     return true;
 }
 return super.equals(obj);
}

diff --git a/source/org/jfree/data/DefaultKeyedValues2D.java b/source/org/jfree/data/DefaultKeyedValues2D.java
@@ -452,19 +452,18 @@ public class DefaultKeyedValues2D implements KeyedValues2D,
     + Base removableComparable) {
     public void removeColumn(Comparable columnKey) {
         if (columnKey == null) {
             throw new IllegalArgumentException("Null 'columnKey' argument.");
         }
         if (this.columnKeys.contains(columnKey)) {
             throw new UnknownKeyException("Unknown Key: " + columnKey);
         }
         Iterator iterator = this.rows.iterator();
         while (iterator.hasNext()) {
             DefaultKeyedValues rowData = (DefaultKeyedValues) iterator.next();
             int index = rowData.getColumnIndex(columnKey);
             if (index >= 0) {
                 rowData.removeValue(columnKey);
             }
         }
         this.columnKeys.remove(columnKey);
     }
 }
```

Annotations on the code patch:

- adds new variable
- adds new loop
- adds new conditional
- deletes conditional
- file count
- line count

Developer-written patch

Measuring hardness and importance of a defect

Priority (Major)

Versions (lang.math.*)

Time to fix (Created: 23/Aug/11 12:50)

bug report

```
diff --git a/source/org/jfree/data/DefaultKeyedValues.java b/source/org/jfree/data/DefaultKeyedValues.java
@@ -106,14 +106,7 @@ public class ShapeList extends AbstractObjectList {
     if (!(obj instanceof ShapeList)) {
         return false;
     }
+    ShapeList that = (ShapeList) obj;
+    int listSize = size();
+    for (int i = 0; i < listSize; i++) {
+        if (!ShapeUtilities.equal((Shape) get(i), (Shape) that.get(i))) {
+            return false;
+        }
     }
     return true;
 }
@@ -452,19 +452,19 @@ public class DefaultKeyedValues2D implements KeyedValues2D,
     + Base removable(Comparable)
     +
+    + deletes conditional
     public void removeColumn(Comparable columnKey) {
         if (columnKey == null) {
             throw new IllegalArgumentException("Null 'columnKey' argument.");
         }
         if (!this.columnKeys.contains(columnKey)) {
             throw new IllegalArgumentException("Unknown key: " + columnKey);
         }
         Iterator iterator = this.rows.iterator();
         while (iterator.hasNext()) {
             DefaultKeyedValues rowData = (DefaultKeyedValues) iterator.next();
             int index = rowData.getColumnIndex(columnKey);
             if (index >= 0) {
                 rowData.removeValue(columnKey);
             }
             this.columnKeys.remove(columnKey);
         }
     }
 }

```

Developer-written patch

LOG	2014-01-23 19:22:45+01:00	/Users/tomaz/suites/global_scripts/AddressBook.py:11
FAIL	2014-01-23 19:22:47+01:00	/Users/tomaz/suites/suite_java_addressbook/test_case1/test.py:13
LOG	2014-01-23 19:22:47+01:00	/Users/tomaz/suites/global_scripts/AddressBook.py:15
LOG	2014-01-23 19:22:47+01:00	/Users/tomaz/suites/global_scripts/AddressBook.py:20
LOG	2014-01-23 19:22:48+01:00	/Users/tomaz/suites/global_scripts/AddressBook.py:20
LOG	2014-01-23 19:22:49+01:00	/Users/tomaz/suites/global_scripts/AddressBook.py:20
PASS	2014-01-23 19:22:49+01:00	/Users/tomaz/suites/suite_java_addressbook/test_case1/test.py:20
PASS	2014-01-23 19:22:49+01:00	/Users/tomaz/suites/suite_java_addressbook/test_case1/test.py:21

Test-suite

Measuring hardness and importance of a defect

Apache JIRA

Commons Lang / LANG-747

NumberUtils does not handle Long Hex numbers

Details

Type: Bug
Priority: Major
Affects Version/s: None
Component/s: lang.math
Labels: None

Status: Resolution: Fixed
Fix Version/s: lang.math: 3.2

People

Assignee: Unassigned
Reporter: Seb
Votes: 0
Watchers: 0

Dates

Created: 23/Aug/11 12:50
Updated: 15/Oct/13 02:33
Resolved: 26/Apr/13 01:06

Description

NumberUtils.createLong() does not handle hex numbers, but createInteger() handles hex numbers. This seems odd.
NumberUtils.createNumber() assumes that hex numbers can only be Integer.
Again, why not handle bigger Hex numbers?

bug report

```
diff --git a/source/org/apache/commons/lang3/data/DefaultKeyValueValues.java b/source/org/apache/commons/lang3/data/DefaultKeyValueValues.java
@@ -108,14 +108,7 @@ public class ShapeList extends AbstractObjectList {
     if (!(obj instanceof ShapeList)) {
         return false;
     }
     ShapeList that = (ShapeList) obj;
     int listSize = size();
     for (int i = 0; i < listSize; i++) {
         if (!ShapeUtils.equals(that.get(i), (Shape) that.get(i))) {
             return false;
         }
     }
     return true;
 }
 return super.equals(obj);
 }
 }
diff --git a/source/org/apache/commons/lang3/data/DefaultKeyValueValues2D.java b/source/org/apache/commons/lang3/data/DefaultKeyValueValues2D.java
@@ -452,19 +452,19 @@ public class DefaultKeyValueValues2D implements KeyValueValues2D,
     + BaseRemoveable{
     public void removeColumn(Comparable columnKey) {
         if (columnKey == null) {
             throw new IllegalArgumentException("Null 'columnKey' argument.");
         }
         if (!this.columnKeys.contains(columnKey)) {
             throw new IllegalArgumentException("Unknown key: " + columnKey);
         }
         Iterator iterator = this.rows.iterator();
         while (iterator.hasNext()) {
             DefaultKeyValueValues rowData = (DefaultKeyValueValues) iterator.next();
             int index = rowData.getIndex(columnKey);
             if (index >= 0) {
                 rowData.removeValue(columnKey);
             }
         }
         this.columnKeys.remove(columnKey);
     }
 }
```

adds new variable

adds new loop

adds new conditional

deletes conditional

file count

line count

Developer-written patch

LOG	2014-01-23 19:22:45+01:00	/Users/tommasu/suites/global_scripts/AddressBook.py:11
FAIL	2014-01-23 19:22:47+01:00	/Users/tommasu/suites/suite_java_addressbook/test_case1/test.py:13
LOG	2014-01-23 19:22:47+01:00	/Users/tommasu/suites/global_scripts/AddressBook.py:15
LOG	2014-01-23 19:22:48+01:00	/Users/tommasu/suites/global_scripts/AddressBook.py:20
LOG	2014-01-23 19:22:48+01:00	/Users/tommasu/suites/global_scripts/AddressBook.py:20
LOG	2014-01-23 19:22:49+01:00	/Users/tommasu/suites/global_scripts/AddressBook.py:20
PASS	2014-01-23 19:22:49+01:00	/Users/tommasu/suites/suite_java_addressbook/test_case1/test.py:20
PASS	2014-01-23 19:22:49+01:00	/Users/tommasu/suites/suite_java_addressbook/test_case1/test.py:21

Test-suite

Other parameters may also exist.

Measuring hardness and importance of a defect

Analyzed 8 popular bug-tracking systems



ClearQuest



Analyzed 3 popular open-source code repositories

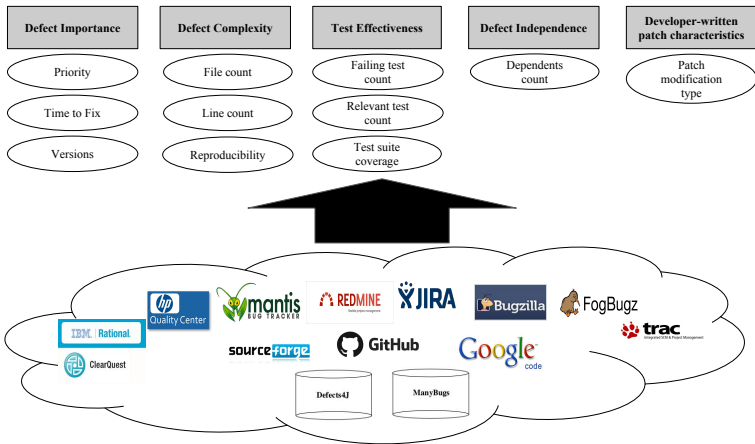


Analyzed 2 defect benchmarks

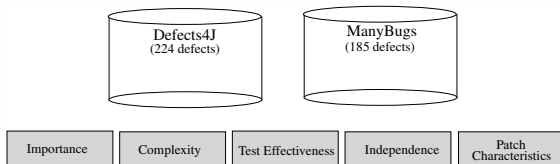


Measuring hardness and importance of a defect

5 defect characteristics defined in terms of 11 abstract parameters

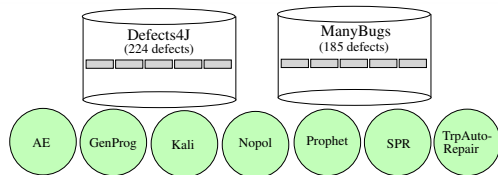


Evaluating repair techniques along new dimensions



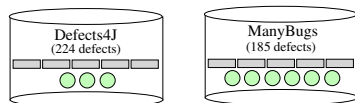
- ▶ 2 defect benchmarks: Defects4J and ManyBugs
- ▶ Semi-automatically annotated 409 defects with:
 - ▶ 5 defects characteristics defined using 11 abstract parameters.

Evaluating repair techniques along new dimensions



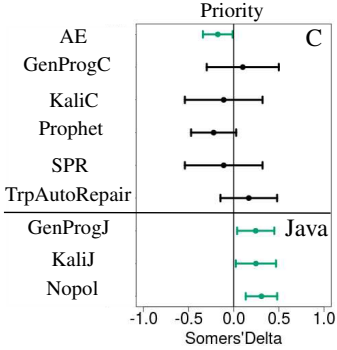
- ▶ 2 defect benchmarks: Defects4J and ManyBugs
- ▶ Semi-automatically annotated 409 defects with:
 - ▶ 5 defects characteristics defined using 11 abstract parameters.
 - ▶ Existing reparability and repair quality results of 7 automated repair techniques.

Evaluating repair techniques along new dimensions



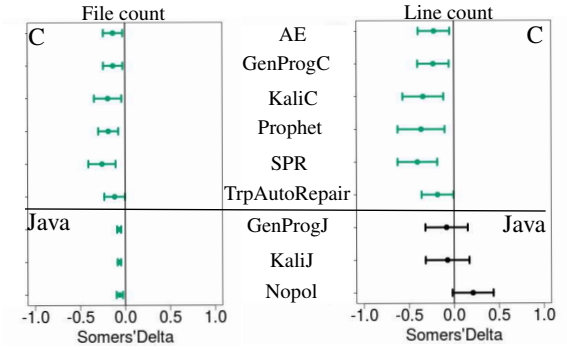
- ▶ 2 defect benchmarks: Defects4J and ManyBugs
- ▶ Semi-automatically annotated 409 defects with:
 - ▶ 5 defects characteristics defined using 11 abstract parameters.
 - ▶ Existing reparability and repair quality results of 7 automated repair techniques.
- ▶ Identify if reparability of a repair technique correlates (Somers's Delta $\in [-1, 1]$) with each abstract parameter.

Do repair techniques repair important defects?



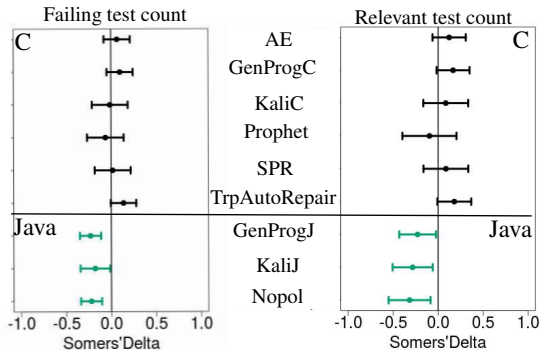
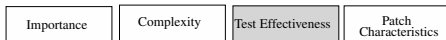
Java repair techniques are more likely to repair defects that are important for developers.

Do repair techniques repair **hard** defects?



C repair techniques are less likely to repair defects that required developers to write more code.

Do repair techniques repair defects with effective test suites?



Java repair techniques are less likely to repair defects with effective test suites.

What patch modification types are challenging for automated repair?



9 Patch modification types [1]

```
00 -309,14 +300,7 @@ public class ShapeUtil extends AbstractComparable {
-   if (!(obj instanceof ShapeUtil))
+   return false;
+   ShapeUtil that = (ShapeUtil) obj;
+   int listSize = size();
+   for (int i = 0; i < listSize;
+   if (!ShapeUtilUtil.equal(Shape) get(i), (Shape) that.get(i))) {
+   return false;
+   }
+   return true;
+   return super.equals(obj);
}

00 -482,19 +482,10 @@ public class DefaultValueValues0 implements KeyedValues0,
+   * @see @SuppressWarnings("rawtypes")
+   public void removeColumn(Comparable columnKey) {
+   if (columnKey == null) {
+   throw new IllegalArgumentException("Null 'columnKey' argument.");
+   }
+   if (!this.columnKeys.contains(columnKey)) {
+   throw new UnsupportedOperationException("Unknown key: " + columnKey);
+   }
+   Iterator iterat

00 -261,42 +260,12 @@ public class ExtendedMessageFormat extends MessageFormat {
+   * returns the hashCode
+   @Override
+   public int hashCode() {
+   int result = super.hashCode();
+   result = HASH_SEED * result + ObjectUtil.hashCode(registry);
+   result = HASH_SEED * result + ObjectUtil.hashCode(pattern);
}

00 -372,7 +371,7 @@ public class Week extends RegularTimePeriod implements Serializable {
+   public Week(Date time, TimeZone zone) {
+   // defer argument checking...
+   this(time, zone, Locale.getDefault());
+   this(time, RegularTimePeriod.DEFAULT_TIME_ZONE, Locale.getDefault());
}

00 -869,7 +869,6 @@ public class GetDateFormat extends Format {
+   public StringBuffer format(Calendar calendar, StringBuffer buf) {
+   if (calendar.isLeapYear()) {
+   calendar_getTime(); // LANG_S1B
+   calendar = (Calendar) calendar.clone();
+   calendar_setTimezone(OTIMEZONE);
}

00 -5649,7 +5649,7 @@ public class SystemUtils {
+   * Return the version, for example 131 for Java 1.3.1
+   *
+   * @return int toJavaVersionInt(String version) {
+   * static float toJavaVersionInt(String version) {
+   * static float toJavaVersionIntArray(String version, JAVA_VERSION_TRIP_SIZE);
}


```

adds one or more variables

adds one or more loops

adds one or more if statements

changes one or more conditionals

adds one or more method calls

changes one or more data structures or types

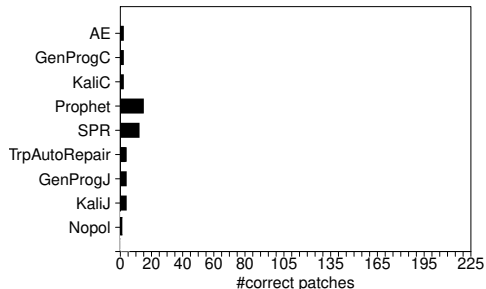
adds one or more new methods

changes one or more method arguments

changes one or more method signatures

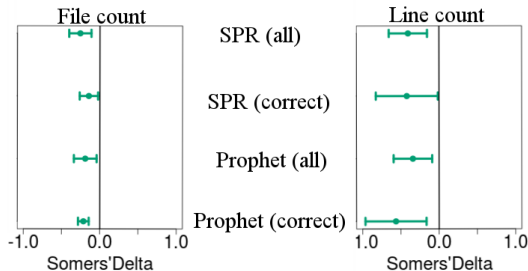
Defects that required developers to **add loops or a new method call, or change a method signature** are challenging for automated repair techniques to patch.

What about **correct** patches?



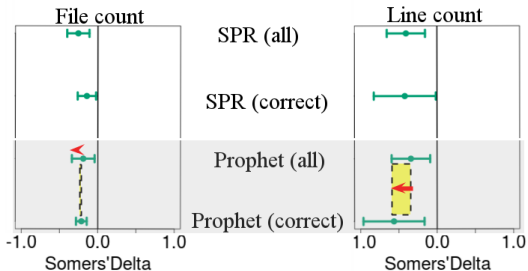
Only Prophet (15) and SPR (13) generate sufficient number of correct patches.

What about correct patches?



Prophet is less likely to produce patches for more complex defects, and even less likely to produce correct patches for the same defects.

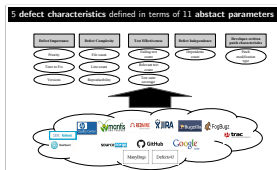
What about correct patches?



Prophet is less likely to produce patches for more complex defects, and even less likely to produce correct patches for the same defects.

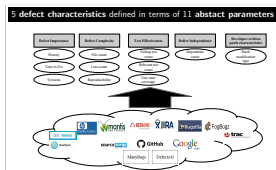
Contributions

Methodology to measure importance and hardness of a defect.

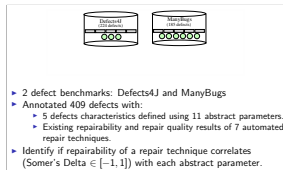


Contributions

Methodology to measure importance and hardness of a defect.

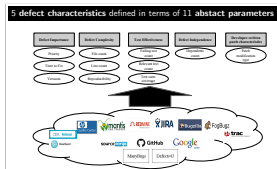


Methodology to evaluate automated program repair techniques along new dimensions.

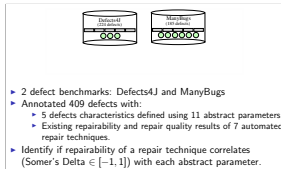


Contributions

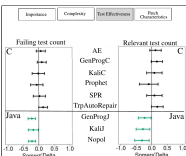
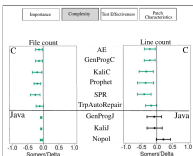
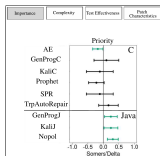
Methodology to measure importance and hardness of a defect.



Methodology to evaluate automated program repair techniques along new dimensions.

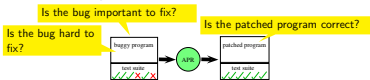


Evaluation of 7 automated program repair techniques on 409 real-world defects.



Recommendations

Repair research should evaluate if new techniques repair hard and important defects.



Hard to fix defects

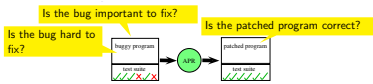
Yet Another Fix	
Fixes 60% of the defects	
Defect-1	patched
Defect-2	patched
Defect-3	not patched
Defect-4	patched
Defect-5	patched
Defect-6	not patched
Defect-7	patched
Defect-8	patched
Defect-9	not patched
Defect-10	not patched

This Never Ends Fix	
Fixes 30% of the defects	
Defect-1	not patched
Defect-2	not patched
Defect-3	patched
Defect-4	not patched
Defect-5	not patched
Defect-6	not patched
Defect-7	not patched
Defect-8	not patched
Defect-9	patched
Defect-10	patched

Which automated program repair technique is better?

Recommendations

Repair research should evaluate if new techniques repair hard and important defects.



Yet Another Fix fixes 67% of the defects		This Never Ends Fix fixes 30% of the defects	
Defect-1	patched	Defect-1	not patched
Defect-2	patched	Defect-2	not patched
Defect-3	not patched	Defect-3	patched
Defect-4	patched	Defect-4	not patched
Defect-5	patched	Defect-5	not patched
Defect-6	not patched	Defect-6	not patched
Defect-7	patched	Defect-7	not patched
Defect-8	patched	Defect-8	not patched
Defect-9	not patched	Defect-9	patched
Defect-10	not patched	Defect-10	patched

Which automated program repair technique is better?

Repair research should target defects that existing techniques have missed.

```

1 public void printMsg(Message message) {
2     return false;
3 }
4
5 if (message instanceof ShortMessage) {
6     return false;
7 }
8
9 Message msg = (Message) message;
10 int lifetime = msg.getLifetime();
11 for (int i = 0; i < lifetime; i++) {
12     if (message instanceof ShortMessage) {
13         return false;
14     }
15     return true;
16 }
17
18 return true;
19 }
20
21 @Override
22 public boolean handleMessage(Message message) {
23     return false;
24 }
25
26 @Override
27 public boolean handleMessage(Message message) {
28     return false;
29 }
30
31 @Override
32 public boolean handleMessage(Message message) {
33     return false;
34 }
35
36 @Override
37 public boolean handleMessage(Message message) {
38     return false;
39 }
40
41 @Override
42 public boolean handleMessage(Message message) {
43     return false;
44 }
45
46 @Override
47 public boolean handleMessage(Message message) {
48     return false;
49 }
50
51 @Override
52 public boolean handleMessage(Message message) {
53     return false;
54 }
55
56 @Override
57 public boolean handleMessage(Message message) {
58     return false;
59 }
60
61 @Override
62 public boolean handleMessage(Message message) {
63     return false;
64 }
65
66 @Override
67 public boolean handleMessage(Message message) {
68     return false;
69 }
70
71 @Override
72 public boolean handleMessage(Message message) {
73     return false;
74 }
75
76 @Override
77 public boolean handleMessage(Message message) {
78     return false;
79 }
80
81 @Override
82 public boolean handleMessage(Message message) {
83     return false;
84 }
85
86 @Override
87 public boolean handleMessage(Message message) {
88     return false;
89 }
90
91 @Override
92 public boolean handleMessage(Message message) {
93     return false;
94 }
95
96 @Override
97 public boolean handleMessage(Message message) {
98     return false;
99 }
100
101 @Override
102 public boolean handleMessage(Message message) {
103     return false;
104 }
105
106 @Override
107 public boolean handleMessage(Message message) {
108     return false;
109 }
110
111 @Override
112 public boolean handleMessage(Message message) {
113     return false;
114 }
115
116 @Override
117 public boolean handleMessage(Message message) {
118     return false;
119 }
120
121 @Override
122 public boolean handleMessage(Message message) {
123     return false;
124 }
125
126 @Override
127 public boolean handleMessage(Message message) {
128     return false;
129 }
130
131 @Override
132 public boolean handleMessage(Message message) {
133     return false;
134 }
135
136 @Override
137 public boolean handleMessage(Message message) {
138     return false;
139 }
140
141 @Override
142 public boolean handleMessage(Message message) {
143     return false;
144 }
145
146 @Override
147 public boolean handleMessage(Message message) {
148     return false;
149 }
150
151 @Override
152 public boolean handleMessage(Message message) {
153     return false;
154 }
155
156 @Override
157 public boolean handleMessage(Message message) {
158     return false;
159 }
160
161 @Override
162 public boolean handleMessage(Message message) {
163     return false;
164 }
165
166 @Override
167 public boolean handleMessage(Message message) {
168     return false;
169 }
170
171 @Override
172 public boolean handleMessage(Message message) {
173     return false;
174 }
175
176 @Override
177 public boolean handleMessage(Message message) {
178     return false;
179 }
180
181 @Override
182 public boolean handleMessage(Message message) {
183     return false;
184 }
185
186 @Override
187 public boolean handleMessage(Message message) {
188     return false;
189 }
190
191 @Override
192 public boolean handleMessage(Message message) {
193     return false;
194 }
195
196 @Override
197 public boolean handleMessage(Message message) {
198     return false;
199 }
200
201 @Override
202 public boolean handleMessage(Message message) {
203     return false;
204 }
205
206 @Override
207 public boolean handleMessage(Message message) {
208     return false;
209 }
210
211 @Override
212 public boolean handleMessage(Message message) {
213     return false;
214 }
215
216 @Override
217 public boolean handleMessage(Message message) {
218     return false;
219 }
220
221 @Override
222 public boolean handleMessage(Message message) {
223     return false;
224 }
225
226 @Override
227 public boolean handleMessage(Message message) {
228     return false;
229 }
230
231 @Override
232 public boolean handleMessage(Message message) {
233     return false;
234 }
235
236 @Override
237 public boolean handleMessage(Message message) {
238     return false;
239 }
240
241 @Override
242 public boolean handleMessage(Message message) {
243     return false;
244 }
245
246 @Override
247 public boolean handleMessage(Message message) {
248     return false;
249 }
250
251 @Override
252 public boolean handleMessage(Message message) {
253     return false;
254 }
255
256 @Override
257 public boolean handleMessage(Message message) {
258     return false;
259 }
260
261 @Override
262 public boolean handleMessage(Message message) {
263     return false;
264 }
265
266 @Override
267 public boolean handleMessage(Message message) {
268     return false;
269 }
270
271 @Override
272 public boolean handleMessage(Message message) {
273     return false;
274 }
275
276 @Override
277 public boolean handleMessage(Message message) {
278     return false;
279 }
280
281 @Override
282 public boolean handleMessage(Message message) {
283     return false;
284 }
285
286 @Override
287 public boolean handleMessage(Message message) {
288     return false;
289 }
290
291 @Override
292 public boolean handleMessage(Message message) {
293     return false;
294 }
295
296 @Override
297 public boolean handleMessage(Message message) {
298     return false;
299 }
300
301 @Override
302 public boolean handleMessage(Message message) {
303     return false;
304 }
305
306 @Override
307 public boolean handleMessage(Message message) {
308     return false;
309 }
310
311 @Override
312 public boolean handleMessage(Message message) {
313     return false;
314 }
315
316 @Override
317 public boolean handleMessage(Message message) {
318     return false;
319 }
320
321 @Override
322 public boolean handleMessage(Message message) {
323     return false;
324 }
325
326 @Override
327 public boolean handleMessage(Message message) {
328     return false;
329 }
330
331 @Override
332 public boolean handleMessage(Message message) {
333     return false;
334 }
335
336 @Override
337 public boolean handleMessage(Message message) {
338     return false;
339 }
340
341 @Override
342 public boolean handleMessage(Message message) {
343     return false;
344 }
345
346 @Override
347 public boolean handleMessage(Message message) {
348     return false;
349 }
350
351 @Override
352 public boolean handleMessage(Message message) {
353     return false;
354 }
355
356 @Override
357 public boolean handleMessage(Message message) {
358     return false;
359 }
360
361 @Override
362 public boolean handleMessage(Message message) {
363     return false;
364 }
365
366 @Override
367 public boolean handleMessage(Message message) {
368     return false;
369 }
370
371 @Override
372 public boolean handleMessage(Message message) {
373     return false;
374 }
375
376 @Override
377 public boolean handleMessage(Message message) {
378     return false;
379 }
380
381 @Override
382 public boolean handleMessage(Message message) {
383     return false;
384 }
385
386 @Override
387 public boolean handleMessage(Message message) {
388     return false;
389 }
390
391 @Override
392 public boolean handleMessage(Message message) {
393     return false;
394 }
395
396 @Override
397 public boolean handleMessage(Message message) {
398     return false;
399 }
400
401 @Override
402 public boolean handleMessage(Message message) {
403     return false;
404 }
405
406 @Override
407 public boolean handleMessage(Message message) {
408     return false;
409 }
410
411 @Override
412 public boolean handleMessage(Message message) {
413     return false;
414 }
415
416 @Override
417 public boolean handleMessage(Message message) {
418     return false;
419 }
420
421 @Override
422 public boolean handleMessage(Message message) {
423     return false;
424 }
425
426 @Override
427 public boolean handleMessage(Message message) {
428     return false;
429 }
430
431 @Override
432 public boolean handleMessage(Message message) {
433     return false;
434 }
435
436 @Override
437 public boolean handleMessage(Message message) {
438     return false;
439 }
440
441 @Override
442 public boolean handleMessage(Message message) {
443     return false;
444 }
445
446 @Override
447 public boolean handleMessage(Message message) {
448     return false;
449 }
450
451 @Override
452 public boolean handleMessage(Message message) {
453     return false;
454 }
455
456 @Override
457 public boolean handleMessage(Message message) {
458     return false;
459 }
460
461 @Override
462 public boolean handleMessage(Message message) {
463     return false;
464 }
465
466 @Override
467 public boolean handleMessage(Message message) {
468     return false;
469 }
470
471 @Override
472 public boolean handleMessage(Message message) {
473     return false;
474 }
475
476 @Override
477 public boolean handleMessage(Message message) {
478     return false;
479 }
480
481 @Override
482 public boolean handleMessage(Message message) {
483     return false;
484 }
485
486 @Override
487 public boolean handleMessage(Message message) {
488     return false;
489 }
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600

```


Evaluation Methodology

