When

Do Changes Induce Fixes?

On Fridays.

Jacek Śliwerski, Thomas Zimmermann, Andreas Zeller
Saarland University
The Risk of Change

- Which changes go wrong?
- How do we find such “bad” changes?
- What can we do with them?
The Risk of Change

- Which changes go wrong?
- How do we find such “bad” changes?
- What can we do with them?

Idea:
- combine bug databases + version archives
  => locate fix-inducing changes
Fix-Inducing Changes

Fix-Inducing Changes are Changes that lead to problems as indicated by later fixes.

Example:

```
... if (foo==null) {
    foo.bar();
}
```

Later:

```
... if (foo!=null) {
    foo.bar();
}
```

FIX-INDUCING FIX
How to Locate Fix-Inducing Changes
How to Locate Fix-Inducing Changes

$ cvs annotate -r 1.17 Foo.java
...
19: 1.11 (john 12-Feb-03): public int a() {
20: 1.11 (john 12-Feb-03):     return i/0;
...
39: 1.10 (mary 12-Jan-03): public int b() {
40: 1.14 (kate 23-May-03):     return 42;
...
59: 1.10 (mary 17-Jan-03): public void c() {
60: 1.16 (mary 10-Jun-03):     int i=0;

Fixed Bug
42233
Changed:
a() b() c()
How to Locate Fix-Inducing Changes

$ cvs annotate -r 1.17 Foo.java

20: 1.11 (john 12-Feb-03):     return i/0;
40: 1.14 (kate 23-May-03):     return 42;
60: 1.16 (mary 10-Jun-03):     int i=0;

FIX-INDUCING

1.11 a() was changed
1.14 b() was changed
1.16 c() was changed
1.18 Fixed Bug 42233
Changed: a() b() c()
Bug 42233 was reported.

How to Locate Fix-Inducing Changes

$ cvs annotate -r 1.17 Foo.java

20: 1.11 (john 12-Feb-03):     return i/0;
40: 1.14 (kate 23-May-03):     return 42;
60: 1.16 (mary 10-Jun-03):     int i=0;

Fixed Bug 42233

Changed:

a() b() c()
Bug-Inducing Changes are Indicators for Risk
Large Transactions are Risky

Average Number of Changed Files

<table>
<thead>
<tr>
<th></th>
<th>Bug-Inducing</th>
<th>Not Bug-Inducing</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fix</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not a Fix</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td></td>
<td></td>
<td><strong>3.52</strong></td>
</tr>
</tbody>
</table>

ECLIPSE, for MOZILLA the average is 3.58.
Large Transactions are Risky

Average Number of Changed Files

<table>
<thead>
<tr>
<th></th>
<th>Bug-Inducing</th>
<th>Not Bug-Inducing</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fix</td>
<td></td>
<td></td>
<td>2.73</td>
</tr>
<tr>
<td>Not a Fix</td>
<td></td>
<td></td>
<td>3.81</td>
</tr>
<tr>
<td>All</td>
<td></td>
<td></td>
<td>3.52</td>
</tr>
</tbody>
</table>

ECLIPSE, for MOZILLA the other way round.
Large Transactions are Risky

**Average Number of Changed Files**

<table>
<thead>
<tr>
<th></th>
<th>Bug-Inducing</th>
<th>Not Bug-Inducing</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fix</td>
<td></td>
<td></td>
<td>2.73</td>
</tr>
<tr>
<td>Not a Fix</td>
<td></td>
<td></td>
<td>3.81</td>
</tr>
<tr>
<td>All</td>
<td><strong>7.49</strong></td>
<td><strong>2.61</strong></td>
<td><strong>3.52</strong></td>
</tr>
</tbody>
</table>

ECLIPSE, for MOZILLA the same tendency.
Large Transactions are Risky

Average Number of Changed Files

<table>
<thead>
<tr>
<th></th>
<th>Bug-Inducing</th>
<th>Not Bug-Inducing</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fix</td>
<td>3.82</td>
<td>2.08</td>
<td>2.73</td>
</tr>
<tr>
<td>Not a Fix</td>
<td>11.30</td>
<td>2.77</td>
<td>3.81</td>
</tr>
<tr>
<td>All</td>
<td>7.49</td>
<td>2.61</td>
<td>3.52</td>
</tr>
</tbody>
</table>

ECLIPSE, for MOZILLA a different distribution.
Fridays are Risky,
Tuesdays are not ;-)
Fixes are very Risky

Likelihood that a Change is Bug-Inducing

22.9% of all fixes on Tuesday induce bugs, only 7.1% of changes for new features do.

<table>
<thead>
<tr>
<th>Day</th>
<th>Fixes (%)</th>
<th>New Features, Refactoring (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday</td>
<td>8.2%</td>
<td>25.1%</td>
</tr>
<tr>
<td>Tuesday</td>
<td>7.1%</td>
<td>22.9%</td>
</tr>
<tr>
<td>Wednesday</td>
<td>8.1%</td>
<td>23.3%</td>
</tr>
<tr>
<td>Thursday</td>
<td>8.8%</td>
<td>23.5%</td>
</tr>
<tr>
<td>Friday</td>
<td>8.7%</td>
<td>23.2%</td>
</tr>
<tr>
<td>Saturday</td>
<td>8.4%</td>
<td>30.3%</td>
</tr>
<tr>
<td>Sunday</td>
<td>8.6%</td>
<td>26.4%</td>
</tr>
</tbody>
</table>
Is MOZILLA broken?

- Fix transactions are larger than regular transactions (4.39 vs 3.05 files).
- 48.5% of all revisions are fixes, 41.5% induce later bugs.
- A fix induces in 45.2% a later bugs, regular changes in “only” 38.1%.
For earthquakes and crime rates...

Risk depends on Locations

The same with risk of changes!
public IRuntimeClasspathEntry[] resolveClasspath(IRuntimeClasspathEntry[] entries, ILaunchConfiguration configuration) throws CoreException {
    List all = new ArrayList(entries.length);
    for (int i = 0; i < entries.length; i++) {
        switch (entries[i].getType()) {
            case IRuntimeClasspathEntry.PROJECT:
                all.add(entries[i]);
                break;
            case IRuntimeClasspathEntry.OTHER:
                IRuntimeClasspathEntry2 entry = (IRuntimeClasspathEntry2) entries[i];
                if (entry.getTypeId().equals(DefaultProjectClasspathEntry.TYPE_ID)) {
                    IRuntimeClasspathEntry[] children = entry.getRuntimeClasspathEntries(configuration);
                    IRuntimeClasspathEntry[] res = JavaRuntime.resolveSourceLookupPath(children, configuration);
                    for (int j = 0; j < res.length; j++) {
                        all.add(res[j]);
                    }
                }
                break;
            default:
                IRuntimeClasspathEntry[] resolved = JavaRuntime.resolveRuntimeClasspathEntry(entries[i], configuration);
                for (int j = 0; j < resolved.length; j++) {
                    all.add(resolved[j]);
                }
                break;
            }
    }
    return (IRuntimeClasspathEntry[]) all.toArray(new IRuntimeClasspathEntry[all.size()]);
}
HATARI
Raising Risk Awareness

Movie with John Wayne (1962)

Swahili for “Danger”
HATARI: Annotations

Unrisky Location (green)

Risky Location (dark red)
Conclusion

- Fix- and Bug-inducing changes are an measure for (past) risk.
- We make this risk visible for developers.
- HATARI will be released in August 2005
  http://www.st.cs.uni-sb.de/softevo/
- Until then: Don’t program on Fridays! ;-)
Sign Gallery
DANGER
ONLY AUTHORIZED PROGRAMMERS MAY CHANGE FUNCTION
NOTICE
DON'T EVEN THINK OF CHANGING
DANGER

DO NOT PROGRAM ON FRIDAYS!