Characterizing, Verifying, and Improving Software Resilience with Exception Contracts and Test Suites

BENEVOL 2013 - UMONS

CORNU Benoit – INRIA / Adam – 16/12/2013

Expected Problems

Unexpected Errors

Goal

Resilience : Capability of a system to heal itself in presence of unexpected errors

Our goal is to identify pieces of codes that are capable to handle unexpected errors.

$Exception \subset Error$

```
try{
   A();
   B();
   C();
}catch(MyException e){
   D();
}finally{
   E();
}
```

- No exception
  - A B C E
- MyException on B
  - A B D E
- OtherException on B
  - A B E

Expected errors in Test Suite ?

- Test colors
  - Pink
  - Blue
  - White

```
public String getProperty(String s) {
    String res = null;
    try{
        res = getPropertyFromFile(s);
    }catch(PropertyNotFoundException pnfe){
        return null;
    }
    return res;
}

@Test
public void testAbsentProperty(){
    assertNull(getProperty("@$%^ù*µ;?!"));
}
```

Formal definition

	Exception thrown In app code	Exception caught in app code	Exception reaching the test code
pink	0	0	0
White (expected)	>0	>0	0
Blue (expected)	>0	>=0	>0

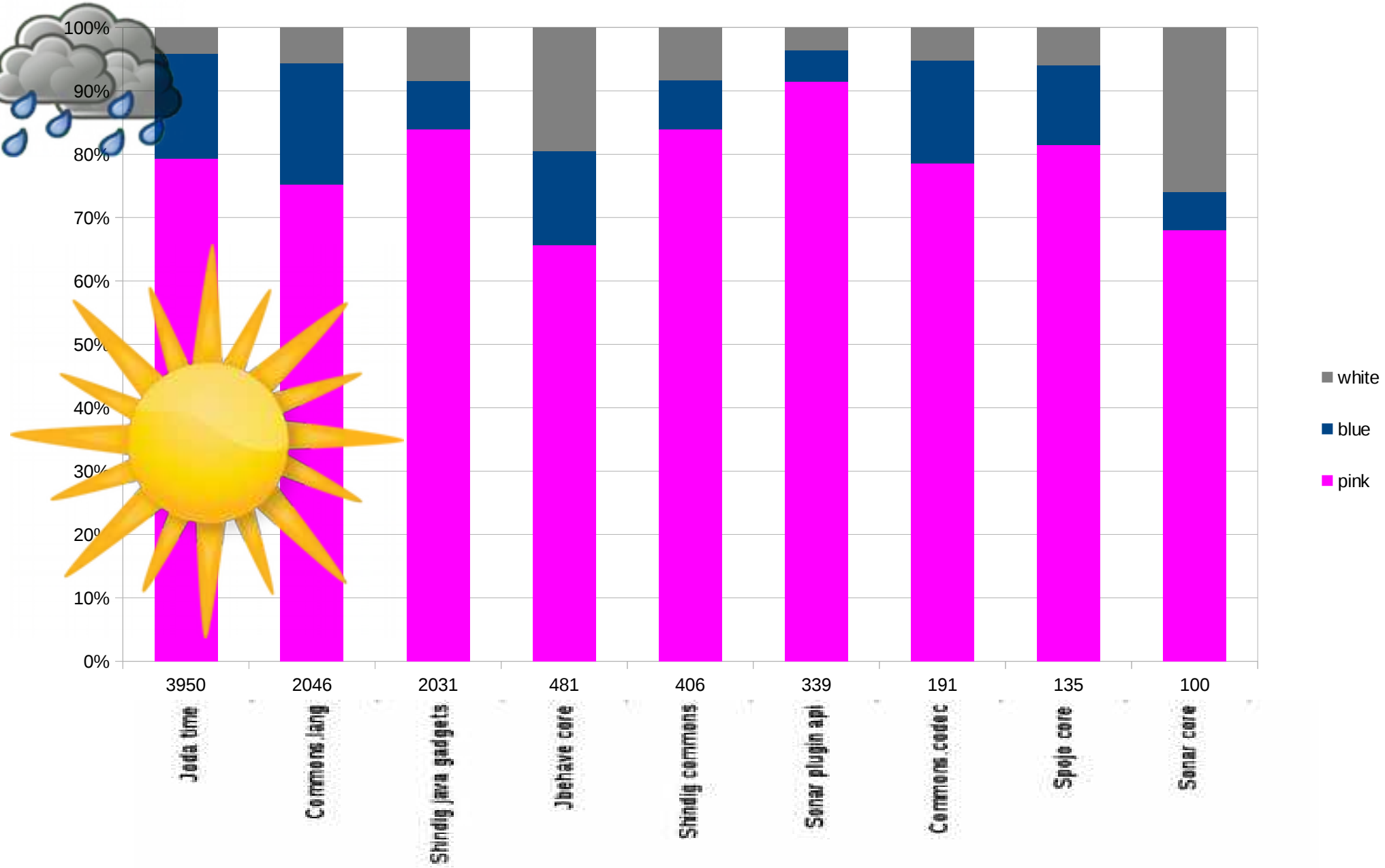
How to determine test cases' color ?

```
try{
    //developper code
}catch(RuntimeException re){
    //developper code
}
```

→ Code Transformation →

```
try{
    Framework.learn("try-start");
    //developper code
    Framework.learn("try-end");
}catch(RuntimeException re){
    Framework.learn("catch-start", re);
    //developper code
    Framework.learn("catch-end");
}
```

There are expected errors

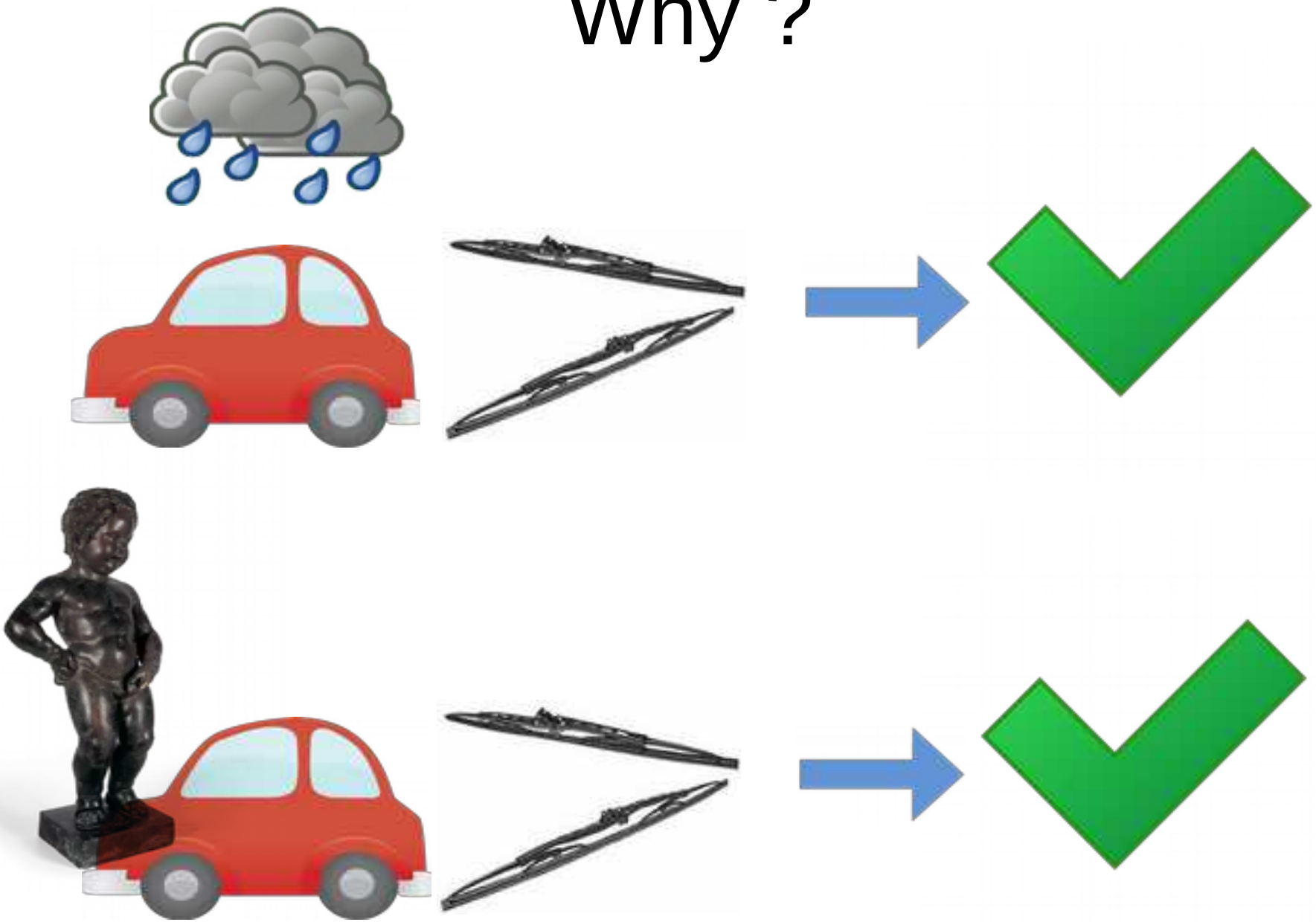


Unexpected Errors

Simulating unexpected errors

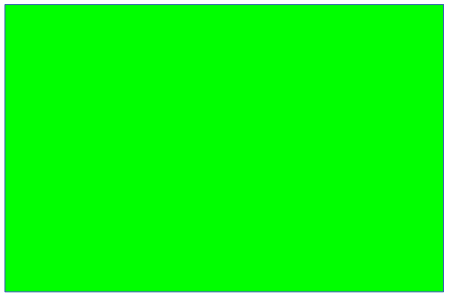
Study behavior of code under unexpected errors

Why ?



Oracle ?

The system state is correct



```
@Test
public void testPropertyNominal(){
    assertEquals("a test value", getProperty("test"));
}

@Test(expected=NullPointerException.class)
public void testPropertyNull(){
    getProperty(null);
}
```

Short-Circuit Testing

White test

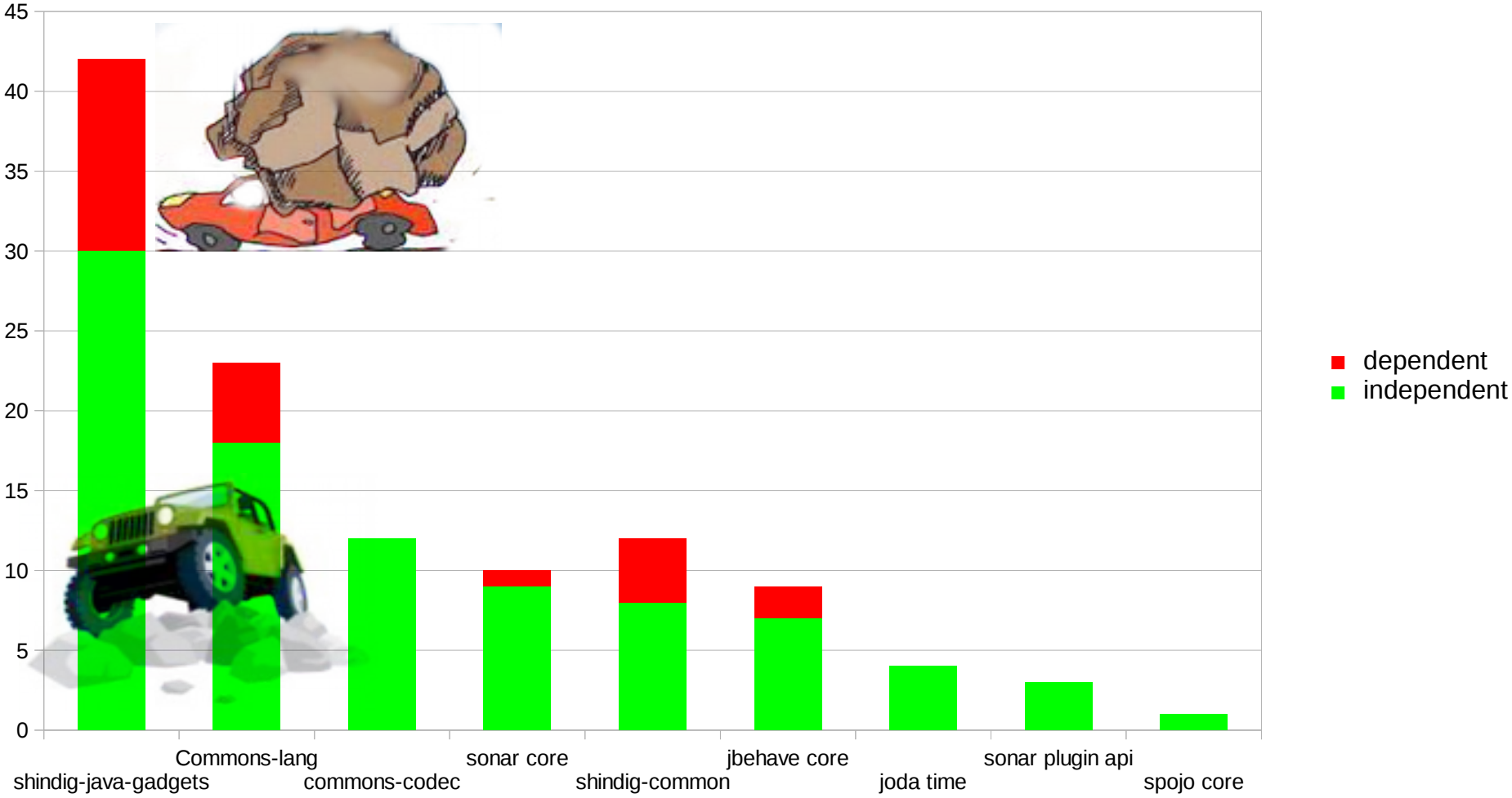
+ throw new X()

=

The try-catch blocks that keep the test green are capable to handle unexpected errors.

We call them "fault-independent"

There are fault independent try-catchs



Related Work

- Amplifying tests to validate exception handling code (Zhang and Elbaum, ICSE 2012)

- Using fault injection to increase software test coverage (Bieman and al., 1996)

Conclusion

- Definition of 3 new test case kinds (colors).
- Algorithm for identifying try-catch able to handle unexpected exceptions.
- Empirical discovery of fault-independent (92) try-catch blocks.

benoit.cornu@inria.fr