Disposable Testing: Avoiding Maintenance of Generated Unit Tests by Throwing Them Away

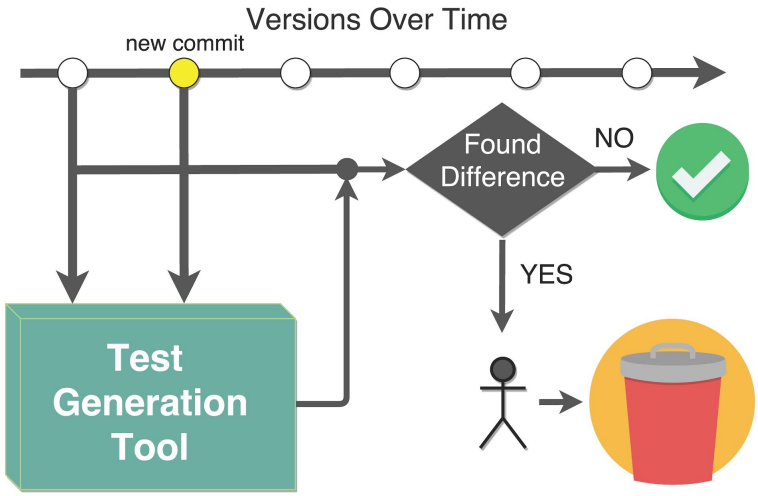
Sina Shamshiri sina.shamshiri@sheffield.ac.uk
José Campos jose.campos@sheffield.ac.uk
Gordon Fraser gordon.fraser@sheffield.ac.uk
Phil McMinn p.mcminn@sheffield.ac.uk

Would you like to maintain tests like this?

```
@Test
public void test() {
  String string0 = "Z,~jsZ/7'{p!wd";
  int int0 = 0;
  SimpleTimeZone simpleTimeZone0 = new SimpleTimeZone(int0, string0);
  Locale locale0 = Locale.GERMAN;
  String string1 = "*z";
  FastDatePrinter fastDatePrinter0 = new FastDatePrinter(string1, simpleTimeZone0, locale0);
  MockGregorianCalendar mockGregorianCalendar0 = new MockGregorianCalendar(locale0);
  String string2 = fastDatePrinter0.format((Calendar) mockGregorianCalendar0);
  assertEquals("*GMT", string2);
}
```

Writing good **tests** is difficult and **tedious**.
Maintaining automatically **generated tests** due to APIs changes or outdated tests is still a **manual task**.
Maintaining automatically **generates tests** can be **more difficult than** maintaining **manually written** ones.

Generate new tests after each change and then throw them away



*We propose **disposable testing:***

Completely **new tests** are generated every time the program under test is changed.

Only **tests** that **reveal a behavioral difference** caused by the change are shown to developers.

Developers then decide whether this difference is intended or not, and generated **tests are thrown away**.

But, is it possible?

Generating effective change-revealing tests

★ Automated **test generation techniques** have to be **effective at revealing** behavioral **differences**
★ **Differential Testing** better suited to implement **disposable testing**?
★ Preliminary results show that **differential testing** can be equally or more **effective** than traditional coverage-driven testing **at revealing real bugs**

Is the maintenance effort really reduced?

★ **No maintenance effort** because generated test cases are discarded. However, **failing tests** still need to be **inspected** to determine whether it is due to an intended change or regression fault
★ Preliminary results show that the **inspection effort is not increased by disposable testing** when compared to a traditional generate-and-maintain approach