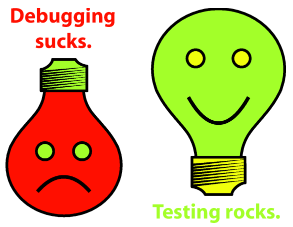


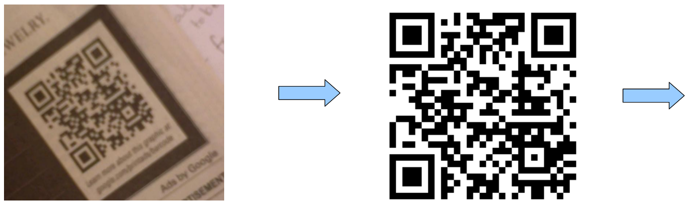
## Testing on the Toilet

# A Matter of Black and White

August 7, 2008

The progressive developer knows that in this complex modern world, things aren't always **black-and-white**, even in testing. Sometimes we know the software won't return the best answer, or even a correct answer, for all input. You may be tempted to write only test cases that your software can pass. After all, we can't have automated tests that fail in even one instance. But, this would miss an opportunity.

Speaking of black-and-white, take decoding of **two-dimensional barcodes, like QR Codes**. From a blurry, skewed, rotated image of a barcode, software has to pick it out, transform it, and decode it:



`http://google.com/gwt/n?u=bluenile.com`

Even the best software can't always find that barcode. What should tests for such software do?

We have some answers, from experience testing such software. We have two groups of black-box tests that verify that images decode correctly: **must-have** and **nice-to-have**. Tests verify that the must-have set – the easy images – definitely decode correctly. This is what traditional tests would include, which typically demand a 100% pass rate. But we also see how well we do on the more difficult nice-to-have set. We might verify that 50% of them decode, and fail otherwise.

The advantage? We can include tougher test cases in unit tests, instead of avoiding them. We can observe small changes – improvements as well as degradations – in decode accuracy over time. It doubles as a crude quality evaluation framework.

Where can this progressive thinking be applied? Maybe when your code...

**Only needs to be correct in most cases**. As here, write tests to verify easy cases work, but also that *some* hard cases pass too.

**Needs to be fast**. You write unit tests that verify it runs "fast enough" on simple input. How about writing tests that make sure it runs "fast enough" on most of some larger inputs too?

**Is heuristic**. You write unit tests that verify that the answer is “really close” to optimal on simple input, but also that it's “kind of close” on difficult input.

ZXING

By the way, did we mention project **ZXing**, Google's open-source decoder project? Or that **Print Ads** is already helping clients place these two-dimensional barcodes in the New York Times? Or that there are other formats like **Data Matrix**? or that you can put **more than just a URL** in these barcodes? This is a technology going global, so, time to read up on it.

**More information, discussion, and archives:**
**http://googletesting.blogspot.com**