

THIS TALK IS BROKEN

Sage LaTorra

Software
Development
Engineer in Test II

Microsoft

Me



A historical black and white photograph showing the crew of the USS Monitor on the deck of their ironclad ship. In the foreground, two sailors are crouching; one is looking directly at the camera while the other looks slightly away. Behind them, several sailors stand in groups, some with their hands crossed. One sailor stands prominently on the central gun platform. The ship's hull, with its characteristic dome-shaped gun mount, is visible in the background under a clear sky.

TITUS ANDRONICUS

The monitor

INFINITE JEST

a novel

DAVID VANCE



Seven Samurai

A film by Akira Kurosawa













RUSH
2 YES

The floggings will continue
until morale improves



```

$_='ev
al("seek\040D
0;");foreach(1..3)
@camel1hump;my$camel;
<DATA>) {$_=sprintf("%-6
1=split("//");if(defined($
p=split("//");}while(@dromeda
;my$CAMEL=3;if(defined($_=shif
t(@dromedary1))&&/\S/){$camel1hump+=1<<$CAMEL;}
$CAMEL--;if(defined($_=shift(@dromedary1))&&/\S/){
$camel1hump+=1 <<$CAMEL;}$CAMEL--;if(defined($_=shift(
@camel1hump))&&/\S/){$camel1hump+=1<<$CAMEL;}$CAMEL--;if(
defined($_=shift(@camel1hump))&&/\S/){$camel1hump+=1<<$CAME
L;;}$camel.= (split("//","\040..m`{/\J\047\134}L^7FX"))[$camel1h
ump];}$camel.= "\n";}@camel1hump=split(/\n/, $camel);foreach(@
camel1hump){chomp;$Camel=$_;y/LJF7\173\175\047\061\062\063\
064\065\066\067\070/;y/12345678/JL7F\175\173\047/;$_=reverse;
print "$_\040$Camel\n";}foreach(@camel1hump){chomp;$Camel=$_;y
/LJF7\173\175\047/12345678/;y/12345678/JL7F\175\173\047/;
$_=reverse;print "\040$$_$Camel\n";}';;s/\s*//g;;eval; eval
("seek\040DATA,0,0;");undef$/;$_=<DATA>;s/\s*//g;();s
;^.*;;:map{eval"print \"$_\040\";"}/.{4}/g; DATA \124
\1 50\145\040\165\163\145\040\157\1 46\040\1 41\0
40\143\141 \155\145\1 54\040\1 51\155\ 141
\147\145\0 40\151\156 \040\141 \163\16 3\
157\143\ 151\141\16 4\151\1 57\156
\040\167 \151\164\1 50\040\ 120\1
45\162\ 154\040\15 1\163\ 040\14
1\040\1 64\162\1 41\144 \145\
155\14 1\162\ 153\04 0\157
\146\ 040\11 7\047\ 122\1
45\15 1\154\1 54\171 \040
\046\ 012\101\16 3\16
3\15 7\143\15 1\14
1\16 4\145\163 \054
\040 \111\156\14 3\056
\040\ 125\163\145\14 4\040\
167\1 51\164\1 50\0 40\160\
145\162 \163\163 \155\151
\163\163 \151\1
57\156\056

```



S U B V E R S I O N



TM

Microsoft®
Silverlight™

Start

mattberg@live.com

RE: Wedding Announcement!
Congratulations! I'm really excited to
celebrate with you all. Thanks for the ...



12



First ever sufboard kickflip
recorded in Santa Cruz



Internet Explorer



Now Playing
Eliza
Viper Creek Club
Letters



Photos



Store



Messaging



78° Anaheim, CA



Tomorrow 88° / 65°



Thursday 87° / 65°



Friday 89° / 65°

Weather

Investments ▲ 2.3%

▲ 3.6%

Roth IRA (TD Ameritrade)

▲ 2.7%

401K Fund (Fidelity)

▼ 1.4%

Investment Portfolio



13
TUESDAY



CATEGORIES

Top News**Sports****Vancouver 2010****Showbiz****U.S. News****World****Wacky****Technology****Business****Lifestyle**

The smartphone screen shows the Associated Press mobile website. At the top is the AP logo and the word "associated". Below that is a large, semi-transparent image of a man's face. On the left side of the screen, there is a "TOP NEWS" section with a thumbnail of President Obama at a podium. To the right of the thumbnail is the headline "Obama open to adding GOP ideas to health plan" and the subtext "WASHINGTON (AP)". Below this is a snippet of text: "President Barack Obama said Tuesday he was open to four new Republican proposals on health care legislation, in a gesture of bipartisanship meant to jump-". Underneath this is a comment icon with "21 Comments". Further down is another news item with a thumbnail of people wearing 3D glasses. The headline is "Precious in 3-D? Filmmakers predict 3-D dramas" and the subtext "NEW YORK (AP)". A snippet of text follows: "Movies are evolving ever more into 3-D, a shake-up of the medium that has been compared to the advent of sound or color in motion pictures.". At the bottom of the screen are three navigation icons: a back arrow, a Windows logo, and a magnifying glass.

The right side of the image shows a vertical news feed from the Associated Press. At the top is a "MOST SHARED" section with a thumbnail of a baseball player, the headline "Many maple bats get banned in minors", and social sharing metrics: 345 likes and 10 comments. Below this is a "digg" section with a thumbnail of a person, the headline "Medvedev demands resignations over", and 512 diggs. Further down is a "Smithsonian turns down suit worn by OJ" section with a thumbnail of two men, 345 likes, and 10 comments. At the bottom is a "Woods back from family counseling" section with a thumbnail of a golfer, the text "TWEETED BY @AP", and the headline "Reebok sued over".

Testing

Spot the Bug #1

```
def gcd(x, y):  
    r = x % y  
    if r == 0:  
        return y  
    else:  
        return gcd(y, r)
```

What if y isn't an integer? What if y is zero?

Functional Testing

Does it meet the spec?

Is the spec complete?

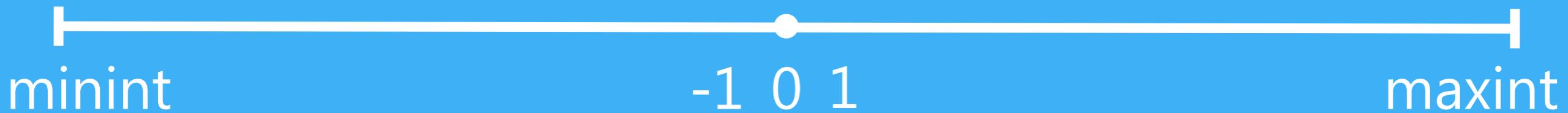
Is the problem with the previous code a code defect or a spec defect?

Depends. If the spec doesn't describe what the code should do, it's a spec issue. If the dev just didn't implement the spec it's a code issue.

Testing Tip

We're testing a method that takes one integer and does something to it. What values do we test it with?

Always start by thinking of the edge cases. For integers that's usually maxint, minint, 0, 1, -1.



Spot the Bug #2

```
def gcd(x, y):
    r = x % y
    if r == 0:
        return y
    else:
        return gcd(y, r)
```

What happens to memory usage as the number of iterations gets very large?

Performance Testing

Does it meet performance goals?

Is it responsive?

How much memory does it use?

How does it perform on a variety of hardware?

Is the problem with the previous code a perf issue?

Depends. What size numbers are we dealing with? How do we find the worst case? (Probably not a perf issue. `sys.maxint` is no problem.)

Spot the Bug #3

```
def receivemessage(message):  
    log.append("Message received @" + datetime.now())
```

What happens when messages are received every second for 3 days?

Stress Testing

Can it run for long enough?

How does memory trend over time?

Does it fail gracefully?

Is the problem with the previous code a stress issue?

Depends. Where does this code fit? How many messages are expected to be received? (Probably a stress issue. That code is awful.)

Spot the Bug #4

We're testing a method where the user must type their entire password within 2 seconds.

What about users with assistive typing devices?

Accessibility Testing

Can everyone use it?

Does it handle different types of input?

Does it work with screen readers and magnification?

Is the problem with the previous code an accessibility issue?

Yes. Many users won't be able to use your code if they have to be able to type at a certain speed.

Spot the Bug #5

print "What's your favorite color?"

This is Adam. He's Canadian. He doesn't know what a "color" is. To him it's a "colour."



Localization Testing

Is it usable in every locale we're targeting?

Are there any hard-coded English strings?

Are there UI elements that are locale-specific?

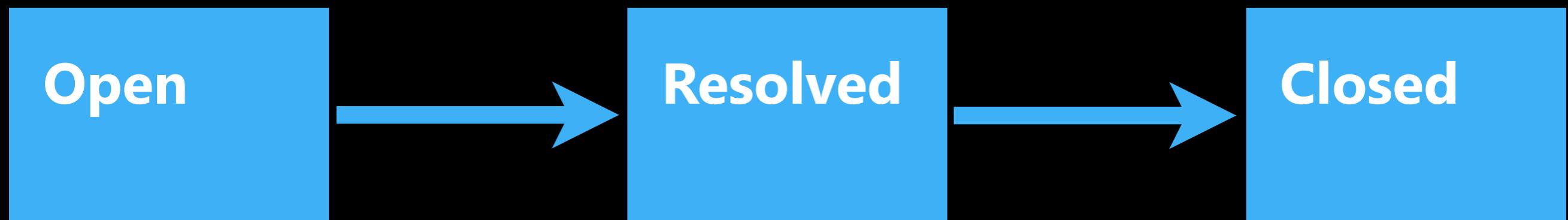
Is the problem with the previous code a localization issue?

Depends. Are we shipping to Canada? If so, yes (but even then it's a minor issue).

Testing

- **Functional**
 - Does it meet the spec?
- **Performance**
 - Is it fast enough?
- **Stress**
 - Can it run for long enough?
- **Accessibility**
 - Can everyone use it?
- **Localization/ Globalization**
 - Does it have assets for our target cultures?

Bug Life Cycle



Initial state.
Sub-states
such as Bug
Understood
and Fix
Available
track fix.

Some action
has been
taken. Bug
might be
Fixed, Won't
Fix, By
Design.

Tester has
agreed with
resolution
and added
automation
as possible.

Challenge

Your Job

Review the Spec and Design

Read it like a tester: what bugs would you log against the spec? What areas are you concerned about?

Write Automation

Start with your highest priority tests.

Ship It

Gather metrics and give your final green light (or raise a red flag).

Your Deadline

15 minutes from now

Spec: Connect Four

Overview: a two-player text-based game of Connect Four.

Use Case: Playing a Game

Adam and Beatrice want to play a game of connect four. They start the program and are given a quick overview of the rules.

They then take turns choosing a column to play into using text input. If they want to end the game before someone wins or the board is filled they use the quit option.

Spec: Connect Four

Use Case: Undo/Redo

Adam makes a mistake (like usual) and wants to take back his move. He uses the undo option to take back his move. It's now his turn again, as if his last move never happened.

If Adam decides he really did want that move after all he can redo it instead of entering it again.

The number of undos and redos is unlimited.

Spec Issues

- **Rules**

- How do you play Connect Four?
- What happens when you win?
- Ties?

- **Failure conditions**

- Invalid Input
- Undo before first move
- Redo before undo

- **Details**

- How do the users pick a column?

- **Goals**

- Is there a goal for how fast a move is applied?
- How long should this run for?

Design

ConnectFourBoard

- isgameover()
- winner()
- play(column)
- undo()
- redo()

ConnectFour

- gameloop()
- reset()



Automation

ConnectFourBoard

- `isgameover()`
- `winner()`
- `play(column)`
- `undo()`
- `redo()`

Code Issues: Board

- `winner()` throws exception if called before someone wins
- Filling entire first row ends game in a tie
- Undo after winning move doesn't undo win

Automation

ConnectFour

- gameloop()
- reset()

Code Issues: Game

- playing more games than
sys.getrecursionlimit()
results in an exception
- inputting an invalid column
to play in prints that many
prime numbers starting
with 2

When are we finished?

Exit Criteria

Bug Trends

Are we still
finding big
issues?

Zero Resolved
Bugs

Zero Bugs

Code Coverage

Use a tool to
measure the
amount of
product code
executed by
tests

Planned Testing

Did we cover
every case
identified during
planning?

Code Coverage

self._isgameover = False

self._winner = None

Code Coverage

Before you added any tests for ConnectFourBoard the `test_one_move` method gave you **60%** line coverage of `connectfourboard.py`

Before you added any tests for ConnectFour the `test_startup` method gave you **27%** line coverage of `connectfour.py`

Testing Tricks

Equivalence Classes

We're testing a method that takes a string of 1s and 0s as input. What strings do we test with?

Which strings are functionally equivalent (will travel the same code paths)?

Possible answers:
Empty string
String of all 1s
String of all 0s
Alternating

Testing Frameworks

Python

unittest, doctest

C

Google Test Framework,
Check, Minunit

Java

Junit

C++

Google C++ Test
Framework, UnitTest++

Perl

TAP, Test::Class

[en.wikipedia.org/wiki/
List_of_unit_testing_frameworks](http://en.wikipedia.org/wiki/List_of_unit_testing_frameworks)

Test Driven

- 1. You are not allowed to write and product code unless it is to make a failing unit test pass.**
- 2. You are not allowed to write any more of a unit test than is sufficient to fail (compilation failures are failures).**
- 3. You are not allowed to write any more product code than is sufficient to pass the one failing unit test.**

```
def setUp(self):  
    self.dice = Dice()
```

```
class Dice:  
    def __init__(self):  
        pass
```

```
def setUp(self):  
    self.dice = Dice()
```

```
class Dice:  
    def __init__(self):  
        pass
```

```
def setUp(self):  
    self.dice = Dice()  
  
def test_d6(self):  
    self.dice.roll("d6")
```

```
class Dice:  
    def __init__(self):  
        self.random =  
            random()  
  
    def roll(self,  
            description):  
        if(description ==  
            "d6"):  
            return  
                self.random.randint(1,6)
```

```
def setUp(self):  
    self.dice = Dice()  
  
def test_d6(self):  
    result =  
        self.dice.roll("d6")
```

Summary

Testing

**Functional
Performance
Stress
Accessibility
Localization**

School

Start testing now. It'll build useful skills and improve your grades.

All the techniques we use in product development, like testing and version control, exist because they help you make better code faster.

Tips:

- Learn a unit test framework for your favorite language
- Learn Git
- Pick up some languages: python, C#
- Challenge yourself

Working

Think about a working as a tester. It's all the fun of being a dev, but with very different challenges.

When you interview:

- Think out loud
- Make mistakes
- Have fun
- Be ready to test all kinds of stuff (a pencil, a soda can)
- Own your algorithms
- Be ready to sort

Contact

salatorr@microsoft.com

I'm happy to give feedback on resumes and submit names for open positions.

@olde_fortran
latorra.org
github.com/Sagelt

I can't guarantee I say anything smart or CS-related on a regular basis on my site or twitter. Mostly I just nerd out about stuff.