

# Why You Should Use the Crappiest Computer Possible

Michael Gat



# A realization: Programming quality sucks



**Katherine Scott**

@kscottz



Replying to @kscottz

We're computer scientists FIRST. Compute time and storage add up to real money; like millions of dollars. That's solving a real world problem that people will gladly pay for. 0.1% is useful but these problems have multivariate solution spaces.

2:52 AM - Feb 17, 2018 · San Francisco, CA



3



See Katherine Scott's other Tweets



# And again...



**Katherine Scott**

@kscottz



I swear, am I the last generation of CS students who learned how to write object oriented code? What are they teaching the kids these days? I take it boot camps don't teach much more than "how to for loop".

6:21 PM - Jun 19, 2018



6



See Katherine Scott's other Tweets



# So I began thinking...

- Really this is nothing new
- Pushing crappy code was pretty much the “WinTel” business model for over 20 years.
- Intel execs were almost proud that they prodded software companies to write bloated, awful code
- So why would anybody learn to do anything else?

# You're supposed to learn in school, right?

- Sort of.
- CS Professors are as captive to big software companies as business school “professors” are to Wall Street.
- Schools increasingly have access to the “latest and greatest,” so nobody has to struggle with limits.
- Students aren't stupid. They know what they need to know to get a job.
- Which may not be what they need to know *on the job!*

# Result: Bad programmers with A+ degrees

- In all likelihood, they were taught all the things they “should” know.
- But in the absence of any need to use them, none of it stuck past the final project/exam.
- Depending on where they work, they might never know how much their bad code is costing the world.
- At this point, the managers don’t know either because nobody’s really been forced to learn this in years.

# Just to be clear:

Because it's important

I'm not just an old guy  
telling you kids to get off  
my lawn.

Really I'm not.

---

@michaelgat

# There's a looming problem

- Moore's Law has been slowing down.
  - Intel will admit it
  - Even nVidia will admit it
  - Exponential advance in hardware has been the backdrop for our tech world for so long that we take it for granted and we should not!
- We cannot count on hardware advances to save us!
- In the absence of a revolutionary new tech, we need to be smarter about how we write software.
  - Don't count on quantum computing just yet.
- Being lazy is no longer an option.



# What can be done?

- The increasing use of “cloud” services like AWS make this more important.
  - You pay by the KB/MB/cpu use
  - Maybe people will start noticing the cost of bad code.
- So do advanced apps on mobile devices
- But... students get free accounts on AWS and elsewhere, so aren't forced to feel the pain.
- We need another idea.

# A modest proposal: crappy computers

- Take us back to those thrilling days of yesteryear, when we had to worry about such things as
  - How much memory we used
  - Whether we released it when done
  - How much CPU was needed?
  - How much space on disk? (Ha!)
  - Could we do it all without overheating?
- Modern computers make us forget all these issues.
  - “Benchmarking” is a poor substitute for the pain of doing it poorly

# How crappy? Not this crappy.



How crappy? Not this crappy either!



# “Crappy” is a state of mind

- It should be limited enough to force you to think about what you’re doing and why.
- It should punish you for doing things inefficiently
- It should force you to interact with the machine as much as practical, given modern technology
  - But no, you shouldn’t need to be an EE
- **We have such devices!**
- And people love them! (CS professors, not so much)

# Yes, *those* devices!



**SwiftOnSecurity**  
@SwiftOnSecurity



The [@Raspberry\\_Pi](#) 3B is the 67th most popular electronic item.

On Amazon. All of Amazon.

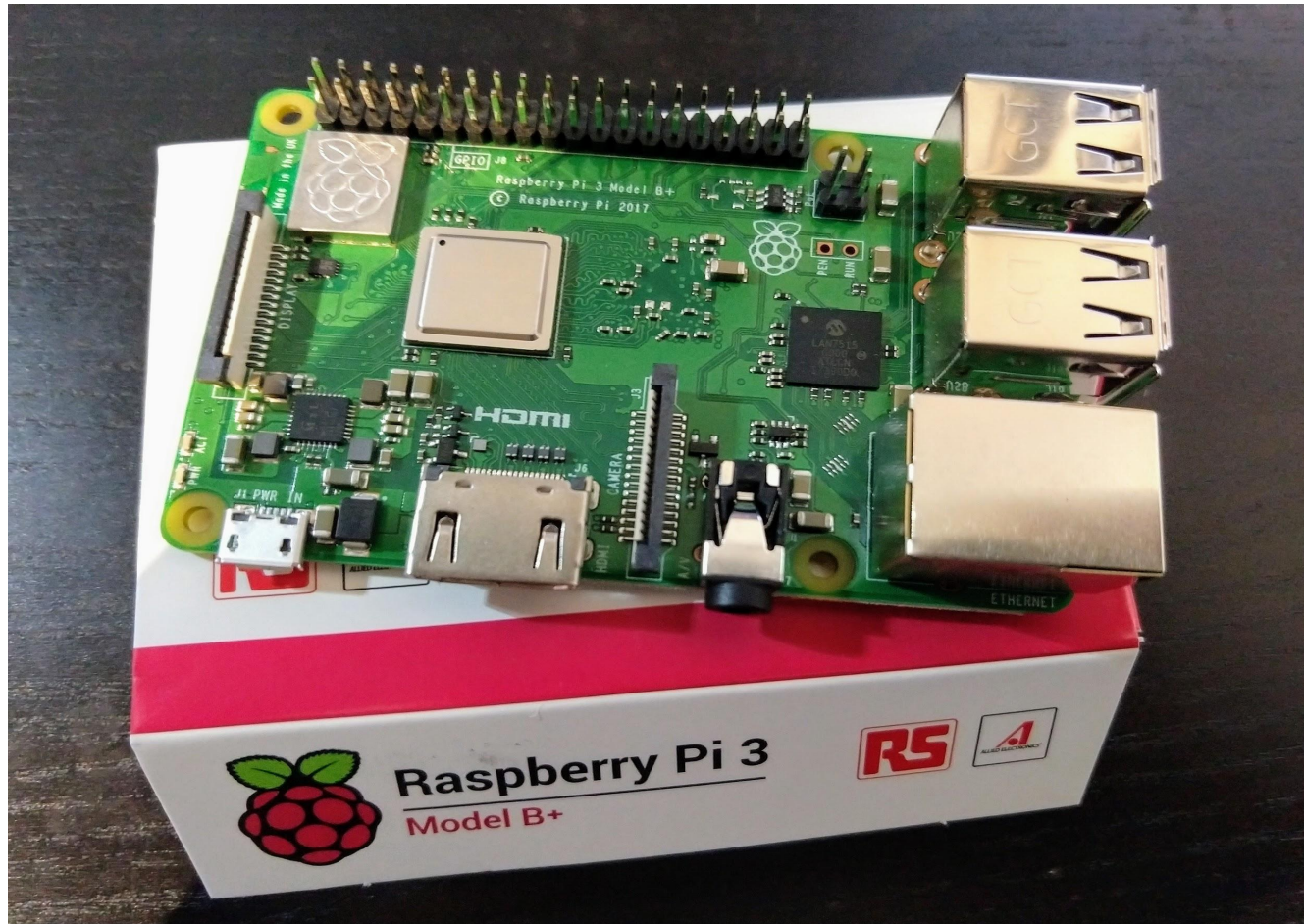
3:58 PM - Apr 17, 2018

♡ 242 💬 47 people are talking about this



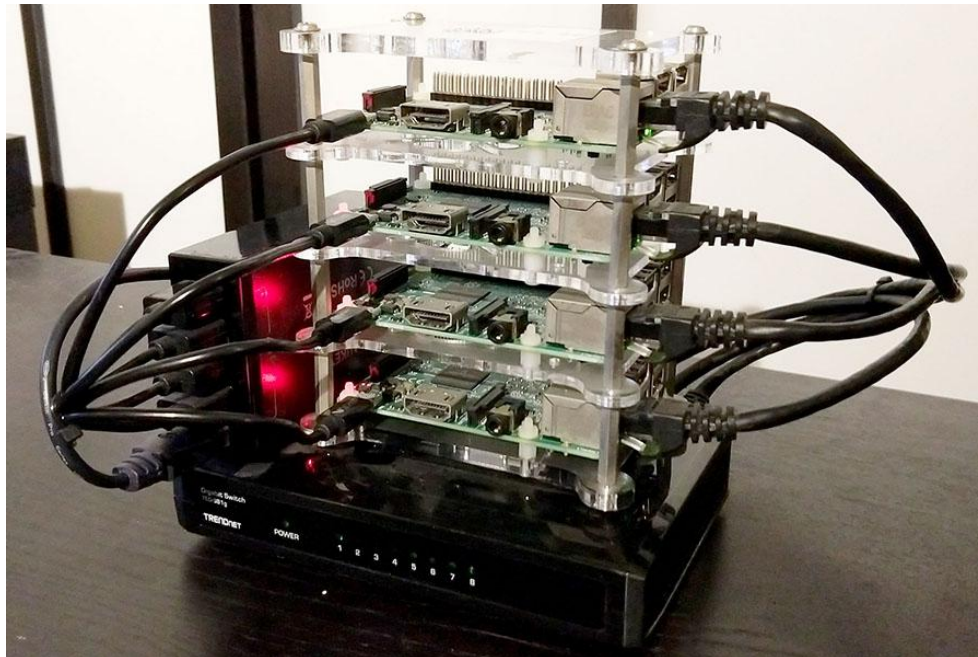


Yes, those!



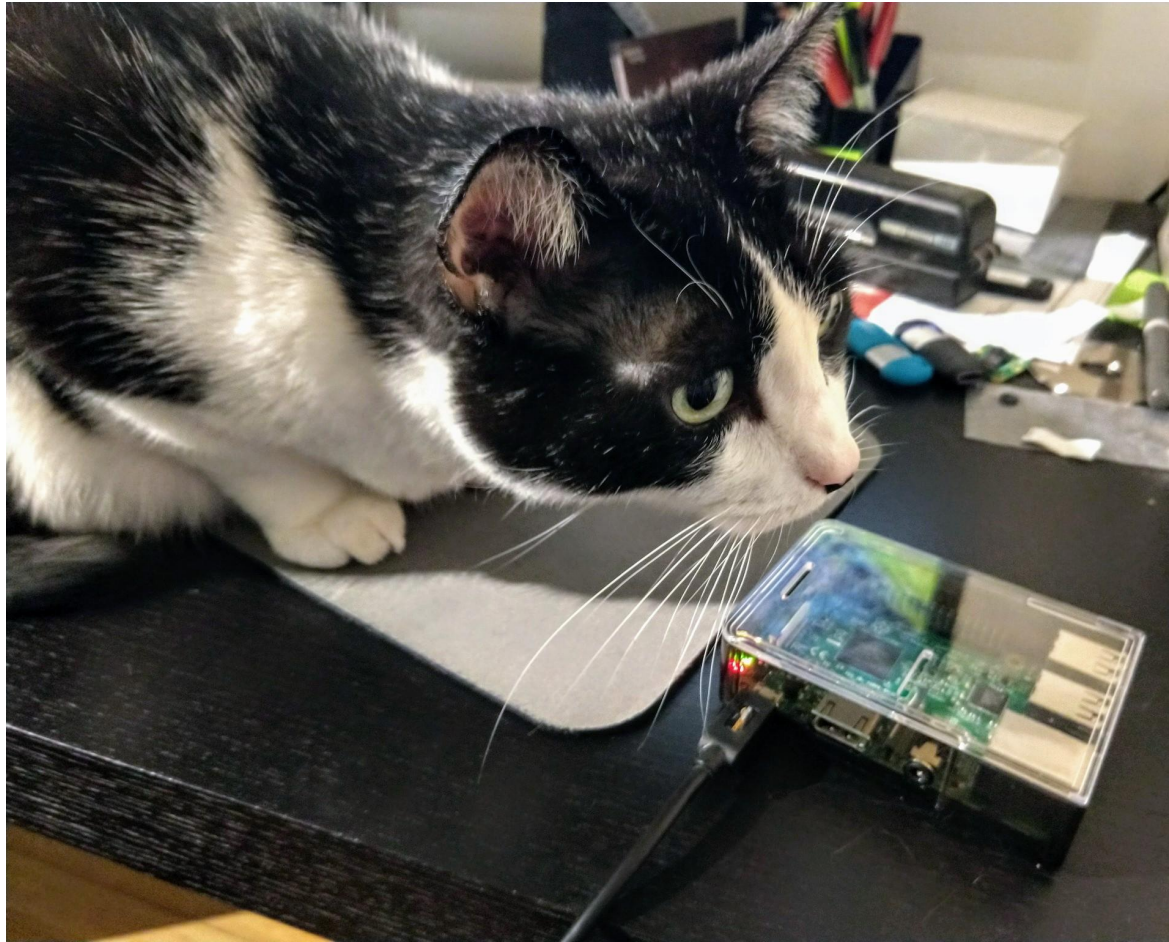
# The Raspberry Pi was made for this

- Originally targeted at primary education, it's become a very versatile and capable platform.
- The latest version is Wifi-enabled, handles video (barely) and graphics, and behaves well in a cluster.





# Obligatory Cat Photo



Want to really push yourself? Try the latest.





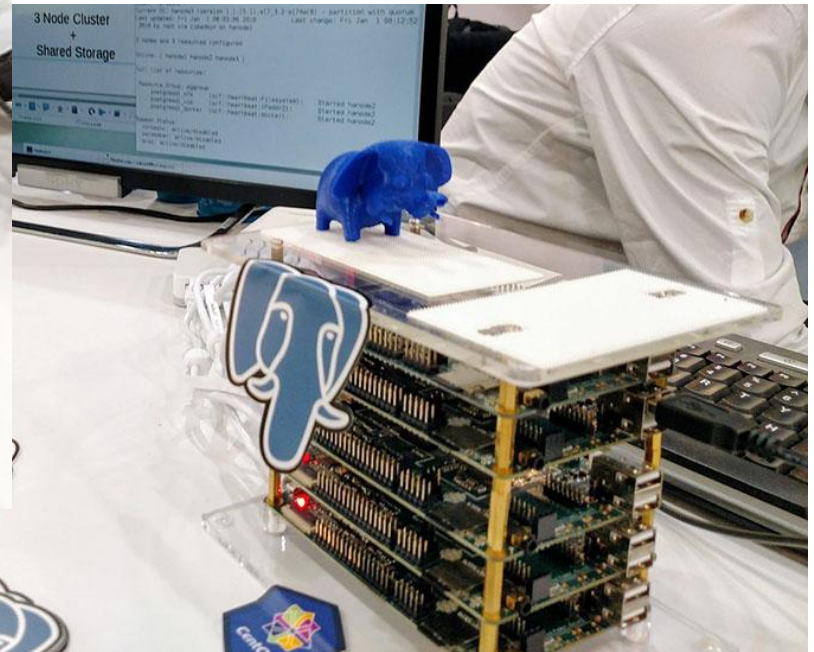
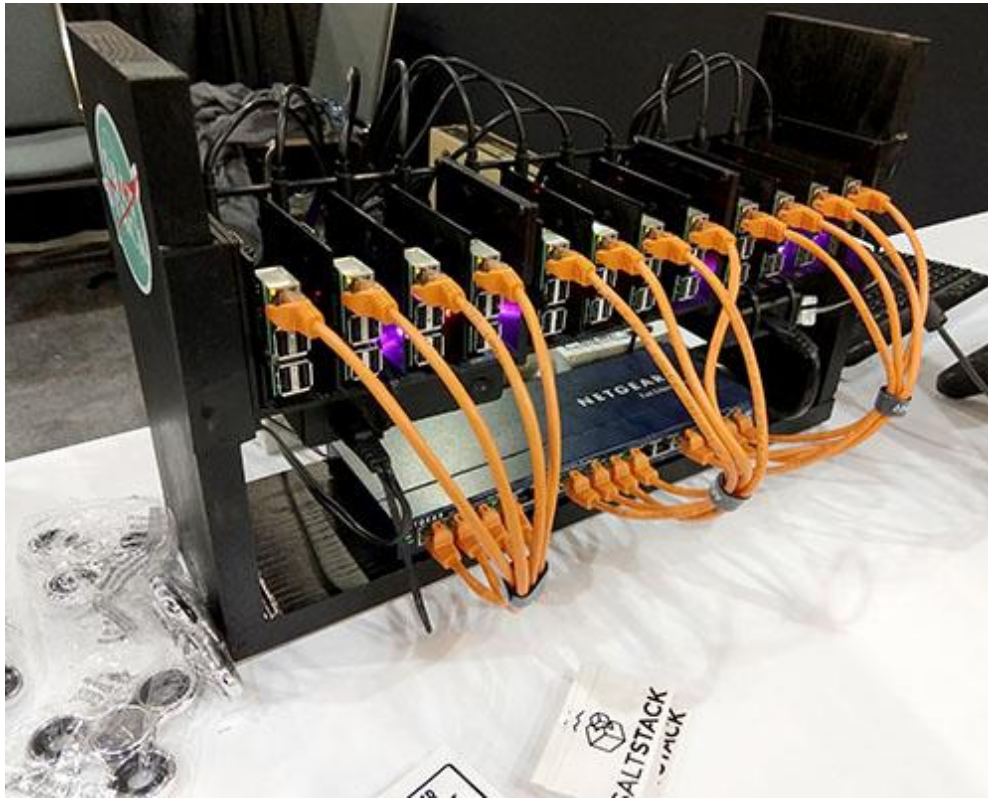
# But also

- Are you going to attach your \$2000 macbook to this and run it all over town?



- What else might you do if the cost is \$35? Or \$10?
  - Launch it on a rocket? Sink it in the ocean?
- The possibilities are pretty amazing.

# Very smart people are already going there



# There are other great options too:

- Pine64
- BeagleBoard
- NanoPC/NanoPi
- Arduino and compatibles (robotics)
- adafruit Circuit Playground (hardware hacking)
- *That old crappy laptop in the back of the closet that won't run any Windows past XP.*
  - *It'll probably run Ubuntu just fine!*
- Lots of different options and pricepoints

# In summary

- We cannot count on Moore's Law to save us from our sloppy practices.
- To save us, we need a new era of simple "crappy" computing in higher education and training.
- Nobody needs more than a Raspberry Pi or similar until they're through their mid-level courses.
  - They'll hate it like I hated "3B2 hell", but they'll be better, as I was.
  - A lot of people don't even need more for their real jobs.
- We need to go back to basics and relearn old lessons.

# Thank You!



Michael Gat  
@michaelgat  
michaelgat.com

Slides at [gitlab.com/michaelgat/Presentations](https://gitlab.com/michaelgat/Presentations)

## Questions!

Not:

- Calling bullshit
- A long story
- Your resume

---

@michaelgat