

# ECOES: Software Engineering

Gregg Vesonder

# Roadmap

- Intro – what is software engineering?
- Ideation and requirements
- Social media exercise
- Citizen science
- A bit on smart cities
- Experiments for home
  - Lab 1 blinkie
    - -raspberry pi
  - Lab 2 Help!
  - LAB 3 Roll them
  - Lab 4 accessibility
  - Lab 5 and on - on your own!

# Gregg Vesonder about me

Stevens for 18 years

University of Pennsylvania for 10 years

at&t/bell labs for 35 years

Software Engineering,  
Artificial Intelligence,  
Human Computer Interaction,  
Software Design and Architecture

# my first dog, my first computer





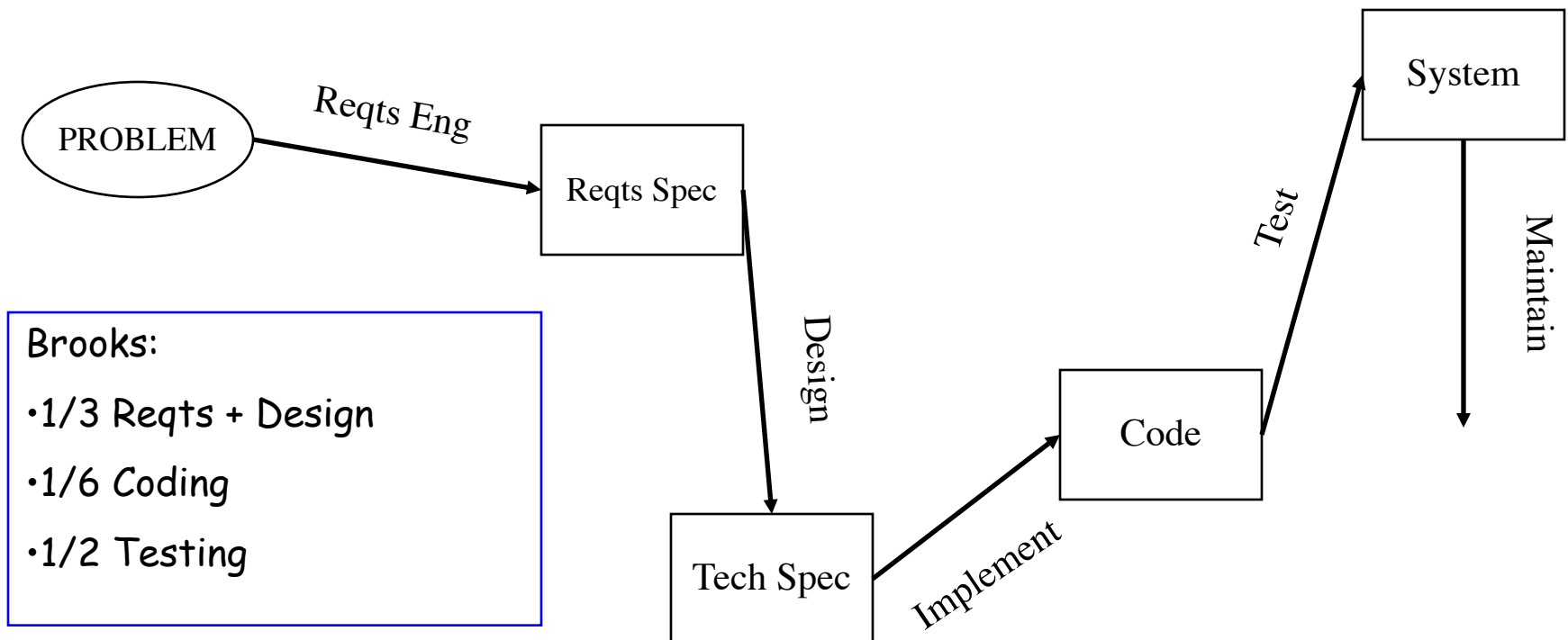
# What is a computer?

What is Computer Science?  
What is Computer Engineering?  
What is Software Engineering?

And even if you are interested in engineering ...

What are the steps to building  
a software product?

# Simplified Model



How frequently do you use social media apps every day?

What social media apps do you use for communication?

# IDEATION

Design a social media app

# Brain storming



# Requirements generation

# Citizen Science

# Citizen Science

- Wikipedia: **Citizen science**(**CS**; also known as **community science**, **crowd science**, **crowd-sourced science**, **civic science**, **volunteer monitoring**, or **networked science**) is scientific research conducted, in whole or in part, by amateur(or nonprofessional) scientists. Citizen science is sometimes described as "public participation in scientific research," participatory monitoring, and participatory action research.

# Japan's Tsunami



[blog.salvationarmyusa.org](http://blog.salvationarmyusa.org)

# Fukushima



PHOTO ISSEI KATO/AFP/GETTY IMAGES

3 of 6 nuclear reactors melted down

[money.cnn.com](http://money.cnn.com)



# Radiation Spread

- Government reported on the spread of radiation
- A 20 kilometer exclusion zone was established
- Citizens were skeptical about the government reports
- Radioactive water was leaking into the ocean



wikipedia

# Personal Radiation Detectors

- Citizens needed another way to assess the threat
- Personal radiation detectors emerged built on microprocessor technology
- iPhone Safecast app
- Data placed in repository
- APIs to access data
- Soon citizens were posting their own data

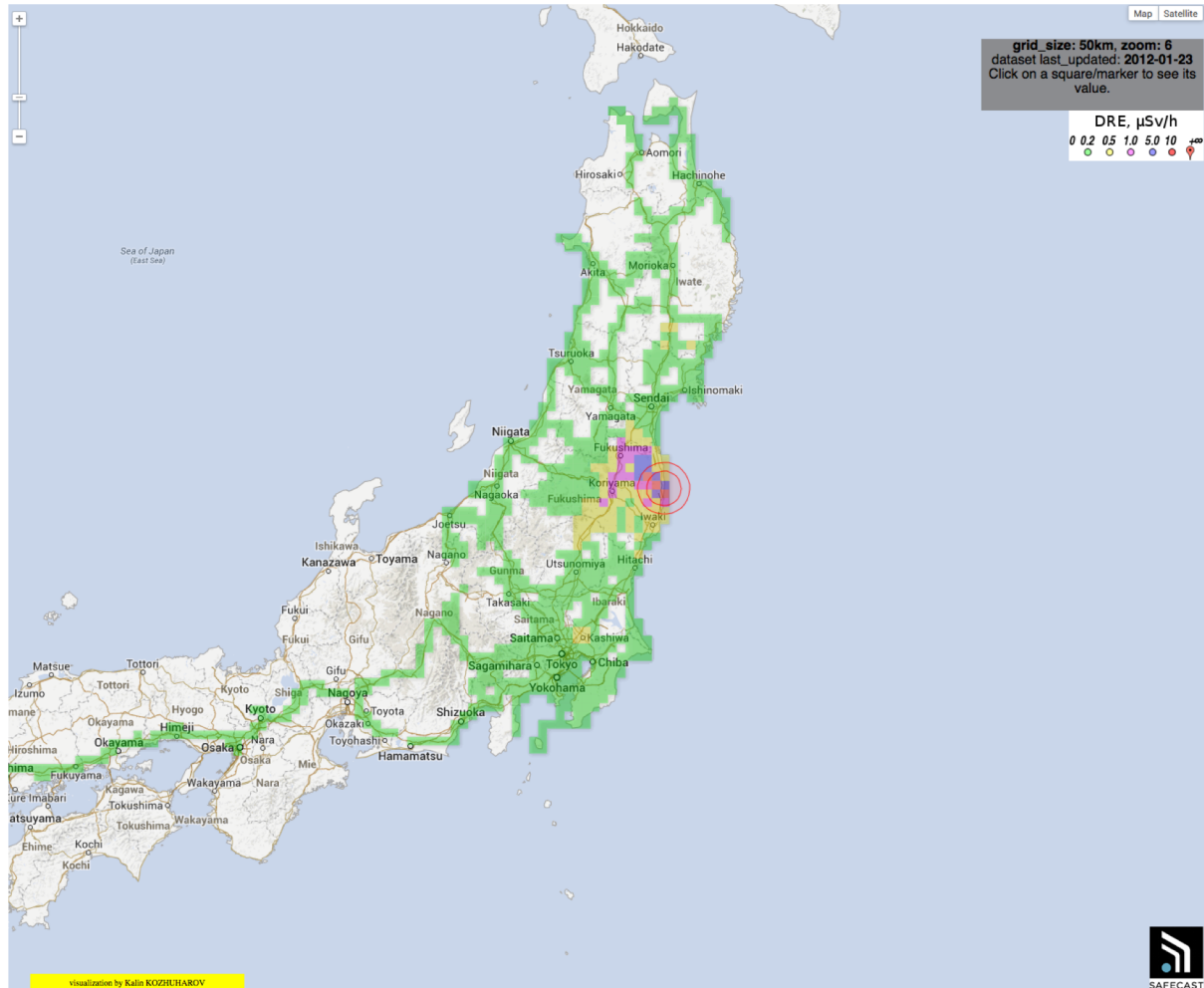


[amazon.com](https://www.amazon.com)



[Medcom.com](https://www.Medcom.com)

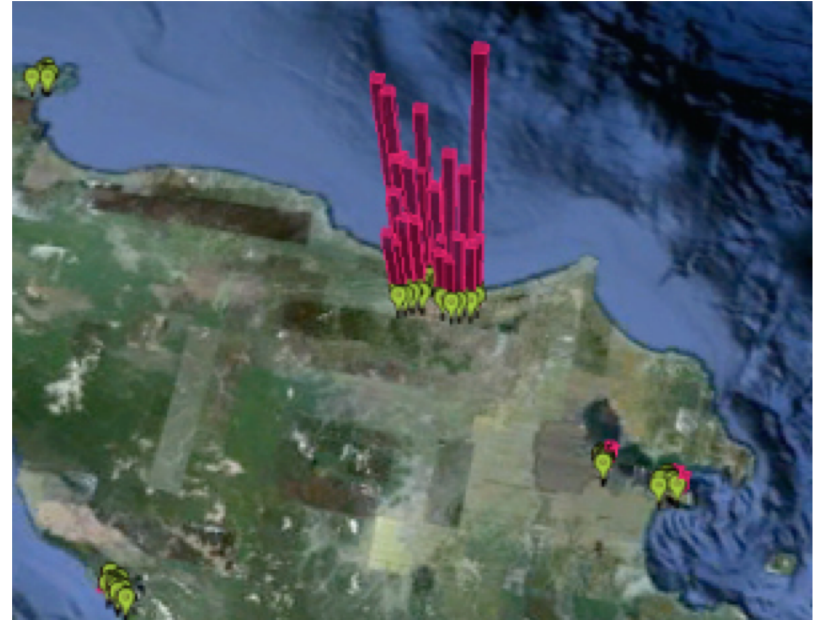
# SafeCast Map





# Other Maps

- Large number of readings invaluable
- Compared against government produced data
- Compared against their data – calibration is an issue
- Real-time data
- Multiple visualizations



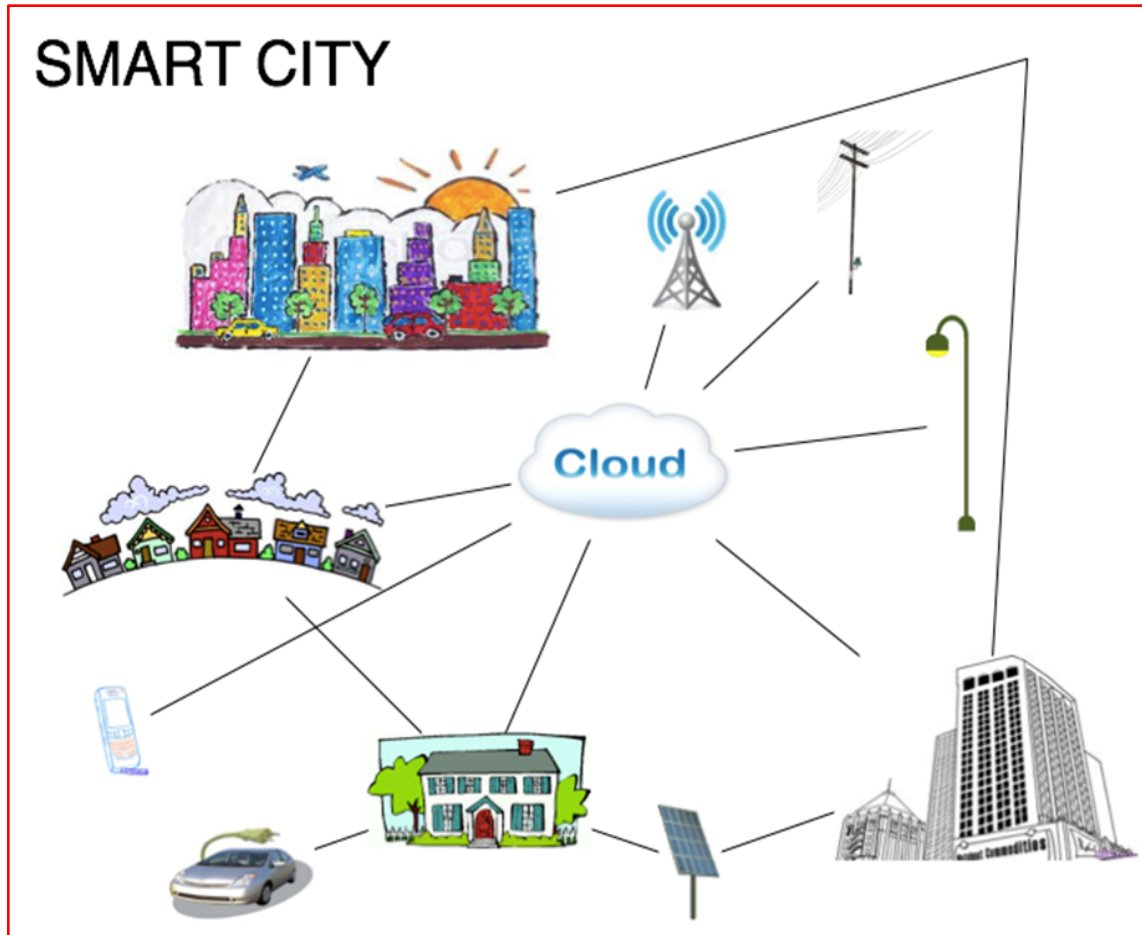
# How it works

- People or machines generate data
  - PAN
- Data is transferred to a collecting point
- Data is ordered in some way
  - Database
- Code is written to access the data -> APIs
- Developers use APIs to access data

# IoT

- Tagging Things
- Sensing Things
- Shrinking Things
- Thinking Things

# Leads to Smart Cities



# Cities serving Citizens

- What do you think of when you think of cities?

# Issues

- Power
- Communication
- Calibration
- Security
- Analysis
  - Big Data

# Citizen Science

- STEM + Citizen Science
- Environment Egg
- What if we made it a STEM project?



<http://shop.wickeddevice.com/product/air-quality-egg-v2-no2-co/>



[http://momhomeguide.com/wp-content/uploads/2012/03/before\\_fridge.jpg](http://momhomeguide.com/wp-content/uploads/2012/03/before_fridge.jpg)

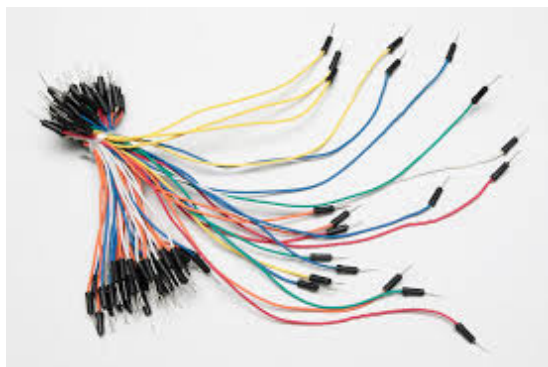
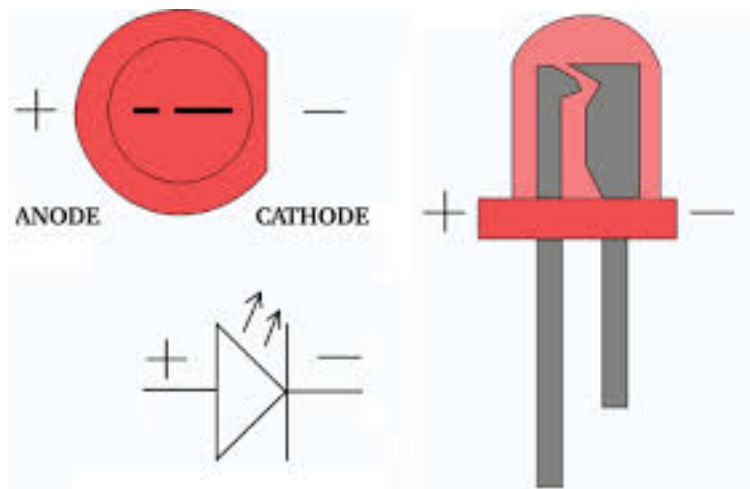
# About the LAB



# Morse Code

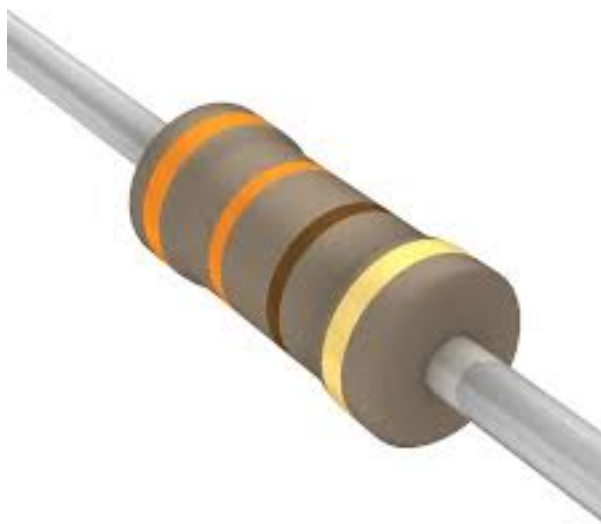
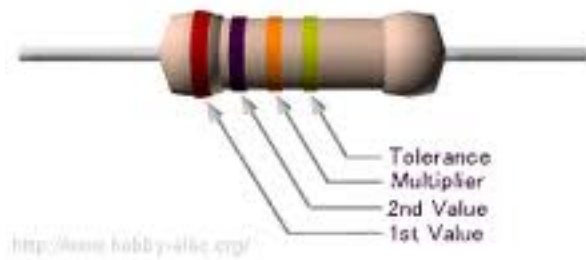
- Long history
- <http://www.youtube.com/watch?v=BgelmcOdS38>

# Hardware components

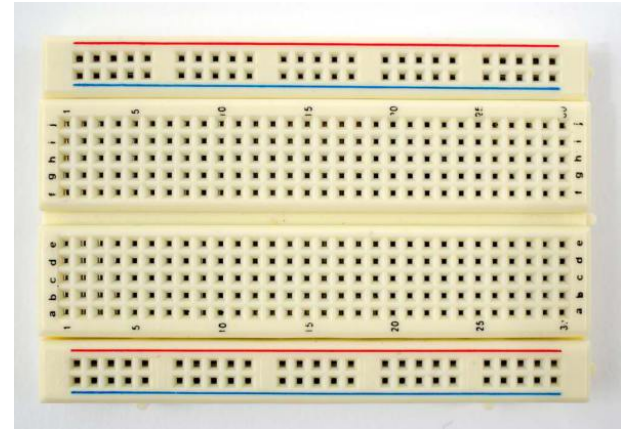
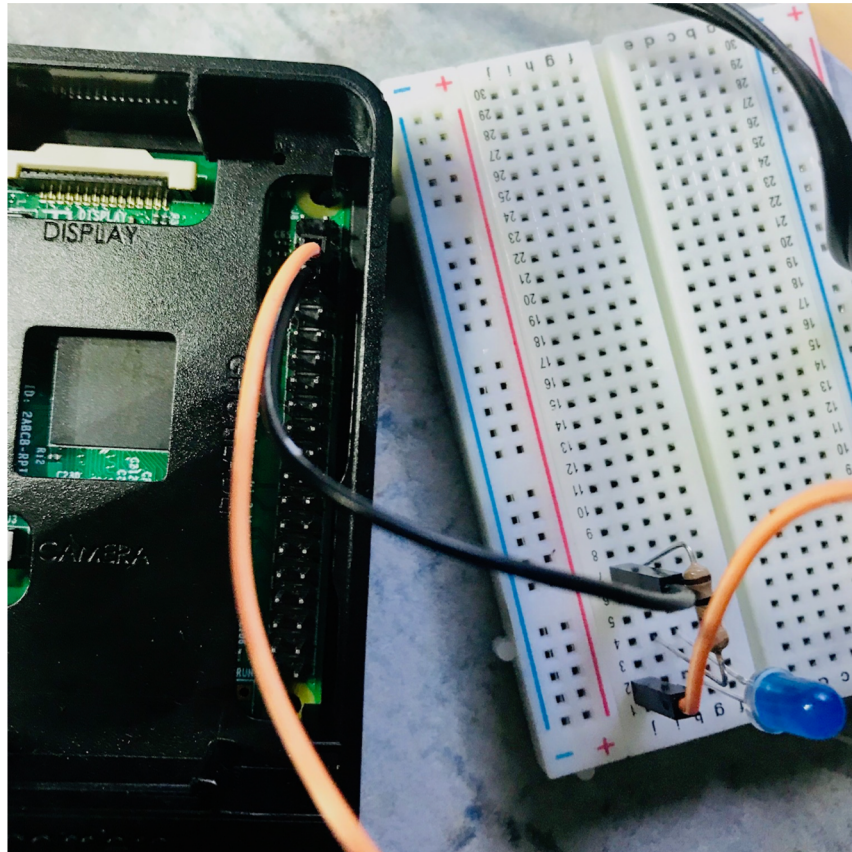


+resistors

# Resistors



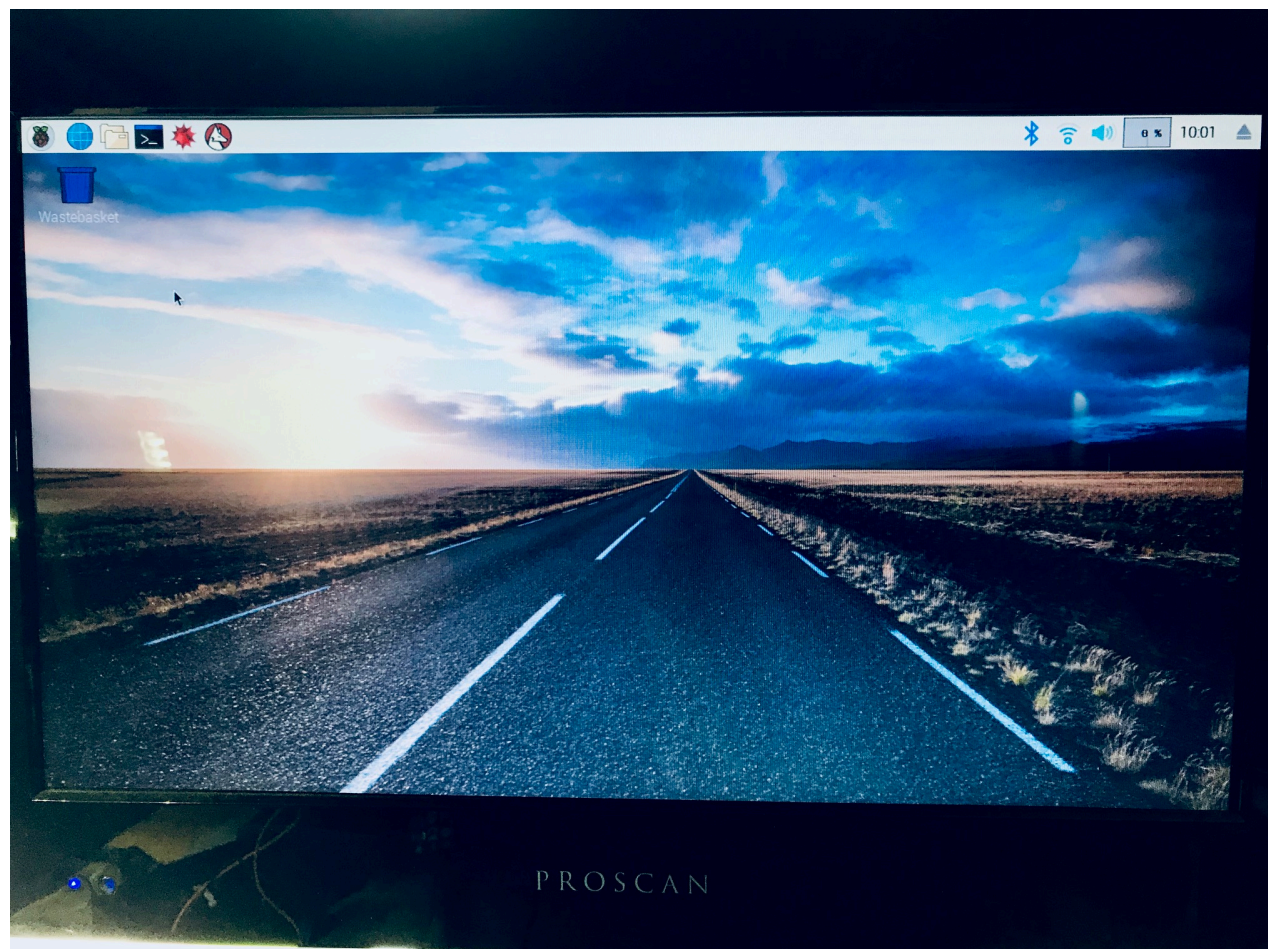
# Hardware Platform





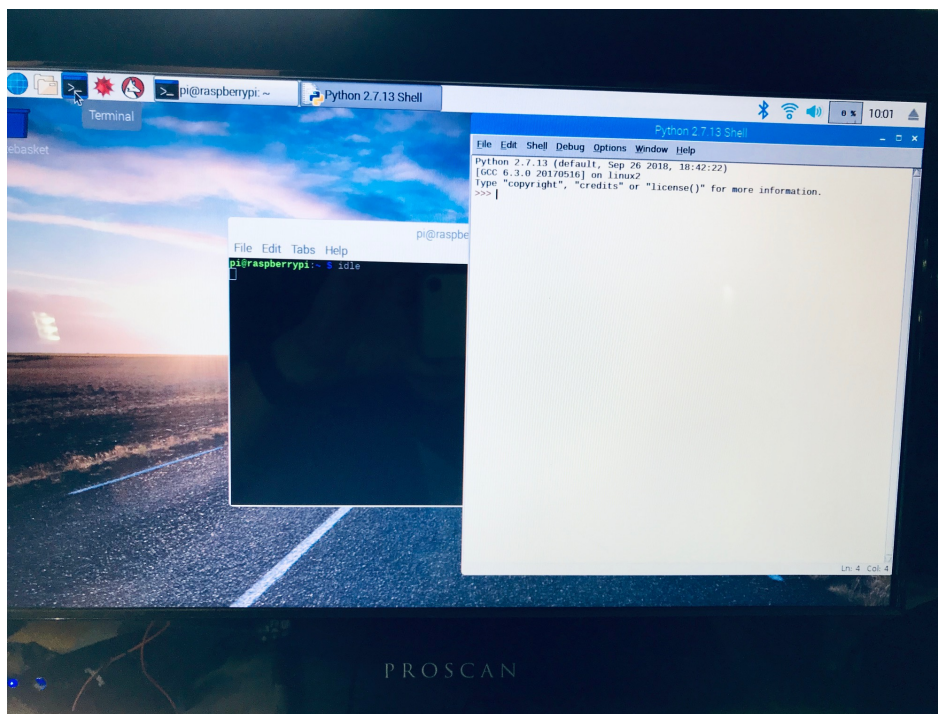
# Software Platform - 1

- Linux
- X Windows



# Software Platform - 2

- python
- idle  
(IDE)



# Code

- Variables
- Iterators:
  - while (condition):
  - for <var> in :
- def myFunctionName (myArgs):
- If (condition) :
- True and False
- # is comment indicator

# Variables

- Represents a value
- Value can be a string, number(integer, real), boolean (True, False), *pointer*, and more
- Variable names should contribute to understanding
- There is a lot going on behind the scene
  - ASCII for instance
- Examples of variable statements:
  - `my_name = Gregg`
  - `my_age = 63`



# Fun With Variables

- `name = "Alice"`
- `name[0]` in `"AEIOU"`
- `name[3]`
- `name = "Gregg Vesonder"`

# import

- import is used to add code and therefore capability to the python interpreter
- Two collections of code known as modules in python are:
  - random
  - `import RPi.GPIO as GPIO`
- Note python is case sensitive

# loops

- The `for` loop used to do tasks a fixed number of times or to iterate (walk through) a list.
- The `while` loop continues so long as a condition is true
  - `break` can be used to escape from a while loop
- These two loops are very powerful
- Indentation delineates the body of the loop

# Fun with Loops

```
dogs = ["spaniel", "collie", "pit  
bull"]
```

```
for dog in dogs:  
    print(dog)
```

```
for j in range(0,4):  
    print(j)
```

```
range(0,4)
```

# LAB 2

# Functions

- Critical part – a way of reusing code
- `def name_of_function():`

```
def greeting():
```

```
    print("hello")
```

```
def greeting(name):
```

```
    print("hello " + name)
```

# Lists

- Lists are defined using square brackets
- They are addressable and have many uses

```
prices[1.50, 2.75, 56.82]
```

```
def add_it(numbers):  
    total= 0  
    for number in numbers:  
        total = total + number  
    return total
```

```
len(prices)
```

```
prices.append(3.39)
```

```
1.50 in prices #containment
```

# LAB 3



# import random

```
import random
random.randint(1,6) # lower and
upper bound
cards = ["ace", "king", "queen",
"jack"]
random.choice(cards)
random.shuffle(cards)
#lots of modules
```

# Dice

```
>>> import random
>>> random.randint(1,10)
10
>>> def dice(size):
...     return(random.randint(1,size))
... #4 spaces!
>>> dice(10)
5
>>> dice(20)
16
>>>
```

## LAB 4

# LAB 5 & 6

# Files

```
#open(file_name, mode - read,  
write, append)  
data_file = open("my_data.dat",  
"r")\  
for line in data_file:  
    print(line)  
f.close()  
attendance_file = open("SaS.txt",  
"w")
```

# Dictionaries

#provides a key value relationship e.g., name-age

```
person_age = { "gregg" : 63, "alice" : 28 }
```

```
person_age["gregg"]
```

#update

```
person_age["gregg"] = 64
```

#keys have to be unique!

# Classes

#important a way of  
representing common  
things

#convention to  
capitalize name of  
class

Class

```
Greeter(object):  
    def hello(self):
```

```
print("hello")
```

```
    def  
goodbye(self)  
        print("good  
bye")
```

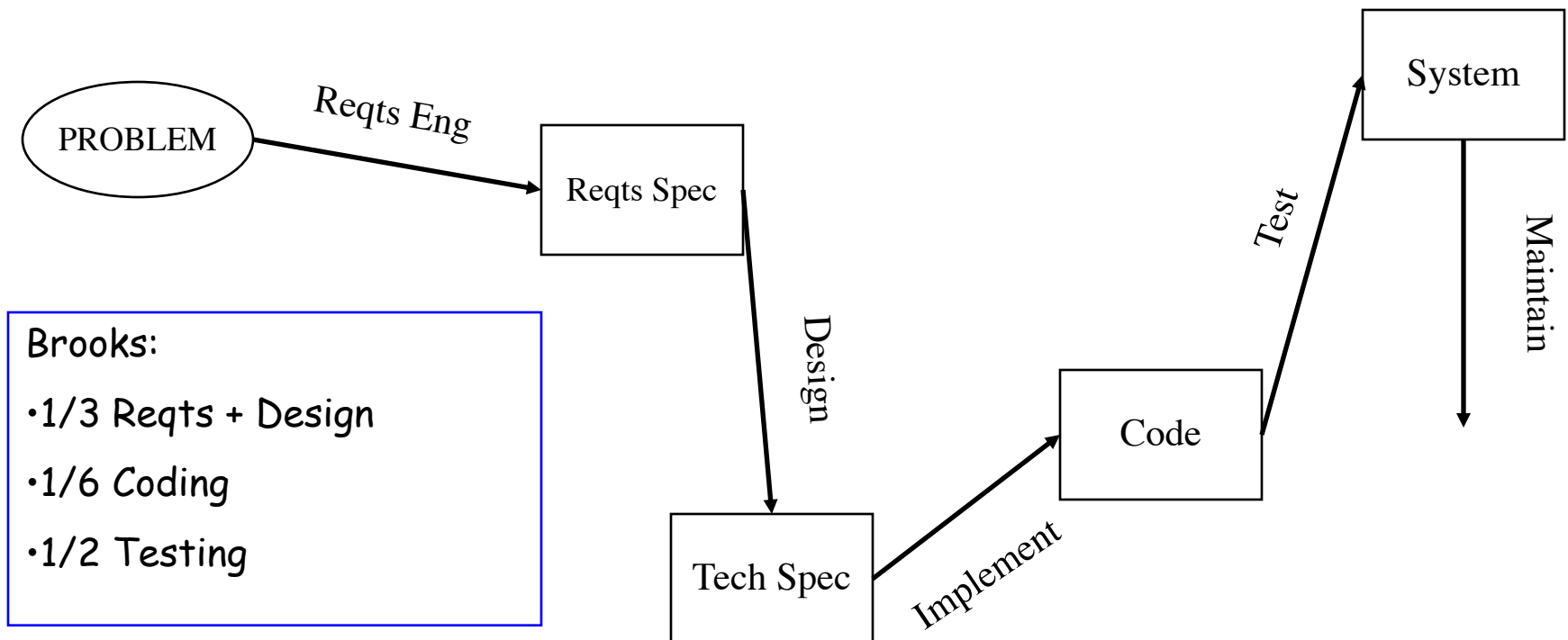
```
g = Greeter()  
#instantiate an  
object  
g.hello()  
g.goodbye()  
g2 = Greeter()
```

# Classier

```
Class Greeter2(object):  
    def __init__(self, name)  
  
    def hello(self):  
        print("hello")  
    def goodbye(self):  
        print("good bye")
```



# Simplified Model



# Ongoing

- Jessica McKellar videos youtube
- Learn computer languages
- Program
- Learn from others
  - Stack overflow
- Build
- Be curious
- Be creative
- Find your Joy
- vesonder.com
- [vesonder@mac.com](mailto:vesonder@mac.com)