WHY IS DATA ENGINEERING SO HARD?

A LOOK AT THE CAUSE, SOLUTION AND FUTURE OF DATA ENGINEERING DIFFICULTIES

ABOUT ME

Nearly 15 years experience dealing with data

Real-Time Streaming

Predictive Analytics

Sports Betting @ PhoenixHSL

Financial Services @ Westpac / Accenture

David Tout

* Philosophy graduate
* Strong opinions on data engineering
* Wants you young kids off his lawn…

Data Engineering as a People Problem

1. Problem:
   We make it harder than it needs to be
2. Solution:
   We need navigators not heroes
3. Result:
   Self-service data consumption is our goal

WE ARE THE PROBLEM!

Engineers are problem focused	Need to be more people focused
Prefer our own ideas (IKEA Effect)	Re-invent rather than re-use and extend
Tools and technologies proliferate, littering the landscape	
DevOps and Data Science don't speak a common language	
Different goals & Different skillsets	

CSV TO PARQUET CONVERSION METHODS

PANDAS & PYARROW

```python
# csv_to_parquet.py

import pandas as pd
import pyarrow as pa
import pyarrow.parquet as pq

csv_file = '/path/to/my.tsv'
parquet_file = '/path/to/my.parquet'
chunksize = 100_000

csv_stream = pd.read_csv(csv_file, sep='\t', chunksize=chunksize, low_memory=False)

for i, chunk in enumerate(csv_stream):
    print("Chunk", i)
    if i == 0:
        # Guess the schema of the CSV file from the first chunk
        parquet_schema = pa.Table.from_pandas(df=chunk).schema
        # Open a Parquet file for writing
        parquet_writer = pq.ParquetWriter(parquet_file, parquet_schema, compression='snappy')
    # Write CSV chunk to the parquet file
    table = pa.Table.from_pandas(chunk, schema=parquet_schema)
    parquet_writer.write_table(table)

parquet_writer.close()
```

SPARK 2.0

```python
from pyspark import SparkConf
from pyspark import SparkContext
from pyspark.sql import SQLContext

conf = SparkConf().setMaster("spark://bigdata-server:7077")
sc = SparkContext(conf=conf, appName="flightDataAnalysis")
sqlContext = SQLContext(sc)

#converts a line into tuple
def airlineTuple(line):
    values = line.split(",")

    return (
    values[0], values[1], values[2], values[3], values[4], values[5
    values[10], values[11], values[12], values[13], values[14], val
    values[20], values[21], values[22], values[23], values[24], val

#load the airline data and covert into an RDD of tuples
lines = sc.textFile("hdfs://localhost:9000/user/bigdata/airline/inp

#convert the rdd into a dataframe
df = sqlContext.createDataFrame(lines, ['Year', 'Month', 'DayofMont
                                        'CRSArrTime', 'UniqueCarrie
                                        'CRSElapsedTime', 'AirTime'
                                        'Distance', 'TaxiIn', 'Taxi
                                        'CarrierDelay', 'WeatherDel
                                        'LateAircraftDelay'])

#save the dataframe as a parquet file in HDFS
df.write.parquet("hdfs://localhost:9000/user/bigdata/airline/input
```

THE NEED FOR NAVIGATORS NOT HEROES

THE WORLD *WILD* WEB

Web design used to be difficult

Fancy tools like wix, or squarespace make many web designer roles redundant.

Web roles still exist, but for much more involved sites

48% chance of automation by AI for Computer Programmers, according to a 2013 study

However, **Database Administrators** scored only 3%

Data Engineers are probably at **3-8%** risk

These figures will grow, year on year, and eventually accelerate.

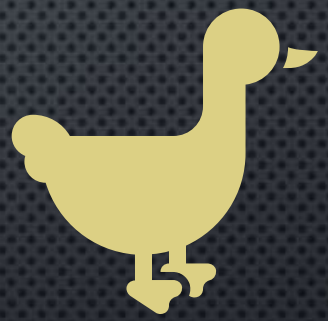
If your job is the 2019 equivalent of Web Design, you stand a good chance to lose your job in the next two decades.

AI IS COMING FOR YOU (R JOB)

willrobotstakemyjob.com

SELF SERVICE PLATFORMS & ENVIRONMENTS

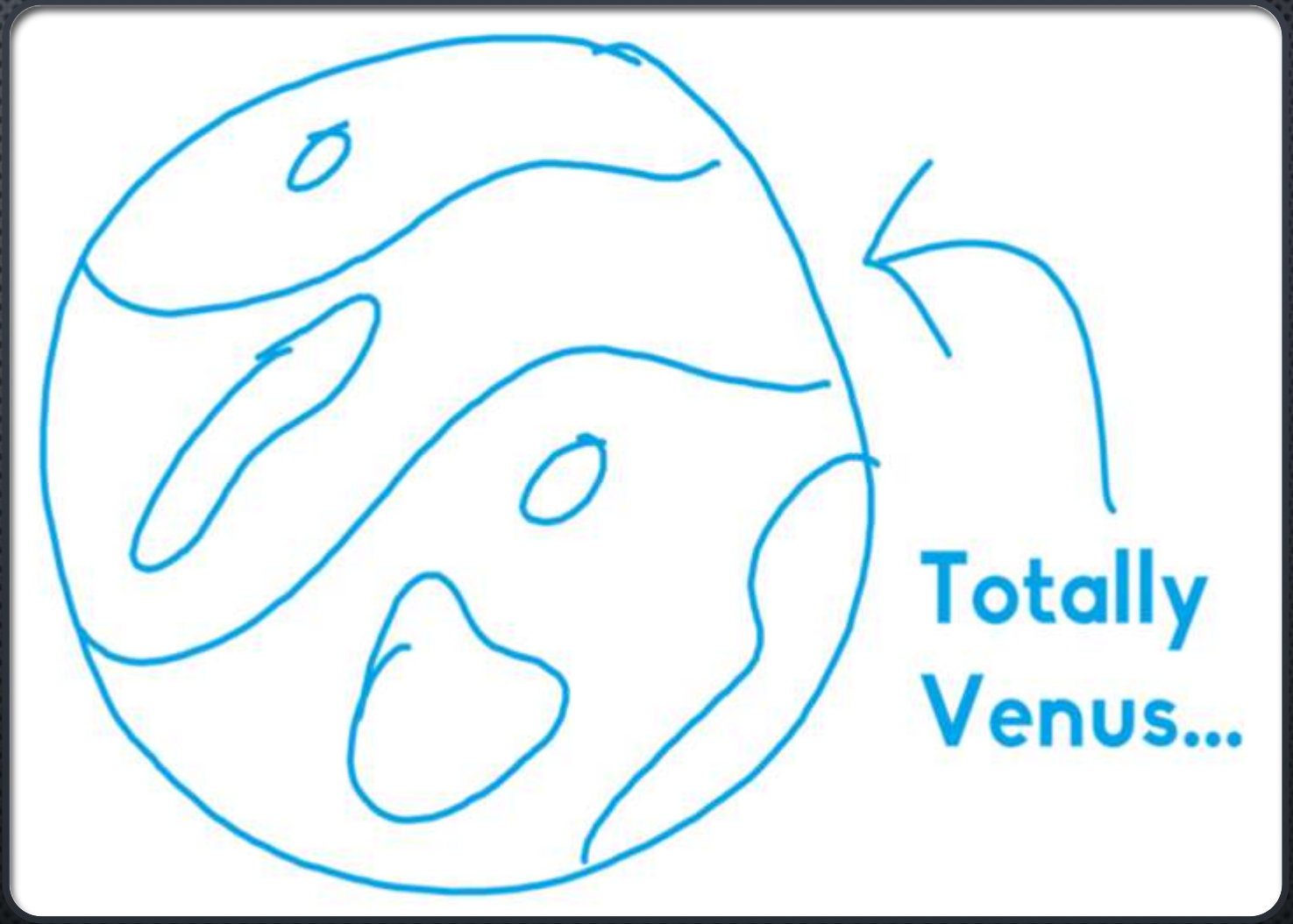
Let others work…



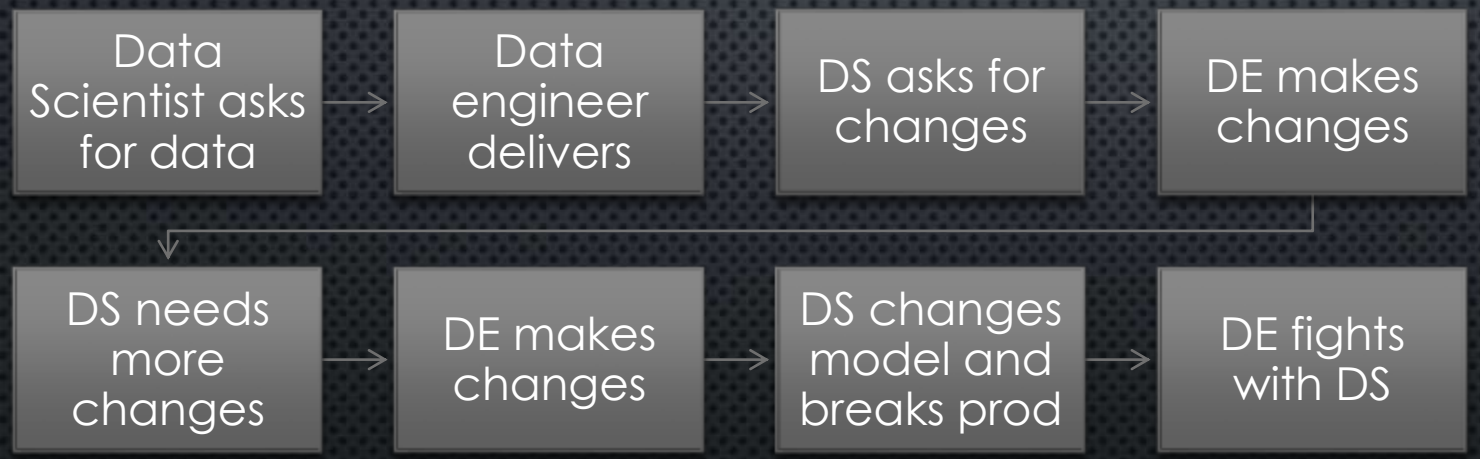
While you be yourself…

PREDICTING WEATHER PATTERNS ON VENUS

- A HYPOTHETICAL SCENARIO FOR THE INTERACTION OF A DATA SCIENTIST WITH TWO TYPES OF DATA ENGINEERS (NAVIGATOR VS HERO)



HERO DATA ENGINEER APPROACH



NAVIGATOR DATA ENGINEER APPROACH

Data Scientist asks for data → DE delivers data → DE documents process → DS asks for changes

DE builds data conversion tool → DS processes own changes → DS needs more changes, uses conversion tool → DE build CI/CD pipeline with approval

DS breaks model. Does not deploy because approval not met → DE goes on relaxing holiday

WHAT HAVE WE LEARNED?

- Design self-service platforms
- Scale beyond yourself
- Make yourself redundant
- Teach & Learn
- Enable others

ARE YOU A NAVIGATOR OR A HERO?