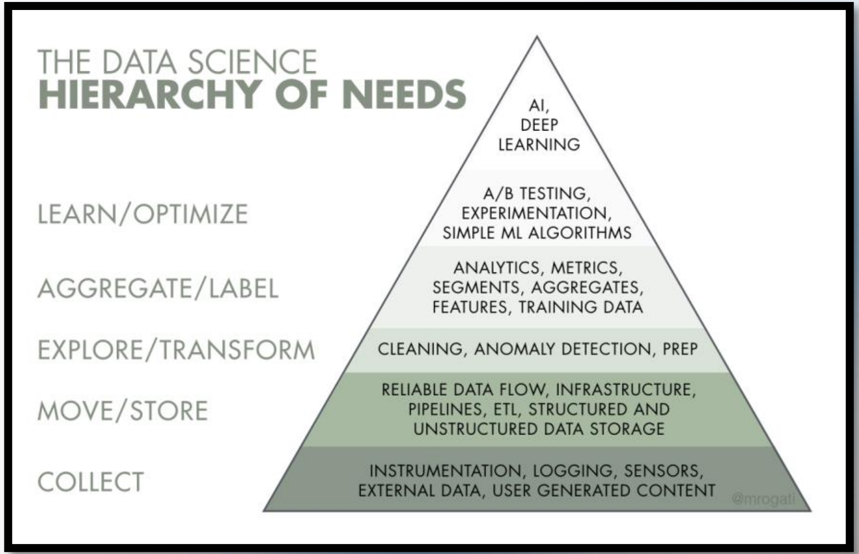
eXtreme Data Engineering

Kurtis Seebaldt, Director of Engineering

Artium (builds your products, internal capabilities, tech teams, & leadership skills)

Why Data Engineering?

The increasing complexity of data and data infrastructure requires software engineering discipline to support the needs of data science.


THE DATA SCIENCE
HIERARCHY OF NEEDS
LEARN/OPTIMIZE
AGGREGATE/LABEL
EXPLORE/TRANSFORM
MOVE/STORE
COLLECT
AI, DEEP LEARNING
A/B TESTING, EXPERIMENTATION, SIMPLE ML ALGORITHMS
ANALYTICS, METRICS, SEGMENTS, AGGREGATES, FEATURES, TRAINING DATA
CLEANING, ANOMALY DETECTION, PREP
RELIABLE DATA FLOW, INFRASTRUCTURE, PIPELINES, ETL, STRUCTURED AND UNSTRUCTURED DATA STORAGE
INSTRUMENTATION, LOGGING, SENSORS, EXTERNAL DATA, USER GENERATED CONTENT
@mrogati

Source: Monica Rogati https://hackernoon.com/the-ai-hierarchy-of-needs-18f111fcc007

The Data Swamp

Data requests start small, but soon grow out of control.

- One off data imports
- Ad hoc ETL scripts
- Manual, repetitive, time-consuming
- Data in various locations and formats
- No clear canonical source
- Not up-to-date

Challenges

Principles and practices need to be adapted to the complex data environment.

- Data infrastructure is complex
- Code deployment is not straightforward
- Few established patterns in the data environment
- Deploying code is not straightforward

eXtreme Data Engineering

Manage data and infrastructure by implementing best practices and methodologies.

01 Test Driven Development

- Write code in a local development environment
- Test first data transformations
- Commit to source control

02 Continuous Delivery

- Run tests on every commit
- Package code into libraries and/or containers
- Automated deploys to staging and production environments

03 Infrastructure as Code

- Automate provisioning of infrastructure
- Document configuration
- Replicate environments

Data for energy and gas: a 2-year journey

Initial Landscape

1. Data in many locations
2. One off data loads
3. Ad hoc scripts or ETL jobs

Issues

1. Takes too much time to ingest a new data source
2. Hard to know what is running
3. Failed jobs are hard to track
4. Making changes is difficult and error prone

Goals

1. Build initial data lake infrastructure
2. Build individual pipelines (new+old, in priority order) based on need
3. Test driven development
4. Continuous Integration
5. Automated deployments to a demo environment
6. Automated provisioning of data infrastructure
7. Repeatable production releases

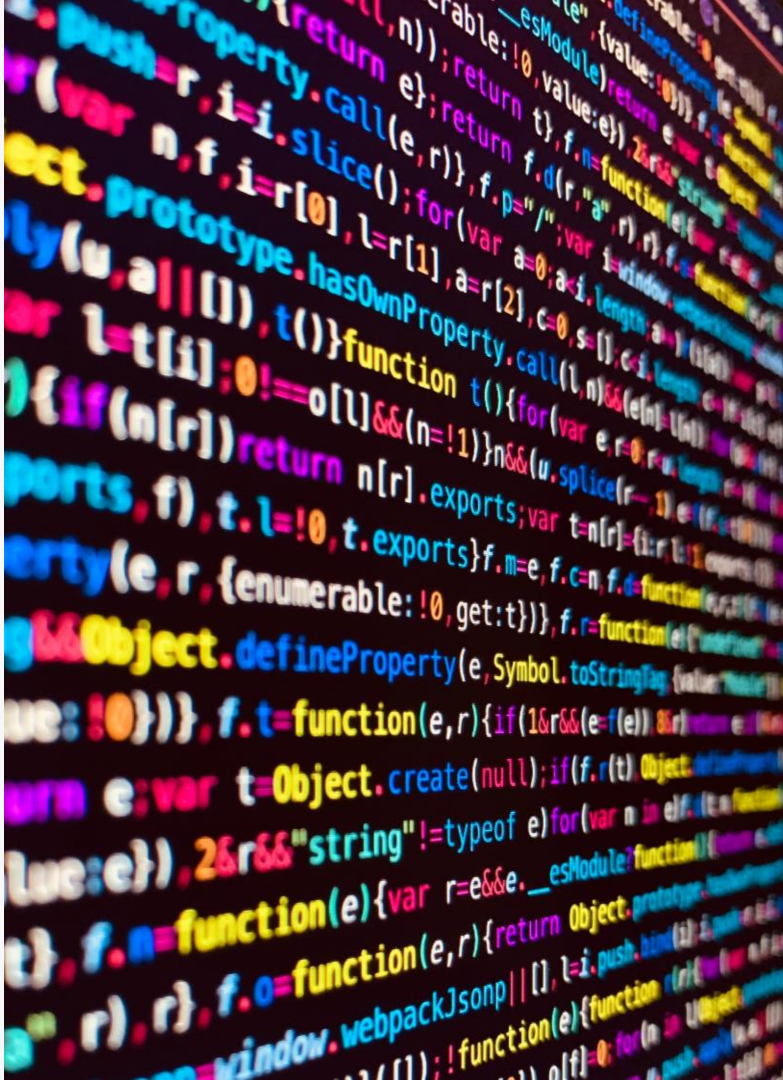
Test Driven Development

- Code can by developed and run on local development machine
- Unit test transformation functions for fast feedback
- Integration tests for pipeline job

```python
def test_converts_dates(spark):

    input = spark.read.format("csv").load("fixtures/austin_traffic/raw")

    output = transform_traffic_csv(spark, input)


    expected = spark.createDataFrame(
        [
            (
                "C163BCD1CF90C984E9EDA4DBA311BCA369A7D1A1_1528871759",
                isoparse("2018-06-13T06:35:59.000Z"),
                isoparse("2018-06-13T09:00:03.000Z"),
            ),
        ],
        ["traffic_report_id", "published_date", "traffic_report_status_date_time"],
    )

    assert_df_equality(
        output.select(
            "traffic_report_id", "published_date", "traffic_report_status_date_time"
        ),
        expected,

    )
```

CI / CD

- Run all tests on every check in
- Package code into libraries
- Deploy to demo/staging environment
- Libraries
- Containers
- Other configuration (Airflow DAGs, etc.)
- Tag commits to trigger production deploys



Infrastructure as Code

Use IaC tools such as Terraform to automate provisioning:

- Networking (VPC)
- Blob Storage (S3)
- Orchestrator (Airflow)
- Compute (Databricks, EMR)
- Credential Storage (AWS Secrets Manager)

Release Early, Release Often

Spending the time and effort to build good engineering practices:

- Enables ingesting and sharing new data sources quickly
- Produces fewer defects and data quality issues
- Eases ramp up new team members
- Allows team to focus on more complex, interesting, high-value data needs

Example using AWS Glue

https://github.com/kseebaldt/samplegluepipelines

Thank You

kurtis@thisisartium.com

Our mission is to empower every organization with the software development capabilities to achieve their vision of the future.