Operationalizing Big Data Workloads: Transformational Analytic Outcomes Through DataOps

Mark Marinelli, Head of Product
Introduction

A DataOps Framework

DataOps Success Stories

Getting Started With DataOps
What are Transformational Analytic Outcomes?

**Question:** How many customers do we have?

<table>
<thead>
<tr>
<th>Total Customers</th>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>225,097</td>
<td>124,145</td>
</tr>
</tbody>
</table>

- Enterprises have hundreds of source systems
- Sources must be combined, consolidated, and classified
- These lists are building blocks for transformational analytics

212 Sources (tables) - mostly SAP
What are Transformational Analytic Outcomes?

Question: What is our customer distribution by sales totals?

- Analytics begin with sell more and/or spend less
- Transformational analytics aren’t new, they are broader
- Business wants speed and up-to-date information
- Data variety skews answers, creating misinformation instead of clarity
What is DataOps? = Modern Data Engineering Practice

DataOps is an automated, process oriented methodology, used by analytic and data teams to improve the quality and reduce the cycle time of data analytics.
Why now? 7 years ago: we need data scientists!
Today: we have data scientists! (and want to do cool AI stuff)
But what about cleaning up/preparing our data

The New York Times
For Big-Data Scientists, ‘Janitor Work’ Is Key Hurdle to Insights

What data scientists spend the most time doing
- Building training sets: 3%
- Cleaning and organizing data: 60%
- Collecting data sets: 19%
- Mining data for patterns: 9%
- Refining algorithms: 4%
- Other: 5%

Data Scientist Survey by Figure Eight
Modern Internet Companies have data advantages

Greenfield Infrastructure & High End Talent Pool

Unified Dataportal

Description, external link, social

Metadata & consumption

Surface relationships, everything’s a link to promote exploration
Traditional companies have significant “legacy drag coefficient”

Manage data from their business systems more as “exhaust” than “asset” > “significant data debt”

**Problem:** Thousands of systems generating data every day that were built over decades to support business processes - idiosyncratic to that time/context.

Data is idiosyncratic to each system - creates fundamental **“data disconnect”** and “data decay”

**Result:** “Random Data Salad”
Data debt from constant change/entropy

**Consequences:**
1. Too much time spent on data prep vs. analysis / action.
2. High failure rate of BI / analytics projects
3. Game changing initiatives deemed ‘impossible’ and never start
Human/behavioral challenges often primary bottleneck

- **Afraid to share data**
  - Due to data quality (worry about being judged or having to take on the responsibility of fixing the data consumers’ requests)

- **Hoard data**
  - A method of organizational control or job preservation

- **Obscuring data complexity**
  - Failure to embrace the complexity, diversity, and idiosyncrasy of data generated in a large enterprise

- **Limiting access to a small number of users**
  - A method of control or as a reflection of insecurity of data quality
Serious Eye Chart defines challenge of picking tools
A DataOps Framework: Process, Technology, Organization
DataOps Framework Components

Process

- **Agile** - incremental delivery model

Technology

- **Architecture** - selection of tools which comprise data supply chain
- **Infrastructure** - selection of platform to support architecture

Organization

- **Roles** - division of labor across mixed-skill teams
- **Structure** - working model for projects across technical and business teams
Process - The Wrong Way

Sources

- Internal Tabular Data
- External Tabular Data

Technology, Organization, Process

- Modeling
- Testing
- Rules

Consumers

- Citizens
- Analysts
- Data Scientists
- Developers

- Labor-intensive
- Monolithic
- IT driven

Time

Remaining Work

Delivery

??

$\$
Process - The Right Way

Sources

- Internal Tabular Data
- External Tabular Data

Technology, Organization, Process

- Automated
- Incremental
- Collaborative

Time

Remaining Work

Consumers

- Citizens
- Analysts
- Data Scientists
- Developers
Technology - Architectural Principles

### Internal Tabular Data
- Scale Out/Distributed
  - Cloud First
- Collaborative (Humans at the Core)
  - Highly Automated - automate whenever possible
  - Bi-Directional (Feedback)
- Open/Best of Breed (not one platform/vendor)
  - Service Oriented (clear endpoints for data)
  - Loosely Coupled (Restful Interfaces Table(s) In/Out)
- Continuous (assume data will change)
  - Both aggregated AND federated storage
  - Both batch AND Streaming
- Lineage/Provenance is essential

### External Tabular Data

### Sources

- **Technology**
  - Organization, Process

### Consumers
- Citizens
- Analysts
- Data Scientists
- Developers
# Infrastructure - Key Components

<table>
<thead>
<tr>
<th>Sources</th>
<th>Technology, Organization, Process</th>
<th>Consumers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Internal Tabular Data</strong></td>
<td><strong>Management</strong></td>
<td><strong>Search</strong></td>
</tr>
<tr>
<td>![Internal Tabular Data Icon]</td>
<td>kubernetes</td>
<td>elasticsearch</td>
</tr>
<tr>
<td>![Internal Tabular Data Icon]</td>
<td>docker</td>
<td>Apache Solr</td>
</tr>
<tr>
<td><strong>Compute</strong></td>
<td>Amazon EMR</td>
<td>HBase</td>
</tr>
<tr>
<td>![Compute Icon]</td>
<td>Apache Spark</td>
<td>HDFS</td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td>Amazon S3</td>
<td>Apache HBase</td>
</tr>
<tr>
<td>![Storage Icon]</td>
<td>Apache HBase</td>
<td>Apache HBase</td>
</tr>
<tr>
<td><strong>Infrastructure</strong></td>
<td>AWS</td>
<td>Google Cloud</td>
</tr>
<tr>
<td>![Infrastructure Icon]</td>
<td>Google Cloud</td>
<td>Azure</td>
</tr>
<tr>
<td>![Infrastructure Icon]</td>
<td>Azure</td>
<td>Red Hat</td>
</tr>
<tr>
<td>![Infrastructure Icon]</td>
<td>Red Hat Linux</td>
<td>Developers</td>
</tr>
</tbody>
</table>

- **Citizens**
- **Analysts**
- **Data Scientists**
- **Developers**
Organization - Roles

### Sources
- **Internal Tabular Data**
- **External Tabular Data**

### Technology, Organization, Process
- **CIO**
  - Source Owner
  - DBA
  - IT Professional
- **CDO**
  - Data Engineer
  - Curator
  - Steward
- **Business Owners and Other CxOs**

### Consumers
- **Citizens**
- **Analysts**
- **Data Scientists**
- **Developers**
## Organization - Roles

<table>
<thead>
<tr>
<th>Role</th>
<th>Goals</th>
<th>Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Consumers</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Citizen</td>
<td>Use data to make business decisions</td>
<td>Viz, CRM, Excel, PowerPoint, Word, Web Search</td>
</tr>
<tr>
<td>Analyst</td>
<td>Deliver insights to the business, typically through dashboards and reports</td>
<td>Viz, Excel, SSDP, Web Search</td>
</tr>
<tr>
<td>Scientist</td>
<td>Deliver insights to the business, typically through models and algorithms</td>
<td>R, Python, SAS, SSDP</td>
</tr>
<tr>
<td>Developer</td>
<td>Build applications which leverage corporate data</td>
<td>Python, Java, JS, SQL, REST</td>
</tr>
<tr>
<td><strong>Preparers</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engineer</td>
<td>Deliver and manage data pipelines</td>
<td>ETL, SQL</td>
</tr>
<tr>
<td>Curator</td>
<td>Ensure consumers have the data they need, in the form they need it</td>
<td>MDM, Catalog</td>
</tr>
<tr>
<td>Steward</td>
<td>Create policies and drive governance</td>
<td>MDM, Catalog, Governance</td>
</tr>
<tr>
<td><strong>Suppliers</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source Owner</td>
<td>Define and manage purpose, processes (data creation, consumption) &amp; users (i.e., access) of the data source</td>
<td>EDW, SQL, ERWin, LDAP, SAP</td>
</tr>
</tbody>
</table>
Organization - Structure

Advisory Model
Bootstraps projects with best of breed tools and approach, but does not complete them

Advantages
- Centralized technical knowledge
- Minimal resourcing - experts, not implementers
- Flexibility - options to deviate from standard tools

Disadvantages
- Resource burden in on each project / department - both in development and ongoing maintenance
- Limited feedback - does the advice get better after each project?

Shared Services Model
Full-service development of data applications, in collaboration with business

Advantages
- Centralized technical knowledge
- Centralized resourcing - one-stop shop
- Accretive experience

Disadvantages
- Bandwidth contention - how to prioritize competing projects?

Appropriate model will fluctuate with scale of DataOps project work
DataOps Success Stories
A major financial institution built a data lab that works to invent solutions that harness data and advanced analytics.

Goals
- Better understanding of 60 million customers
- Create simpler, more intuitive and intelligent products and customer experiences
- Help businesses do more business with each other using the bank’s cards

Holistic approach
- Interdisciplinary Shared Service team is made up of DevOps and data scientists
- Mingles human-centered design, full-stack engineering and data science
- Organized around Scaled Agile Framework

Hybrid tooling - rules + machine learning
- Data Engineers cleanse and deduplicate with rules first, then fed into ML for classification and training
- Subject matter experts act as Curators to improve accuracy of ML models
A major pharmaceutical company realized that its R&D environment wasn’t up to par, which was preventing them from developing new drugs with the level of innovation and speed required.

Goals
- Make it easier to access and use data for exploratory analysis and decision-making about new medicines
- CDO: Identify highest-value work for COE approach

Approach
- Conducted a survey about data across the Consumers
  - Result: very difficult to work with data outside of a departmental silo
  - Identified top 10 use cases for integrating diverse data

Results and Benefits
- Turned to machine learning since a traditional MDM approach would have taken too long
- Use cases have expanded from 10 to 250
- Reduction in time to get answers to ad hoc questions
Getting Started With DataOps
DataOps Framework Components

Process

- **Agile** - incremental delivery model

Technology

- **Architecture** - selection of tools which comprise data supply chain
- **Infrastructure** - selection of platform to support architecture

Organization

- **Roles** - division of labor across mixed-skill teams
- **Structure** - working model for projects across technical and business teams
Getting Started - Process

Agile is the key
- If not already there, choose a model that works (Scrum, SAFe)

Inventory the set of available projects
- Score on availability of data vs. value of solving a problem

Define high-value, data-rich project that will demand a complex solution
- Forcing function to ensure end-to-end functionality will be covered
Getting Started - Technology

Identify path to a modern, modular service architecture
  ● Create blueprint for next generation data management platform
  ● Revisit cloud migration strategy

Inventory current tool set
  ● TCO / skill requirements / etc.
  ● Determine which should be replaced, and when this is viable

Decouple monolithic processes
  ● Wrap components in APIs, expose as services

Start building with new tech
  ● Choose subset of tools for proof of concepts to replace old tech
Getting Started - Organization

Inventory current team
- Identify existing key roles - data engineers and their consumers
- Find best candidates for new roles - data curators and data stewards

Create cross-functional team
- Data consumers - will depend upon project
- Data Engineer(s)
- Data Curator
- Data Steward

Choose your operating model
- Start with Shared Services for first project

Ensure executive alignment
- CDO or equivalent