

# THE



# COOKBOOK

Methodologies and Tools That Reduce Analytics  
Cycle Time While Improving Quality

# **The DataOps Cookbook**

**Methodologies and Tools That Reduce  
Analytics Cycle Time While Improving Quality**

Second Edition

**by  
Christopher Bergh, Gil Benghiat,  
and Eran Strod**

The DataOps Cookbook

© 2019 DataKitchen, Inc. All Rights Reserved.

To order additional copies of this book:

**[info@datakitchen.io](mailto:info@datakitchen.io)**

DataKitchen Headquarters:

101 Main Street, 14th Floor

Cambridge, MA 02142

Our Mailing Address:

One Broadway, 14th Floor

Cambridge, MA 02142

Printed in the United States of America

Cover design and layout by Ariel Plotkin-Gould

# Table of Contents

|   |           |
|---|-----------|
| <b>PREFACE TO THE SECOND EDITION</b>        | <b>1</b>  |
| <b>INTRODUCTION</b>                         | <b>3</b>  |
| <b>THE DATAOPS MANIFESTO</b>                | <b>5</b>  |
| DataOps Principles                          | 5         |
| 1. Continually satisfy your customer        | 5         |
| 2. Value working analytics                  | 5         |
| 3. Embrace change                           | 6         |
| 4. It's a team sport                        | 6         |
| 5. Daily interactions                       | 6         |
| 6. Self-organize                            | 6         |
| 7. Reduce heroism                           | 6         |
| 8. Reflect                                  | 6         |
| 9. Analytics is code                        | 6         |
| 10. Orchestrate                             | 6         |
| 11. Make it reproducible                    | 6         |
| 12. Disposable environments                 | 6         |
| 13. Simplicity                              | 7         |
| 14. Analytics is manufacturing              | 7         |
| 15. Quality is paramount                    | 7         |
| 16. Monitor quality and performance         | 7         |
| 17. Reuse                                   | 7         |
| 18. Improve cycle times                     | 7         |
| <b>EIGHT CHALLENGES OF DATA ANALYTICS</b>   | <b>10</b> |
| 1 – The Goalposts Keep Moving               | 11        |
| 2 – Data Lives in Silos                     | 11        |
| 3 – Data Formats are not Optimized          | 12        |
| 4 – Data Errors                             | 12        |
| 5 – Bad Data Ruins Good Reports             | 13        |
| 6 – Data Pipeline Maintenance Never Ends    | 13        |
| 7 – Manual Process Fatigue                  | 13        |
| 8 – The Trap of “Hope and Heroism”          | 13        |
| Overcoming the Challenges                   | 14        |
| <b>“WHAT IS DATAOPS”</b>                    | <b>17</b> |
| Delivering Analytics at <i>Amazon Speed</i> | 17        |
| The Seven Steps to Implement DataOps        | 27        |
| Step 1 - Add Data and Logic Tests           | 27        |
| Step 2 - Use a Version Control System       | 28        |
| Step 3 - Branch and Merge                   | 29        |
| Step 4 - Use Multiple Environments          | 30        |
| Step 5 - Bad Data Ruins Good Reports        | 30        |
| Step 6 - Parameterize Your Processing       | 31        |
| Step 7: Work Without Fear or Heroism        | 32        |

|   |            |
|---|------------|
| DataOps is NOT Just DevOps for Data   | 35         |
| DataOps Resolves the Struggle Between Centralization and Freedom in Analytics                 | 47         |
| <b>DATAOPS FOR THE CHIEF DATA OFFICER</b>   | <b>56</b>  |
| Warring Tribes into Winning Teams: Improving Teamwork in Your Data Organization — <b>NEW!</b> | 56         |
| Improving Teamwork in Data Analytics with DataOps — <b>NEW!</b>                               | 65         |
| Eliminate Your Analytics Development Bottlenecks — <b>NEW!</b>                                | 78         |
| Prove Your Awesomeness with Data: The CDO DataOps Dashboard                                   | 86         |
| Surviving Your Second Year as CDO   | 91         |
| CAOs and CDOs: Earn the Trust of your CEO   | 95         |
| The Four Stage Journey to Analytics Excellence  | 97         |
| <b>DATAOPS FOR THE DATA ENGINEER AND THE DATA SCIENTIST</b>                                   | <b>102</b> |
| A Great Model is Not Enough: Deploying AI Without Technical Debt — <b>NEW!</b>                | 102        |
| The “Right to Repair” Data Architecture with DataOps — <b>NEW!</b>                            | 110        |
| Enabling Design Thinking in Data Analytics with DataOps — <b>NEW!</b>                         | 115        |
| DataOps Puts Agility into Agile Data Warehousing  | 119        |
| Speed Up Innovation with DataOps  | 121        |
| How to Inspire Code Reuse in Data Analytics   | 124        |
| What Data Scientists Really Need  | 126        |
| <b>DATAOPS FOR DATA QUALITY</b>   | <b>130</b> |
| Disband Your Impact Review Board: Automate Analytics Testing                                  | 130        |
| Build Trust Through Test Automation and Monitoring  | 139        |
| How Data Analytics Professionals Can Sleep Better   | 145        |
| <b>DATAOPS AND YOUR CAREER</b>  | <b>149</b> |
| DataOps Engineer Will Be the Sexiest Job in Analytics   | 149        |
| Building a DataOps Team   | 151        |
| <b>DATAOPS EXAMPLES AND CASE STUDIES</b>  | <b>156</b> |
| Grow Sales Using a DataOps-Powered Customer Data Platform                                     | 156        |
| Achieving Growth Targets by Implementing a DataOps-Powered Customer Data Platform             | 160        |
| How a Mixed Martial Arts Fighter Would Approach Data Analytics                                | 164        |
| Reinvent Marketing Automation with the DataKitchen DataOps Platform                           | 166        |
| Meeting the Product Launch Challenge with DataOps   | 168        |
| <b>DATAOPS SURVEY</b>   | <b>173</b> |
| Tomorrow’s Forecast: Cloudy with a Chance of Data Errors — <b>NEW!</b>                        | 173        |
| <b>ADDITIONAL RECIPES</b>   | <b>178</b> |
| <b>DATAOPS RESOURCES</b>  | <b>181</b> |
| <b>ABOUT THE AUTHORS</b>  | <b>182</b> |

# Preface to the Second Edition

Welcome to the Second Edition of the “DataOps Cookbook.” With over 5,000 copies distributed, the first edition of the book far exceeded our expectations. Managers have asked us for boxes of books to distribute to their entire organization. Data professionals are forming study groups around the “DataOps Cookbook.” DataOps is a methodology truly coming of age.

The name “DataOps” has always been somewhat problematic, misleading people to believe that we are simply talking about *DevOps for data*. This misconception started to gain traction in the technical press in mid-2018, shortly after Gartner placed DataOps on the fastest rising part of their Hype Cycle curve for Data Management.

In response, we wrote the post “DataOps is NOT Just DevOps for Data” (see the chapter “What is DataOps” below). On Medium, the post has received over 28,000 views (and counting), making it one of 2019’s most widely read and referenced thought pieces on data analytics. The DataOps view that analytics is a combination of software development and manufacturing operations seems to have struck a chord within the data industry.

The remarkable interest in DataOps has opened the door to many conversations with data professionals, both in individual contributor and management roles. These discussions spurred further thinking about DataOps, and we are now pleased to expand upon the original book with several new additions. We hope these further advance the industry-wide dialogue about data organization productivity and quality. In this latest edition of the DataOps Cookbook, you’ll find the following new sections:

- “Warring Tribes into Winning Teams: Improving Teamwork in Your Data Organization” on inter-team teamwork
- “Improving Teamwork in Data Analytics with DataOps” on intra-team collaboration
- “Eliminate Your Analytics Development Bottlenecks”
- “A Great Model is Not Enough. Deploying AI Without Technical Debt.”
- “The ‘Right To Repair’ DataOps Data Architecture”
- “Enabling Design Thinking in Data Analytics with DataOps”

Also, we present the surprising results of our DataOps survey:

- “Tomorrow’s Forecast: Cloudy with a Chance of Data Errors — Key Findings of the 2019 DataOps Survey “

DataOps is a foundational topic that requires data teams to fundamentally rethink the ways that they perform their duties. Despite the inherent challenges, we are confident you will find this to be a fruitful and worthwhile endeavor. We look forward to continuing the conversation.

# Introduction

In the early 2000s, Chris and Gil worked at a company that specialized in analytics for the pharmaceutical industry. It was a small company that offered a full suite of services related to analytics — data engineering, data integration, visualization and what is now called “data science.” Their customers were marketing and sales executives who tend to be challenging because they are busy, need fast answers and don’t understand or care about the underlying mechanics of analytics. They are business people, not technologists.

When a request from a customer came in, Chris and Gil would gather their team of engineers, data scientists and consultants to plan out the how to get the project done. After days of planning, they would propose their project plan to the customer. “It will take two weeks.” The customer would shoot back, “I need it in two hours!”

Walking back to their office, tail between their legs, they would pick up the phone. It was a customer boiling over with anger. There was a data error. If it wasn’t fixed immediately the customer would find a different vendor.

The company had hired a bunch of smart people to deliver these services. “I want to innovate — Can I try out this new open source tool,” the team members would ask. “No,” the managers would have to answer. “We can’t afford to introduce technical risk.”

They lived this life for many years. How do you create innovative data analytics? How do you not have embarrassing errors? How do you let your team easily try new ideas? There had to be a better way.

They found their answer by studying the software and manufacturing industries which had been struggling with these same issues for decades. They discovered that data-analytics cycle time and quality can be optimized with a combination of tools and methodologies that they now call **DataOps**. They decided to start a new company. The new organization adopted the kitchen metaphor for data analytics. After all, cooking up charts and graphs requires the right *ingredients* and *recipes*.

The DataKitchen founders (Chris, Gil and Eric) built their own DataOps Platform. What happened next was remarkable. When analytics go faster, users embrace analytics and innovate. Rapid, high-quality analytics can unlock an organization's creative potential.

Having experienced this transformation, the DataKitchen founders sought a way to help other data professionals. There are so many talented people stuck in no-win situations. This book is for data professionals who are living the nightmare of slow, buggy analytics and frustrated users. It will explain why working weekends isn't the answer. It provides you with practical steps that you can take tomorrow to improve your analytics cycle time.

DataKitchen markets a DataOps Platform that will help analytics organizations implement DataOps. However, this book isn't really about us and our product. It is about you, your challenges, your potential and getting your analytics team back on track.

The values and principles that are central to DataOps are listed in the DataOps Manifesto which you can read below. If you agree with it, please join the thousands of others who share these beliefs by signing the manifesto. There may be aspects of the manifesto that require further explanation. Please read on. By the end of this book, it should all make sense.

You'll also notice that we've included some real recipes in this book. These are some of our favorites. We hope you enjoy them!

Please reach out to us at [info@datakitchen.io](mailto:info@datakitchen.io) with any comments or questions.

Chris, Gil and Eran

# The DataOps Manifesto

## Background

Through firsthand experience working with data across organizations, tools, and industries we have uncovered a better way to develop and deliver analytics that we call DataOps. Whether referred to as data science, data engineering, data management, big data, business intelligence, or the like, through our work we have come to value in analytics:

- Individuals and interactions over processes and tools
- Working analytics over comprehensive documentation
- Customer collaboration over contract negotiation
- Experimentation, iteration, and feedback over extensive upfront design
- Cross-functional ownership of operations over siloed responsibilities

## DataOps Principles

### 1. CONTINUALLY SATISFY YOUR CUSTOMER

Our highest priority is to satisfy the customer through the early and continuous delivery of valuable analytic insights from a couple of minutes to weeks.

### 2. VALUE WORKING ANALYTICS

We believe the primary measure of data analytics performance is the degree to which insightful analytics are delivered, incorporating accurate data, atop robust frameworks and systems.

### **3. EMBRACE CHANGE**

We welcome evolving customer needs, and in fact, we embrace them to generate competitive advantage. We believe that the most efficient, effective, and agile method of communication with customers is face-to-face conversation.

### **4. IT'S A TEAM SPORT**

Analytic teams will always have a variety of roles, skills, favorite tools, and titles.

### **5. DAILY INTERACTIONS**

Customers, analytic teams, and operations must work together daily throughout the project.

### **6. SELF-ORGANIZE**

We believe that the best analytic insight, algorithms, architectures, requirements, and designs emerge from self-organizing teams.

### **7. REDUCE HEROISM**

As the pace and breadth of need for analytic insights ever increases, we believe analytic teams should strive to reduce heroism and create sustainable and scalable data analytic teams and processes.

### **8. REFLECT**

Analytic teams should fine-tune their operational performance by self-reflecting, at regular intervals, on feedback provided by their customers, themselves, and operational statistics.

### **9. ANALYTICS IS CODE**

Analytic teams use a variety of individual tools to access, integrate, model, and visualize data. Fundamentally, each of these tools generates code and configuration which describes the actions taken upon data to deliver insight.

### **10. ORCHESTRATE**

The beginning-to-end orchestration of data, tools, code, environments, and the analytic team's work is a key driver of analytic success.

### **11. MAKE IT REPRODUCIBLE**

Reproducible results are required and therefore we version everything: data, low-level hardware and software configurations, and the code and configuration specific to each tool in the toolchain.

### **12. DISPOSABLE ENVIRONMENTS**

We believe it is important to minimize the cost for analytic team members to experiment by giving them easy to create, isolated, safe, and disposable technical environments that reflect their production environment.

### **13. SIMPLICITY**

We believe that continuous attention to technical excellence and good design enhances agility; likewise simplicity—the art of maximizing the amount of work not done—is essential.

### **14. ANALYTICS IS MANUFACTURING**

Analytic pipelines are analogous to lean manufacturing lines. We believe a fundamental concept of DataOps is a focus on process-thinking aimed at achieving continuous efficiencies in the manufacture of analytic insight.

### **15. QUALITY IS PARAMOUNT**

Analytic pipelines should be built with a foundation capable of automated detection of abnormalities (jidoka) and security issues in code, configuration, and data, and should provide continuous feedback to operators for error avoidance (poka yoke).

### **16. MONITOR QUALITY AND PERFORMANCE**

Our goal is to have performance, security and quality measures that are monitored continuously to detect unexpected variation and generate operational statistics.

### **17. REUSE**

We believe a foundational aspect of analytic insight manufacturing efficiency is to avoid the repetition of previous work by the individual or team.

### **18. IMPROVE CYCLE TIMES**

We should strive to minimize the time and effort to turn a customer need into an analytic idea, create it in development, release it as a repeatable production process, and finally refactor and reuse that product.

Join the Thousands of People Who Have Already Signed The Manifesto



# DataOps Ribeye

*by Christopher Bergh*

## INGREDIENTS AND TOOLS

- Rib eye steaks 1 ½-2" thick
- Large baking potatoes
- Corn on the husk
- Olive oil
- Ground sea salt & Fresh coarse ground pepper or Penzey's Chicago Steak Seasoning
- Green Egg or another charcoal grill
- Instapen thermometer

## PREPARE CORN AND POTATOES

1. Season/marinade steaks to your liking, but I now like the 'TRex way of prepping/grilling my steaks (see below): coat the steaks with olive oil, liberally apply salt and pepper (or Penzey's Chicago Steak Seasoning) and rub the mixture into the steaks on both sides.
2. Soak corn in water for at least 30 minutes or more.
3. Brush potatoes with olive oil and salt on both sides. Heat Green Egg to 400 degrees with a few chunks of Hickory wood and put potatoes directly on grill grid, direct heat, turning once after 30 minutes. About 20-25 minutes before the potatoes are done, put the corn on the grill grid. Turn the corn about every 8-10 minutes or just enough to keep the corn from getting burned (a little charring of the corn is okay, though).
4. Remove the potatoes and corn and wrap in aluminum foil and place in a warming oven or drawer at 170 degrees.

## PREPARE THE STEAKS

1. TRex method of grilling/cooking steaks: Crank the draft door and daisy wheel open all the way on the Egg for maximum temperature. When you get the Green Egg up to at least over 600 degrees (be careful to "burp" the Egg at this high temp), put the steaks on and sear for 90 seconds per side.
2. Take the steaks off the Egg and let sit/rest for 20 minutes. Shut down the lower vent and Daisy Wheel to get the Green Egg back down to 425 degrees.
3. Throw some more chunks of Hickory wood on the fire and try and maintain a temp around 400 degrees. After 10 minutes of letting the steaks sit/rest put back on the Green Egg and cook for approximately 4-5 minutes per side for a medium-rare/medium result. Check internal temperature of steak so that it is 120 degrees

# Eight Challenges of Data Analytics

Companies increasingly look to analytics to drive growth strategies. As the leader of the data-analytics team, you manage a group responsible for supplying business partners with the analytic insights that can create a competitive edge. Customer and market opportunities evolve quickly and drive a relentless series of questions. Analytics, by contrast, move slowly, constrained by development cycles, limited resources and brittle IT systems. The gap between what users need and what IT can provide can be a source of conflict and frustration. Inevitably this mismatch between expectations and capabilities can cause dissatisfaction, leaving the data-analytics team in an unfortunate position and preventing a company from fully realizing the strategic benefit of its data.



As a manager overseeing analytics, it's your job to understand and address the factors that prevent the data-analytics team from achieving peak levels of performance. If you talk to your team, they will tell you exactly what is slowing them down. You'll likely hear variations of the following eight challenges:

# 1 – The Goalposts Keep Moving

Users are demanding customers for a data-analytics team. Their requirements change constantly. They require immediate responses, and no matter how much the analytics team delivers, users keep generating new requests. It's enough to overwhelm any data-analytics team.

*They don't know what they want.* Users are not data experts. They don't know what insights are possible until someone from your team shows them. Sometimes they don't know what they want until after they see it in production (and maybe not even then). Often, business stakeholders do not know what they will need next week, let alone next quarter or next year. It's not their fault. It's the nature of pursuing opportunities in a fast-paced marketplace.

*They need everything ASAP.* Business is a competitive endeavor. When an opportunity opens, the company needs to move on it faster than the competition. When users bring a question to the data-analytics team, they expect an immediate response. They can't wait weeks or months — the opportunity will close as the market seeks alternative solutions.

*The questions never end.* Sometimes providing business stakeholders with analytics generates more questions than answers. Analytic insights enable users to understand the business in new ways. This spurs creativity, which leads to requests for more analytics. A healthy relationship between the analytics and users will foster a continuous series of questions that drive demand for new analytics. However, this relationship can sour quickly if the delivery of new analytics can't meet the required time frames.



## 2 – Data Lives in Silos

In pursuit of business objectives, companies collect an enormous amount of data: orders, deliveries, returns, website page views, mobile app navigations, downloads, clicks, metrics, audio logs, social media and more. Further, this data can be combined with demographic,

psychographic or other third-party market data. All of this data is collected in separate databases which typically do not talk to each other. They utilize numerous platforms, APIs and technologies. Accessing all of this data is a daunting task requiring such a wide range of skills that it is rare to find a single person who can do it all. Integrating data from these myriad sources becomes a major undertaking.

Business stakeholders want fast answers. Meanwhile, the data-analytics team has to work with IT to gain access to operational systems, plan and implement architectural changes, and develop/test/deploy new analytics. This process is complex, lengthy and subject to numerous bottlenecks and blockages.

## 3 – Data Formats are not Optimized

Data in operational systems is usually *not* structured in a way that lends itself to the efficient creation of analytics. For example, an ERP system might have a schema that is optimized for inserts, updates, and for display in a web user interface. For operational systems, these are the actions that need to happen in real time.



A database optimized for data analytics is structured to optimize reads and aggregations. It's also important for the schema of an analytics database to be easily understood by humans. For example, the field names would be descriptive of their contents and data tables would be linked in ways that make intuitive sense.

## 4 – Data Errors

Whether your data sources are internal or from external third parties, data will eventually contain errors. Data errors can prevent your data pipeline from flowing correctly. Errors may also be subtle, such as duplicate records or individual fields that contain erroneous data. Data errors could be caused by a new algorithm that doesn't work as expected, a database schema change that broke one of your feeds, an IT failure or one of many other possibilities. Data errors can be difficult to trace and resolve quickly.

## 5 – Bad Data Ruins Good Reports

When data errors work their way through the data pipeline into published analytics, internal stakeholders can become dissatisfied. This causes unplanned work, which diverts your key contributors from the highest priority projects. Bad data also harms the hard-won credibility of the data-analytics team. If business colleagues repeatedly see bad data in analytics reports, they might learn not to trust or value the work product of the data-analytics team.

## 6 – Data Pipeline Maintenance Never Ends

Data-analytics is a pipeline process that executes a set of operations and attempts to produce a consistent output at a high level of quality. Every new or updated data source, schema enhancement, analytics improvement or other change triggers an update to the pipeline. The data-analytics team is continuously making changes and improvements to the data pipeline. Each one of these changes must be made carefully so that it doesn't break operational analytics. The effort required to validate and verify changes often takes longer than the time required to create the changes in the first place. You may not realize it, but your analysts, [data scientists](#) and engineers may be spending 80% of their time updating, maintaining and assuring the quality of the data pipeline. This is necessary work, but much of it is behind the scenes and unappreciated when viewed against the growing backlog of new requests from business customers.

## 7 – Manual Process Fatigue

Data integration, cleansing, transformation, quality assurance and deployment of new analytics must be performed flawlessly day in and day out. The data-analytics team may have automated a portion of these tasks, but some teams perform numerous manual processes on a regular basis. These rote procedures are error prone, time consuming and tedious.

Further, manual processes can also lead to high employee turnover. Many managers have watched high-performing data-analytics team members burn out due to having to repeatedly execute manual data procedures. Manual processes strain the productivity of the data team in numerous ways.

## 8 – The Trap of “Hope and Heroism”

The landscape is littered with projects and initiatives cancelled or deferred due to changing requirements, slipped schedules, disappointed users, inflexibility, poor quality, low ROI, and irrelevant features.

According to the research firm Gartner, Inc., half of all [chief data officers](#) (CDO) in large organizations will not be deemed a *success* in their role. Per Forrester Research, 60% of the data and analytics decision-makers surveyed said they *are not very confident* in their analytics insights. Only ten percent responded that their organizations sufficiently *manage the quality of data and analytics*. Just sixteen percent believe they perform well in producing *accurate models*.

Many CDO's and data-analytics professionals respond to these challenges in one of three ways:

**Heroism** - Data-analytics teams work long hours to compensate for the gap between performance and expectations. When a deliverable is met, the data-analytics team is considered heroes. However, yesterday's heroes are quickly forgotten when there is a new deliverable to meet. Also, this strategy is difficult to sustain over a long period of time, and it, ultimately, just resets expectations at a higher level without providing additional resources. The heroism approach is also difficult to scale up as an organization grows.

**Hope** - When a deadline must be met, it is tempting to just quickly produce a solution with minimal testing, push it out to the users and *hope* it does not break. This approach has inherent risks. Eventually, a deliverable will contain data errors, upsetting the users and harming the hard-won credibility of the data-analytics team.

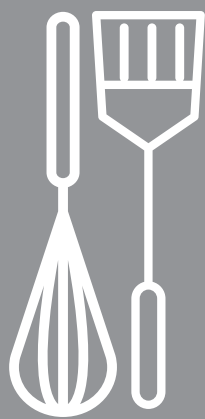
**Caution** - The team decides to give each data-analytics project a longer development and test schedule. Effectively, this is a decision to deliver higher quality, but fewer features to users. One difficulty with this approach is that users often don't know what they want until they see it, so a detailed specification might change considerably by the end of a project. The slow and methodical approach might also make the users unhappy because the analytics are delivered more slowly than their stated delivery requirements and as requests pile up, the data-analytics team risks being viewed as bureaucratic and inefficient.

None of these approaches adequately serve the needs of both users and data-analytics professionals, but there is a way out of this bind. The challenges above are not unique to analytics, and in fact, are shared by other organizations.

## Overcoming the Challenges

Some say that an analytics team can overcome these challenges by buying a new tool. While it is true that new tools are helpful, they are not enough by themselves. You cannot truly transform your staff into a high-performance team without an overhaul of the methodologies and processes that guide your workflows. In this book, we will discuss how to combine tools and new processes in a way that improves the productivity of your data analytics team by orders of magnitude.





# DataOps Chili Mole

*by Gil Benghiat*

## INGREDIENTS AND TOOLS

- 1 pound ground beef (90% lean or leaner)
- 1 diced onion
- 2 diced Jalapeño or 2 Serrano or 1 habanero pepper
- 2 pounds frozen sweet corn
- 28 oz can crushed tomatoes
- 15.5 oz can of black beans, NOT drained
- 15.5 oz can of black beans, drained
- 15.5 oz can of red beans, drained
- 15.5 oz can of white beans, drained
- 7 cloves of crushed garlic
- ½ cup sugar
- 3 ½ table spoons of unsweetened cocoa powder
- 3 table spoons of chili powder
- 1 teaspoon of salt

## INSTRUCTIONS

1. Crock Pot: Combine all ingredients and cook on high for 5-8 hours. Stir occasionally.
2. Stove Top: Combine ground beef, onion, and pepper. Cook on medium high until beef is cooked through. Add the remaining ingredients and cook on low-simmer for 1-2 hours. Stir occasionally.
3. For vegan chili: Substitute 5 tablespoons of canola oil for the ground beef.
4. Serve with rice.

# “What Is DataOps”

You can view DataOps in the context of a century-long evolution of ideas that improve how people manage complex systems. It started with pioneers like W. Edwards Deming and [statistical process control](#) - gradually these ideas crossed into the technology space in the form of [Agile](#), [DevOps](#) and now, [DataOps](#). In the next section we will examine how these methodologies impact productivity, quality and reliability in data analytics.

## Delivering Analytics at *Amazon Speed*

The world changed in February 2005 when Amazon Prime brought flat-rate, unlimited, two-day shipping into a world where people expected to pay extra to receive packages in four to six business days. Since its launch, Amazon Prime has completely transformed the retail market, making low-cost, predictable shipping an integral part of consumer expectations. This business model, which some have called the “on-demand economy,” is popping up in many industries and markets across the globe.

For example, some may remember video stores where movies were rented for later viewing. Today, 65 percent of global respondents to a recent Nielsen survey watch video on demand (VOD), many of them daily. With VOD, a person's desire to watch a movie is fulfilled within seconds. Amazon participates in the VOD market with their Amazon Prime Video service.

Instant fulfillment of customer orders seems to be part of Amazon's business model. They have even brought that capability to IT. About 10 years ago, Amazon Web Services (AWS) began offering computing, storage, and other IT infrastructure on an as-needed basis. Whether the need is for one server or thousands and whether for hours, days, or months, you only pay for what you use, and the resources are available in just a few minutes.

To successfully compete in today's on-demand economy, companies need to deliver their products and services just as Amazon has done—in other words, at *Amazon speed*. What might be surprising to many is how the expectations of instant fulfillment are crossing over

into data analytics, which, along with everything else in the digital economy, is now expected to happen at Amazon speed and with Amazon predictability.

A typical example: the VP of sales enters the office of the [chief data officer](#) (CDO). She'd like to cross-reference the customer database with some third-party consumer data. The CDO asks for time to study the problem and, days later, has planned the project. Resources will be allocated and configured, schemas will be updated, reports will be elegantly designed, and the delivery pipeline will be thoroughly tested. The changes will take several weeks. "Not acceptable," the VP of sales fires back. The new analytics are needed for a meeting with the board later in the week. "The competition is ahead of us; we can't wait weeks." This scenario is playing out in one form or another in corporations around the globe.



## ANALYTICS IN THE ON-DEMAND ECONOMY

Analytics must be delivered rapidly in order to meet user expectations in the on-demand economy. This is simply not possible with an approach that depends upon [“hope and heroism”](#) or “caution.”

In order to deliver value consistently, quickly and accurately, data-analytics teams must learn to create and publish analytics in a new way. We call this new approach [DataOps](#). DataOps is a combination of tools and methods, which streamline the development of new analytics while ensuring impeccable [data quality](#). DataOps helps shorten the cycle time for producing analytic value and innovation, while avoiding the trap of “hope, heroism and caution.”



## Data Analytics Can Learn from Agile

If you were managing 100 software developers, you would have to choose the best way to maximize their productivity. Since the dawn of the computer era many software project management approaches have been tried. The waterfall model dominated software project management up until the 1990's. In the early days of computing, project management was adapted from the manufacturing and construction industries, which required detailed planning and a great degree of structure. Projects were organized into phases (conception, initiation, analysis, design, construction, testing, production/implementation and maintenance) and progressed through these phases sequentially. Once a phase was done, the team moved forward to the next phase.

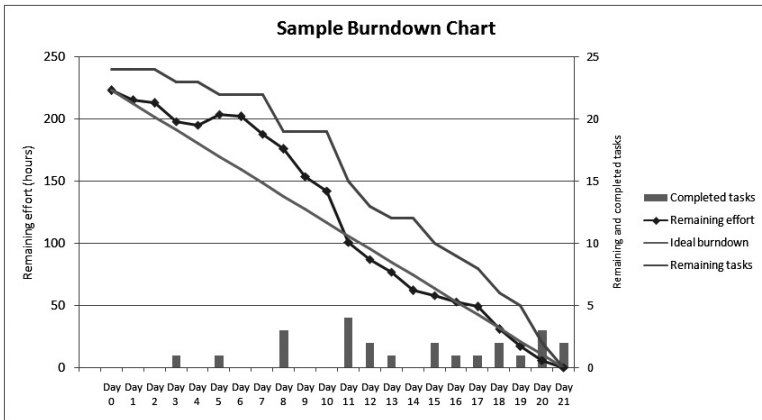


Figure 1: In Agile development, a burndown chart shows work remaining over time.

The waterfall model is better suited to situations where the requirements are fixed and well understood up front. This is nothing like the technology industry where the competitive environment evolves rapidly. In the 1980's a typical software project required about 12 calendar months. In technology-driven businesses (i.e. nearly everyone these days) customers demand new features and services, and competitive pressures change priorities on a seemingly daily basis. The waterfall model has no mechanism to respond to these changes. In waterfall, changes trigger a seemingly endless cycle of replanning causing delays and resulting in project budget overruns.

In the early 2000's, the software industry embraced a new approach to code production called [Agile Development](#). Agile is an umbrella term for several different iterative and incremental software development methodologies.

In Agile Software Development, the team and its processes and tools are organized around the goal of publishing releases to the users every few weeks (or at most every few months). A development cycle is called a *sprint* or an *iteration*. At the beginning of an iteration, the team commits to completing working and (the most) valuable changes to the code base. Features are associated with *user stories*, which help the development team

understand the context behind requirements. User stories include descriptions of features and acceptance criteria. The Agile methodology is particularly good for non-sequential product development where market requirements are quickly evolving. This is similar to the data-analytics environment where each new analysis and report of the data inspires requests for additional queries.

Agile is widely credited with boosting software productivity. One study sponsored by the Central Ohio Agile Association and Columbus Executive Agile Special Interest Group found that Agile projects were completed 31 percent faster and with a 75 percent lower defect rate than the industry norm. The vast majority of companies are getting on-board. In a survey of 400 IT professionals by [TechBeacon](#), two-thirds described their company as either “pure agile” or “leaning towards agile. Among the remaining one third of companies, most use a hybrid approach, leaving only nine percent using a pure waterfall approach.

In an increasingly competitive marketplace, Agile methods allow companies to become more responsive to customer requirements and accelerate time to market. Agile also improves ROI because features delivered in each iteration can be immediately monetized instead of waiting months for a big release. Agile is the major reason that release frequency improved from around 3 months in the 1990's to about 3 weeks in the 2000's. However, improvements didn't stop there. Today releases are occurring every few seconds using an updated approach which builds upon Agile.

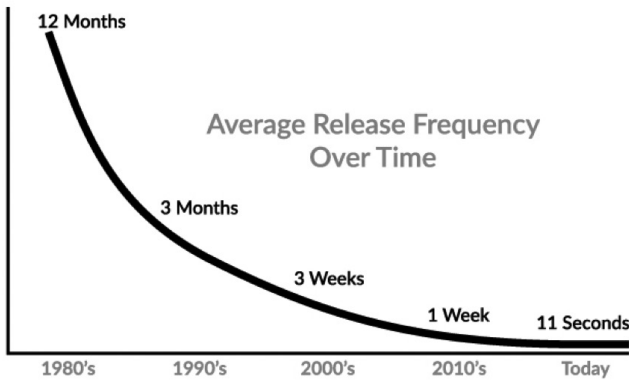


Figure 2: The decrease in release cycle time as software development evolved from waterfall to Agile to DevOps.

## Data Analytics Can Learn from DevOps

Before the advent of on-demand cloud services, the various groups in software development (design, development, test, quality, support, ...) had to set-up their own infrastructure. Whatever components were needed (physical servers, networks, storage, software, ...) had to be ordered, installed, configured and managed by the IT department. Servers would be ordered at different times and from different vendors, each slightly different from the other.

Depending on the task at hand, different machines could have a different array of software applications with revisions of each app being continuously updated. Target devices could range from embedded IoT (Internet of Things) to the largest, most powerful servers. With all of this variability, it was quite common for individuals within the company to be running code in different environments. Outside the four walls of the company, customers could be running in yet another environment. This situation presented challenges.

The software development pipeline can be organized as follows: planning, resourcing, development, testing, quality assurance, and deployment. Continuous delivery requires automation from planning all the way through to delivery/deployment. Previously, software development, testing, quality assurance, and customers could each be running in different environments. This hampered their ability to communicate and led to misunderstandings and delays.

If, for example, the customer reported a problem, it might not be replicable in the support, test or development groups due to differences in the hardware and software environments being run. This lack of alignment fostered misunderstandings and delays and often led to a lack of trust and communication between the various stakeholders.

In a complex world requiring the physical provisioning of servers, installation of stacks and frameworks, and numerous target devices, the standardization and control of the run-time environment has been difficult and slow. It became necessary to break down barriers between the respective teams in the software development pipeline. This merging of development and IT/Operations is widely known as [DevOps](#), which also has had enormous impact on the world of software development. DevOps improves collaboration between employees from the planning through the deployment phases of software. It seeks to reduce time to deployment, decrease time to market, minimize defects, and shorten the time required to fix problems.

About a decade ago, Amazon Web Services (AWS) and other cloud providers, began offering computing, storage and other IT resources as an on-demand service. No more waiting weeks or months for the IT department to fulfill a request for servers. Cloud providers now allow you to order computing services, paying only for what you use, whether that is one processor for an hour or thousands of processors for months. These on-demand cloud services have enabled developers to write code that provisions processing resources with strictly specified environments, on-demand, in just a few minutes. This capability has been called *Infrastructure as Code (IaC)*. IaC has made it possible for everyone in the software development pipeline, all the different groups mentioned above, to use an identical environment tailored to application requirements. With IaC, design, test, QA and support



can easily get *on the same page*. This leads to much better collaboration between the groups and breaks down barriers that prevented open communication. In other words, no more finger pointing.

With IT infrastructure being defined by code, the hard divisions between IT operations and software development are able to blur. The merger of development and operations is how the term *DevOps* originated.

With the automated provisioning of resources, DevOps paved the way for a fully automated test and release process. The process of deploying code that used to take weeks, could now be completed in minutes. Major organizations including Amazon, Facebook and Netflix are now operating this way. At a recent conference, Amazon disclosed that their AWS team performs [50,000,000 code releases per year](#). This is more than one per second! This methodology of rapid releases is called [continuous delivery](#) or alternatively, *continuous deployment*, when new features (and fixes) are not only delivered internally but fully deployed to customers.

DevOps starts with continuous delivery and Agile development and adds automated provisioning of resources (infrastructure as code) and cloud services (platform as a service) to ensure that the same environment is being utilized at every stage of the software development pipeline. The cloud provides a natural platform that allows individuals to create and define identical run-time environments. DevOps is beginning to achieve critical mass in terms of its adoption within the world of software development.

DevOps improves collaboration between employees from the planning through the deployment phases of software. It seeks to reduce time to deployment, decrease time to market, minimize defects, and shorten the time required to fix problems.

The impact of DevOps on development organizations was shown in a 2014 [survey](#), “The 2014 State of DevOps Report” by Puppet Labs, IT Revolution Press and ThoughtWorks, based on 9,200 survey responses from technical professionals. The survey found that IT organizations implementing DevOps were deploying code 30 times more frequently and with 50 percent fewer failures. Further, companies with these higher performing IT organizations tended to have stronger business performance, greater productivity, higher profitability and larger market share. In other words, DevOps is not just something that engineers are doing off in a dark corner. It is a core competency that helps good companies become better.

The Data analytics team transforms raw data into actionable information that improves decision making and provides market insight. Imagine an organization with the best data analytics in the industry. That organization would have a tremendous advantage over competitors. That could be you.

## **Data Analytics Can Learn from Manufacturing**

What could data analytics professionals possibly learn from industrial manufacturers? It turns out, a lot. Automotive giant [Toyota](#) pioneered a set of methods, later folded into a discipline called [lean manufacturing](#), in which employees focus relentlessly on improving quality and reducing non-value-add activities. This culture enabled Toyota to grow into the one of the world’s leading car companies. The Agile and DevOps methods that have led to stellar improvements in coding velocity are really just an example of lean manufacturing principles applied to software development.



Conceptually, manufacturing is a pipeline process. Raw materials enter the manufacturing floor through the stock room, flow to different work stations as work-in-progress and exit as finished goods. In data-analytics, data progresses through a series of steps and exits in the form of reports, models and visualizations. Each step takes an input from the previous step, executes a complex procedure or set of instructions and creates output for the subsequent step. At an abstract level, the data-analytics pipeline is analogous to a manufacturing process. Like manufacturing, data analytics executes a set of operations and attempts to produce a consistent output at a high level of quality. In addition to lean-manufacturing-inspired methods like Agile and DevOps, there is one more useful tool that can be taken from manufacturing and applied to data-analytics process improvement.

W. Edwards Deming championed [\*statistical process control\*](#) (SPC) as a method to improve manufacturing quality. SPC uses real-time product or process measurements to monitor and control quality during manufacturing processes. If the process measurements are maintained within specific limits, then the manufacturing process is deemed to be functioning properly. When SPC is applied to the data-analytics pipeline, it leads to remarkable improvements in efficiency and quality. For example, Google executes over one hundred million automated test scripts per day to validate any new code released by software developers. In the Google consumer surveys group, code is deployed to customers eight minutes after a software engineer finishes writing and testing it.

In data analytics, tests should verify that the results of each intermediate step in the production of analytics matches expectations. Even very simple tests can be useful. For example, a simple row-count test could catch an error in a join that inadvertently produces a Cartesian product. Tests can also detect unexpected trends in data, which might be flagged as warnings. Imagine that the number of customer transactions exceeds its historical average by 50%. Perhaps that is an anomaly that upon investigation would lead to insight about business seasonality.

Tests in data analytics can be applied to data or models either at the input or output of a phase in the analytics pipeline. Tests can also verify business logic.

**Business logic** tests validate assumptions about the data. For example:

- Customer Validation – Each customer should exist in a dimension table
- Data Validation – At least 90 percent of data should match entries in a dimension table

**Input** tests check data prior to each stage in the analytics pipeline. For example:

- Count Verification – Check that row counts are in the right range, ...
- Conformity – US Zip5 codes are five digits, US phone numbers are 10 digits, ...
- History – The number of prospects always increases, ...
- Balance – Week over week, sales should not vary by more than 10%, ...
- Temporal Consistency – Transaction dates are in the past, end dates are later than start dates, ...
- Application Consistency – Body temperature is within a range around 98.6F/37C, ...
- Field Validation – All required fields are present, correctly entered, ...

**Output** tests check the results of an operation, like a Cartesian join. For example:

- Completeness – Number of customer prospects should increase with time
- Range Verification – Number of physicians in the US is less than 1.5 million

The data analytics pipeline is a complex process with steps often too numerous to be monitored manually. SPC allows the data analytics team to monitor the pipeline end-to-end from a big-picture perspective, ensuring that everything is operating as expected. As an automated test suite grows and matures, the quality of the analytics is assured without adding cost. This makes it possible for the data analytics team to move quickly – enhancing analytics to address new challenges and queries – without sacrificing quality.

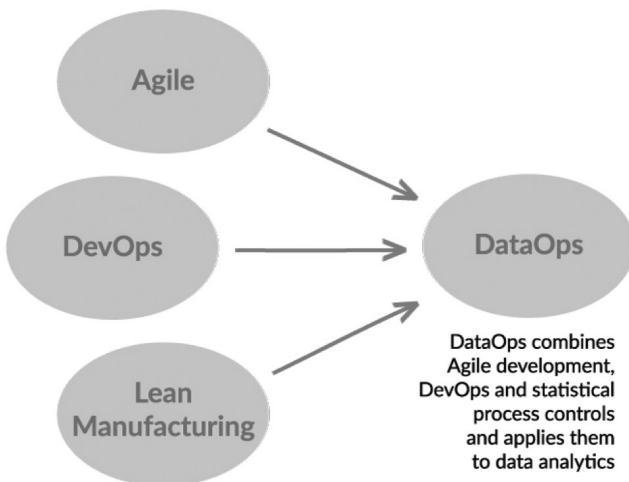


Figure 3: DataOps has evolved from lean manufacturing and software methodologies.

## DataOps for Data Analytics

The speed and flexibility achieved by Agile and [DevOps](#), and the quality control attained by SPC, can be applied to data analytics. Leading edge proponents of this approach are calling it [DataOps](#). DataOps, simply stated, is Agile development and DevOps with [statistical process control](#), for data analytics. DataOps applies Agile methods, DevOps, and manufacturing quality principles, methodologies and tools, to the data-analytics pipeline. The result is a rapid-response, flexible and robust data-analytics capability, which is able to keep up with the creativity of internal stakeholders and users.

DataOps is an analytic development method that emphasizes communication, collaboration, integration, automation, measurement and cooperation between [data scientists](#), analysts, data/ETL (extract, transform, load) engineers, information technology (IT), and quality assurance/governance. The method acknowledges the interdependence of the entire end-to-end analytic process. It aims to help organizations rapidly produce insight, turn that insight into operational tools, and continuously improve analytic operations and performance. It enables the whole analytic team involved in the analytic process to follow the values laid out in the [DataOps Manifesto](#).

When DataOps is implemented correctly, it addresses many of the issues discussed earlier that have plagued data-analytics teams.

| Challenge             | DataOps Approach  |
|-----------------------|---|
| Changing Requirements | The team delivers something of value to users at each iteration. If the requirements change, it is simple to put new requirements on the request list for a future iteration. |
| Slipped schedules     | Iterations occur in rapid succession allowing greater visibility to the progress that is being made. As a team gains experience with DataOps, their forecasting improves.     |
| Disappointed users    | Users receive new features quickly and give feedback to the development team about how to keep improving the data analytics with even more new features.                      |
| Inflexibility         | DataOps enables teams to respond quickly to change. The team can pivot at the beginning of the next iteration, which by definition is always relatively soon.                 |
| Poor quality          | Extensive, automated testing, in the form of statistical process control, is a key element in DataOps.  |
| Low ROI               | With features being delivered in short increments, the monetization of the data-analytics investment begins much earlier, improving ROI.                                      |
| Irrelevant features   | The data-analytics team is churning out management's highest priority features in quick succession.   |

Table 1: Challenge/DataOps Approach

DataOps views the data-analytics pipeline as a process and as such focuses on how to make the entire process run more rapidly and with higher quality, rather than optimizing the productivity of any single individual or tool by itself.

## Key Benefits of DataOps

[DataOps](#) can accelerate the ability of data-analytics teams to create and publish new analytics to users. It requires an Agile mindset and must also be supported by an automated platform which incorporates existing tools into a DataOps development pipeline. DataOps spans the entire analytic process, from data acquisition to insight delivery. Its goal is to achieve more insight and better analysis, while still being faster, cheaper and higher quality.

The key business benefits of adopting DataOps are:

- Reduce time to insight
- Improve analytic quality
- Lower the marginal cost to ask the next business question
- Improve analytic team morale by going beyond hope, heroism and caution
- Promote team efficiency through agile process, reuse and refactoring

[DataKitchen](#) markets an automated DataOps platform that helps companies accelerate their DataOps implementation, but this book is about DataOps not us. This book is not trying to sell you anything. You can implement DataOps all by yourself, using your existing tools, by implementing the [seven steps](#) described in the next section. If you desire assistance, there is an [ecosystem](#) of DataOps vendors who offer a variety of innovative solutions and services.

# The Seven Steps to Implement DataOps

Data analytics has become business critical, but requirements quickly evolve and data-analytics teams that respond to these challenges in the traditional ways often end up facing disappointed users. [DataOps](#) offers a more effective approach that optimizes the productivity of the data analytics pipeline by an order of magnitude.

Imagine the next time that the Vice President of Marketing requests a new customer segmentation, by tomorrow. With DataOps, the data-analytics team can respond 'yes' with complete confidence that the changes can be accomplished quickly, efficiently and robustly. How then does an organization implement DataOps? You may be surprised to learn that an analytics team can migrate to DataOps in seven simple steps.

## STEP 1 - ADD DATA AND LOGIC TESTS

If you make a change to an analytic pipeline, how do you know that you did not break anything? Automated testing insures that a feature release is of high quality without requiring time-consuming, manual testing. The idea in DataOps is that every time a data-analytics team member makes a change, he or she adds a test for that change. Testing is added incrementally, with the addition of each feature, so testing gradually improves and quality is literally built in. In a big run, there could be hundreds of tests at each stage in the pipeline.

Adding tests in data analytics is analogous to the [statistical process control](#) that is implemented in a manufacturing operations flow. Tests insure the integrity of the final output by verifying that work-in-progress (the results of intermediate steps in the pipeline) matches expectations. Testing can be applied to data, models and logic. The figure below shows examples of tests in the data-analytics pipeline.

For every step in the data-analytics pipeline, there should be at least one test. The philosophy is to start with simple tests and grow over time. Even a simple test will eventually catch an error before it is released out to the users. For example, just making sure that row counts are consistent throughout the process can be a very powerful test. One could easily make a mistake on a *join* and make a *cross product* which fails to execute correctly. A simple row-count test would quickly catch that.

Tests can detect warnings in addition to errors. A warning might be triggered if data exceeds certain boundaries. For example, the number of customer transactions in a week may be OK if it is within 90% of its historical average. If the transaction level exceeds that, then a warning could be flagged. This might not be an error. It could be a seasonal occurrence for example, but the reason would require investigation. Once recognized and understood, the users of the data could be alerted.

DataOps is not about being perfect. In fact, it acknowledges that code is imperfect. It's natural that a data-analytics team will make a best effort, yet still miss something. If so, they can determine the cause of the issue and add a test so that it never happens again. In a rapid release environment, a fix can quickly propagate out to the users.

With a suite of tests in place, [DataOps](#) allows you to move fast because you can make changes and quickly rerun the test suite. If the changes pass the tests, then the data-analytics team member can be confident and release it. The knowledge is built into the system and the process stays under control. Tests catch potential errors and warnings before they are released so the quality remains high.

Automated tests continuously monitor the data pipeline for errors and anomalies. They work nights, weekends and holidays without taking a break. If you build a DataOps dashboard, you can view the high-level state of your data operations at any time. If warning and failure alerts are automated, you don't have to constantly check your dashboard. Automated testing frees the data-analytics team from the drudgery of manual testing, so they can focus on higher value-add activities.



Figure 4: Tests enable the data professional to apply statistical process controls to the data pipeline

## STEP 2 - USE A VERSION CONTROL SYSTEM

There are many processing steps that turn raw data into useful information for stakeholders. To be valuable, data must progress through these steps, linked together in some way, with the ultimate goal of producing a data-analytics output. Data may be preprocessed, cleaned, checked, transformed, combined, analyzed, and reported. Conceptually, the data-analysis pipeline is a set of stages implemented using a variety of tools including ETL tools, data science tools, self-service data prep tools, reporting tools, visualization tools and more. The stages may be executed serially, but many stages can be parallelized. The pipeline is deterministic because the pipeline stages are defined by scripts, source code, algorithms, html, configuration files, parameter files, containers and other files. All of these items are *essentially* just code. Code controls the entire data-analytics pipeline from end to end in a reproducible fashion.

The artifacts (files) that make this reproducibility possible are usually subject to continuous improvement. Like other software projects, the source files associated with the data pipeline should be maintained in a version control (source control) system such as [Git](#). A version control tool helps teams of individuals organize and manage the changes and revisions to code. It also keeps code in a known repository and facilitates disaster recovery. However, the most important benefit of version control relates to a process change that it facilitates. It allows data-analytics team members to *branch and merge*.

```

    'send_welcome' => false,
  });

  (function(error, result) {
    var result = array ('response'=>'error', 'message'
    = {
    var result = array ('response'=>'success');
    var encode(result);
  }

```

### STEP 3 - BRANCH AND MERGE

In a typical software project, developers are continuously updating various code source files. If a developer wants to work on a feature, he or she pulls a copy of all relevant code from the version control tool and starts to develop changes on a local copy. This local copy is called a *branch*. This approach can help data-analytics teams maintain many coding changes to the data-analytics pipeline in parallel. When the changes to a branch are complete and tested, the code from the branch is *merged* back into the trunk, where the code came from.

Branching and merging can be a major productivity boost for data analytics because it allows teams to make changes to the same source code files in parallel without slowing each other down. Each individual team member has control of his or her work environment. They can run their own tests, make changes, take risks and experiment. If they wish, they can discard their changes and start over. Another key to allowing team members to work well in parallel relates to providing them with an isolated machine environment.

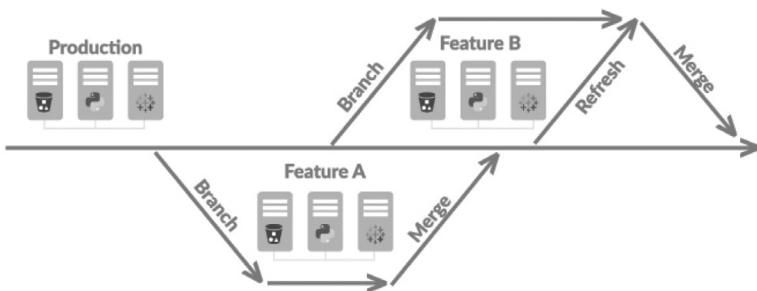
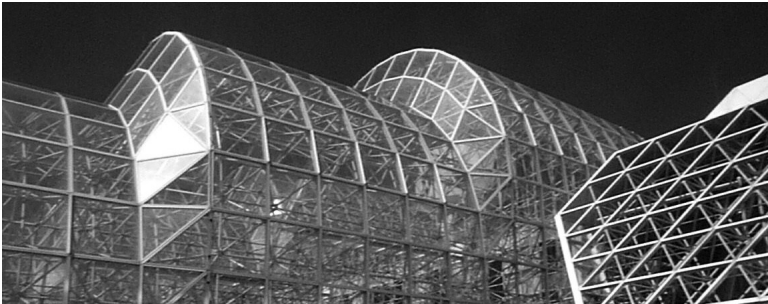


Figure 5: Branching and merging enables parallel development in data analytics.

## STEP 4 - USE MULTIPLE ENVIRONMENTS

Every data-analytics team member has their own development tools on their own laptop. Version control tools allow team members to work on their own private copy of the source code while still staying coordinated with the rest of the team. In data analytics, a team member can't be productive unless they also have a copy of the data that they need. Most use cases can be covered in less than a Terabyte (TB). Historically, disk space has been prohibitively expensive, but today, at less than \$25 per TB per month (cloud storage), costs are now less significant than the opportunity cost of a team member's time. If the data set is still too large, then a team member can take only the subset of data that is needed. Often the team member only needs a representative copy of the data for testing or developing one set of features.

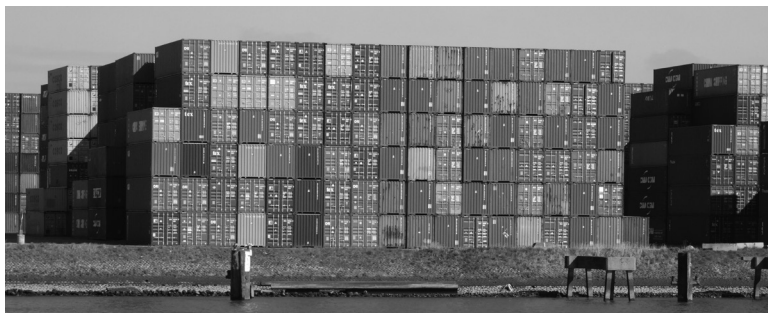
When many team members work on the production database, it can lead to conflicts. A database engineer changing a schema may break reports. A [data scientist](#) developing a new model might get confused as new data flows in. Giving team members their own *Environment* isolates the rest of the organization from being impacted by their work.



## STEP 5 - REUSE & CONTAINERIZE

Another productivity boosting method for teams is the ability to reuse and containerize code. Each middle step in the data-analytics pipeline receives output from a prior stage and provides input to the next stage. It is cumbersome to work with an entire data-analytics pipeline as one monolith, so it is common to break it down into smaller components. It's easiest for other team members to reuse smaller components if they can be segmented or containerized. One popular container technology is [Docker](#).

Some steps in the data-analytics pipeline are messy and complicated. For example, one operation might call a custom tool, run a python script, use FTP and other specialized logic. This operation might be hard to set up (because it requires a specific set of tools) and difficult to create (because it requires a specific skill set). This scenario is another common use case for creating a container. Once the code is placed in a container, it is much easier to use by other programmers who aren't familiar with the custom tools inside the container but know how to use the container's external interfaces. It is also easier to deploy that code to each environment.



## STEP 6 - PARAMETERIZE YOUR PROCESSING

There are cases when the data-analytic pipeline needs to be flexible enough to incorporate different run-time conditions. Which version of the raw data should be used? Is the data directed to production or testing? Should records be filtered according to some criterion (such as private health care data)? Should a specific set of processing steps in the workflow be included or not? To increase development velocity, these options need to be built into the pipeline. A robust pipeline design will allow the engineer or analyst to invoke or specify these options using parameters. In software development, a parameter is some information (e.g. a name, a number, an option) that is passed to a program that affects the way that it executes. If the data-analytic pipeline is designed with the right flexibility, it will be ready to accommodate different run-time circumstances.

For example, imagine a pharmaceutical company that obtains prescription data from a 3rd party company. The data is incomplete, so the data producer uses algorithms to fill in those gaps. In the course of improving their product, the data producer develops a different algorithm to fill in the gaps. The data has the same shape (rows and columns), but certain fields are modified using the new algorithm. With the correct built-in parameters, an engineer or analyst can easily build a parallel data mart with the new algorithm and have both the old and new versions accessible through a parameter change.



## STEP 7: WORK WITHOUT FEAR OR HEROISM

Many data analytics professionals live in fear. In data analytics there are two common ways to be professionally embarrassed (or get fired):

- Allow poor quality data to reach users
- Deploy changes that break production systems

[Data engineers](#), scientists and analysts spend an excessive amount of time and energy working to avoid these disastrous scenarios. They attempt “heroism” — working weekends. They do a lot of hoping and praying. They devise creative ways to avoid overcommitting. The problem is that heroic efforts are eventually overcome by circumstances. Without the right controls in place, a problem will slip through and bring the company’s critical analytics to a halt.

The [DataOps](#) enterprise puts the right set of tools and processes in place to enable data and new analytics to be deployed with a high level of quality. When an organization implements DataOps, engineers, scientists and analysts can relax because quality is assured. They can *Work Without Fear or Heroism*. DataOps accomplishes this by optimizing two key workflows.

### The Value Pipeline

Data analytics seeks to extract value from data. We call this the Value Pipeline. The diagram below shows the Value Pipeline progressing horizontally from left to right. Data enters the pipeline and moves into production processing. Production is generally a series of stages: access, transforms, models, visualizations, and reports. When data exits the pipeline, in the form of useful analytics, value is created for the organization. DataOps utilizes toolchain workflow automation to optimize operational efficiency in the Value Pipeline. Data in the Value Pipeline is updated on a continuous basis, but code is kept constant. Step 2 in the [seven steps](#) of implementing DataOps — using [version control](#) — serves as the foundation for controlling the code deployed.

As mentioned above, the worst possible outcome is for poor quality data to enter the Value Pipeline. DataOps prevents this by implementing data tests (step 1). Inspired by the [statistical process control](#) in a manufacturing workflow, data tests ensure that data values lay within an acceptable statistical range. Data tests validate data values at the inputs and outputs of each processing stage in the pipeline. For example, a US phone number should be ten digits. Any other value is incorrect or requires normalization.

Once data tests are in place, they work 24x7 to guarantee the integrity of the Value Pipeline. Quality becomes *literally* built in. If anomalous data flows through the pipeline, the data tests catch it and take action — in most cases this means firing off an alert to the data analytics team who can then investigate. The tests can even, in the spirit of auto manufacturing, “stop the line.” Statistical process control eliminates the need to worry about what might happen. With the right data tests in place, the data analytics team can *Work Without Fear or Heroism*. This frees DataOps engineers to focus on their other major responsibility — the Innovation Pipeline.

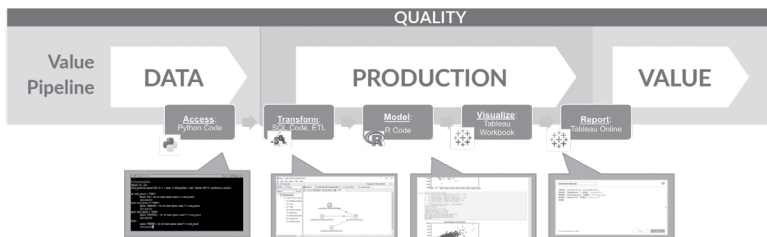


Figure 6: The value pipeline

## The Innovation Pipeline

The Innovation Pipeline seeks to improve analytics by implementing new ideas that yield analytic insights. As the diagram illustrates, a new feature undergoes development before it can be deployed to production systems. The Innovation Pipeline creates a feedback loop. Innovation spurs new questions and ideas for enhanced analytics. This requires more development leading to additional insight and innovation. During the development of new features, code changes, but data is kept constant. Keeping data static prevents changes in data from being falsely attributed to the impact of the new algorithm. A fixed data set can be set-up when creating a [development environment](#) — step 4 in the [seven steps](#) of implementing [DataOps](#).

DataOps implements continuous deployment of new ideas by automating the workflow for building and deploying new analytics. It reduces the overall cycle time of turning ideas into innovation. While doing this, the development team must avoid introducing new analytics that break production. The DataOps enterprise uses logic tests (step 1) to validate new code before it is deployed. Logic tests ensure that data matches business assumptions. For example, a field that identifies a customer should match an existing entry in a customer dimension table. A mismatch should trigger some type of follow-up.

With logic tests in place, the development pipeline can be automated for continuous deployment, simplifying the release of new enhancements and enabling the data analytics team to focus on the next valuable feature. With DataOps the dev team can deploy without worrying about breaking the production systems — they can [Work Without Fear or Heroism](#). This is a key characteristic of a fulfilled, productive team.

## The Value-Innovation Pipeline

In real world data analytics, the [Value Pipeline](#) and Innovation Pipeline are not separate. The same team is often responsible for both. The same assets are leveraged. Events in one affect the other. The two workflows are shown combined into the Value-Innovation Pipeline in the figure below. The Value-Innovation Pipeline captures the interplay between development and production and between data and code. DataOps breaks down this

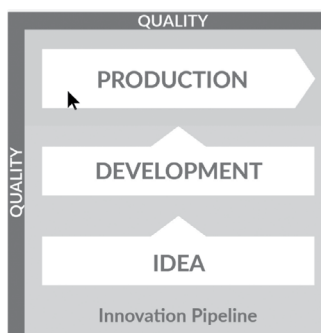
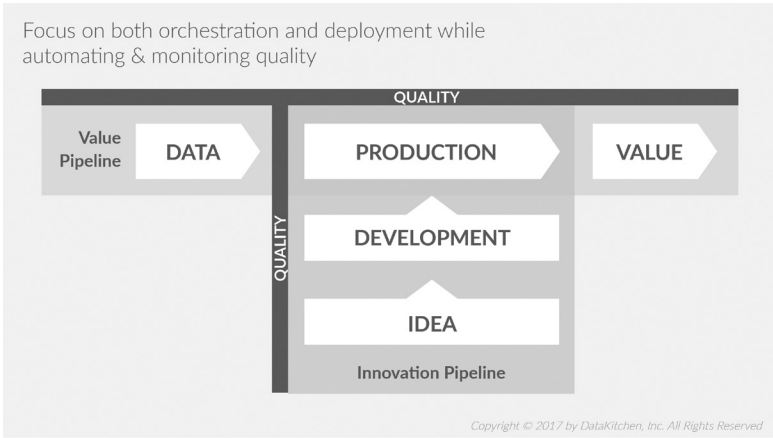


Figure 7: The innovation pipeline

barrier so that cycle time, quality and creativity can all be improved. The DataOps enterprise masters the orchestration of data to production and the deployment of new features both while maintaining impeccable quality. Reduced cycle time enables DataOps engineers to impact the organization in highly visible ways. Improved quality enables the team to move forward with confidence. DataOps speeds the extraction of value from data and improves the velocity of new development while ensuring the quality of data and code in production systems. With confidence in the Value-Innovation pipeline that stems from DataOps, the data analytics team avoids the anxiety and over-caution that characterizes a non-DataOps enterprise. [Work Without Fear or Heroism!](#)



**Figure 8: The Value and Innovation Pipelines illustrate how new analytics are introduced into data operations.**

# DataOps is NOT Just DevOps for Data

One common misconception about [DataOps](#) is that it is just [DevOps](#) applied to [data analytics](#). While a little semantically misleading, the name “DataOps” has one positive attribute. It communicates that data analytics can achieve what software development attained with DevOps. That is to say, DataOps can yield an order of magnitude improvement in quality and cycle time when data teams utilize new tools and methodologies. The specific ways that DataOps achieves these gains reflect the unique people, processes and tools characteristic of data teams (versus software development teams using DevOps). Here’s our in-depth take on both the pronounced and subtle differences between DataOps and DevOps.

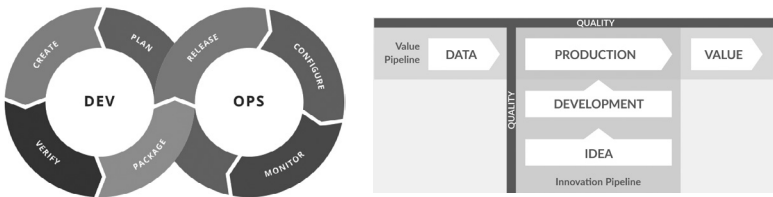


Figure 9: DevOps is often depicted as an infinite loop, while DataOps is illustrated as intersecting Value and Innovation Pipelines

## The Intellectual Heritage of DataOps

DevOps is an approach to software development that accelerates the build lifecycle (formerly known as release engineering) using automation. DevOps focuses on [continuous integration](#) and [continuous delivery](#) of software by leveraging on-demand IT resources (infrastructure as code) and by automating integration, test and deployment of code. This [merging](#) of software development and IT operations (“DEvelopment” and “OPerationS”) reduces time to deployment, decreases time to market, minimizes defects, and shortens the time required to resolve issues.

Using DevOps, leading companies have been able to reduce their software release cycle time from months to (literally) seconds. This has enabled them to grow and lead in fast-paced, emerging markets. Companies like Google, Amazon and many others now release software many times per day. By improving the quality and cycle time of code releases, DevOps deserves a lot of credit for these companies’ success.

Optimizing code builds and delivery is only one piece of the larger puzzle for data analytics. DataOps seeks to reduce the end-to-end cycle time of data analytics, from the origin of ideas to the literal creation of charts, graphs and models that create value. The data lifecycle relies upon people in addition to tools. For DataOps to be effective, it must manage collaboration and innovation. To this end, DataOps introduces Agile Development into data analytics so that data teams and users work together more efficiently and effectively.

In [Agile Development](#), the data team publishes new or updated analytics in short increments called “sprints.” With innovation occurring in rapid intervals, the team can continuously reassess its priorities and more easily adapt to evolving requirements. This type of responsiveness is impossible using a [Waterfall project management](#) methodology which locks a team into a long development cycle with one “big-bang” deliverable at the end.

Studies show that Agile software development projects complete faster and with fewer defects when Agile Development replaces the traditional Waterfall sequential methodology. The Agile methodology is particularly effective in environments where requirements are quickly evolving — a situation well known to data analytics professionals. In a DataOps setting, Agile methods enable organizations to respond quickly to customer requirements and accelerate time to value.

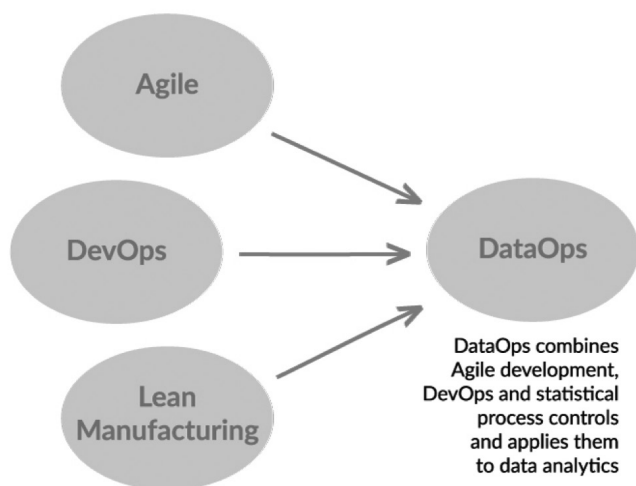


Figure 10: DataOps has evolved from lean manufacturing and software methodologies.

Agile development and DevOps add significant value to data analytics, but there is one more major component to DataOps. Whereas Agile and DevOps relate to analytics development and deployment, data analytics also manages and orchestrates a data pipeline. Data continuously enters on one side of the pipeline, progresses through a series of steps and exits in the form of reports, models and views. The data pipeline is the “operations” side of data analytics. It is helpful to conceptualize the data pipeline as a manufacturing line where quality, efficiency, constraints and uptime must be managed. To fully embrace this manufacturing mindset, we call this pipeline the “data factory.”

In DataOps, the flow of data through operations is an important area of focus. DataOps orchestrates, monitors and manages the data factory. One particularly powerful [lean-manufacturing](#) tool is [statistical process control](#) (SPC). SPC measures and monitors data and

operational characteristics of the data pipeline, ensuring that statistics remain within acceptable ranges. When SPC is applied to data analytics, it leads to remarkable improvements in efficiency, quality and transparency. With SPC in place, the data flowing through the operational system is verified to be working. If an anomaly occurs, the data analytics team will be the first to know, through an automated alert.

While the name “DataOps” implies that it borrows most heavily from DevOps, it is all three of these methodologies - Agile, DevOps and statistical process control — that comprise the intellectual heritage of DataOps. Agile governs analytics development, DevOps optimizes code verification, builds and delivery of new analytics and SPC orchestrates and monitors the data factory. Figure 10 illustrates how Agile, DevOps and statistical process control flow into DataOps.

You can view DataOps in the context of a century-long evolution of ideas that improve how people manage complex systems. It started with pioneers like [Deming](#) and statistical process control — gradually these ideas crossed into the technology space in the form of Agile, DevOps and now, DataOps.

### DevOps vs. DataOps — The Human Factor

As mentioned above, [DataOps](#) is as much about managing people as it is about tools. One subtle difference between DataOps and [DevOps](#) relates to the needs and preferences of stakeholders.

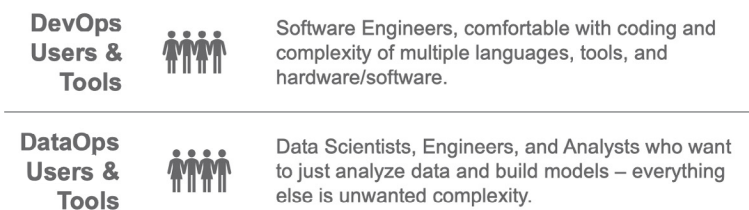


Figure 11: DataOps and DevOps users have different mindsets

DevOps was created to serve the needs of software developers. Dev engineers love coding and embrace technology. The requirement to learn a new language or deploy a new tool is an opportunity, not a hassle. They take a professional interest in all the minute details of code creation, integration and deployment. DevOps embraces complexity.

DataOps users are often the opposite of that. They are [data scientists](#) or analysts who are focused on building and deploying models and visualizations. Scientists and analysts are typically not as technically savvy as engineers. They focus on domain expertise. They are interested in getting models to be more predictive or deciding how to best visually render data. The technology used to create these models and visualizations is just a means to an end. Data professionals are happiest using one or two tools — anything beyond that adds unwelcome complexity. In extreme cases, the complexity grows beyond their ability to manage it. DataOps accepts that data professionals live in a multi-tool, heterogeneous world and it seeks to make that world more manageable for them.

## DevOps vs. DataOps - Process Differences

We can begin to understand the unique complexity facing data professionals by looking at data analytics development and lifecycle processes. We find that data analytics professionals face challenges both similar and unique relative to software developers.

The [DevOps](#) lifecycle is commonly illustrated using a diagram in the shape of an infinite symbol — See Figure 12. The end of the cycle (“plan”) feeds back to the beginning (“create”), and the process iterates indefinitely.

The [DataOps](#) lifecycle shares these iterative properties, but an important difference is that DataOps consists of two active and intersecting pipelines (Figure 13). The data factory, described above, is one pipeline. The other pipeline governs how the data factory is updated — the creation and deployment of new analytics into the data pipeline.

The data factory takes raw data sources as input and through a series of orchestrated steps produces analytic insights that create “value” for the organization. We call this the “[Value Pipeline](#).” DataOps automates orchestration and, using SPC, monitors the quality of data flowing through the Value Pipeline.

The “[Innovation Pipeline](#)” is the process by which new analytic ideas are introduced into the Value Pipeline. The Innovation Pipeline conceptually resembles a DevOps development process, but upon closer examination, several factors make the DataOps development process more challenging than DevOps. Figure 13 shows a simplified view of the Value and Innovation Pipelines.

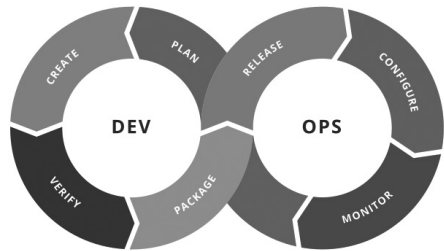


Figure 12: The DevOps lifecycle is often depicted as an infinite loop

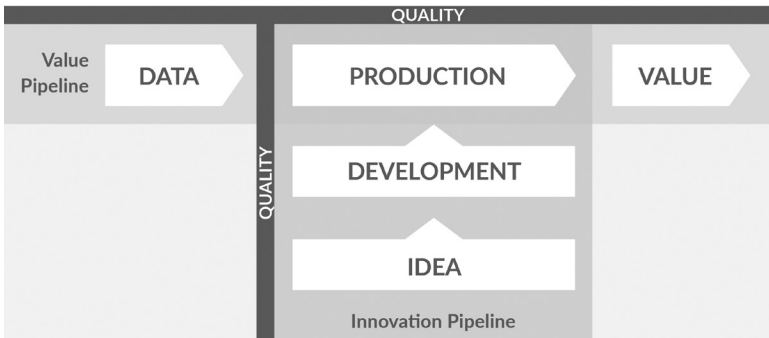


Figure 13: The DataOps lifecycle — the Value and Innovation Pipelines

## DevOps vs. DataOps — Development and Deployment Processes

[DataOps](#) builds upon the [DevOps](#) development model. As shown in Figure 14, the DevOps process flow includes a series of steps that are common to software development projects:

- **Develop** – create/modify an application
- **Build** – assemble application components
- **Test** – verify the application in a test environment
- **Deploy** – transition code into production
- **Run** – execute the application

DevOps introduces two foundational concepts: [Continuous Integration](#) (CI) and [Continuous Deployment](#) (CD). CI continuously builds, integrates and tests new code in a development environment. Build and test are automated so they can occur rapidly and repeatedly. This allows issues to be identified and resolved quickly. Figure 14 illustrates how CI encompasses the build and test process stages of DevOps.

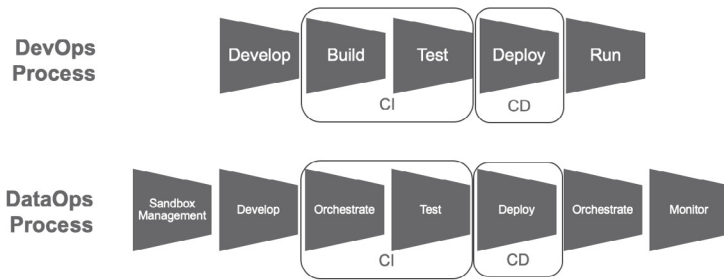


Figure 14: Comparing the DataOps and DevOps processes

CD is an automated approach to deploying or delivering software. Once an application passes all qualification tests, DevOps deploys it into production. Together CI and CD resolve the main constraint hampering [Agile development](#). Before DevOps, Agile created a rapid succession of updates and innovations that would stall in a manual integration and deployment process. With automated CI and CD, DevOps has enabled companies to update their software many times per day.

### The Duality of Orchestration in DataOps

It's important to note that “orchestration” occurs twice in the DataOps process shown in Figure 14. As we explained above, DataOps orchestrates the data factory (the [Value Pipeline](#)). The data factory consists of a pipeline process with many steps. Imagine a complex [directed acyclic graph](#) (DAG). The “orchestrator” could be a software entity which controls the execution of the steps, traverses the DAG, and handles exceptions. For example, the orchestrator might create [containers](#), invoke runtime processes with context-sensitive [parameters](#), transfer data from stage to stage, and “monitor” pipeline execution. Orchestration of the data factory is the second “orchestration” in the DataOps process in Figure 15.

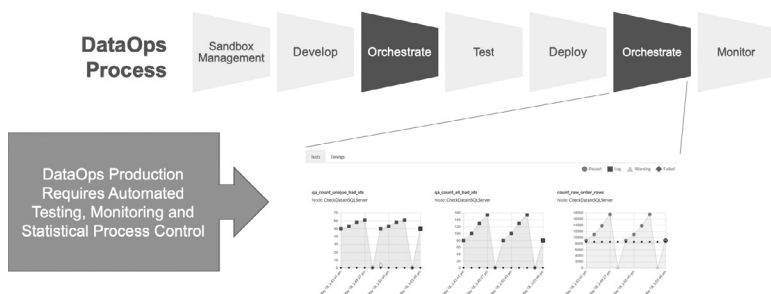


Figure 15: DataOps orchestrates the data factory.

As noted above, the [Innovation Pipeline](#) has a representative copy of the data pipeline which is used to [test](#) and verify new analytics before deployment into production. This is the orchestration that occurs in conjunction with “testing” and prior to “deployment” of new analytics — as shown in Figure 16.

Orchestration occurs in both the Value and Innovation Pipelines. Similarly, testing fulfills a dual role in DataOps.

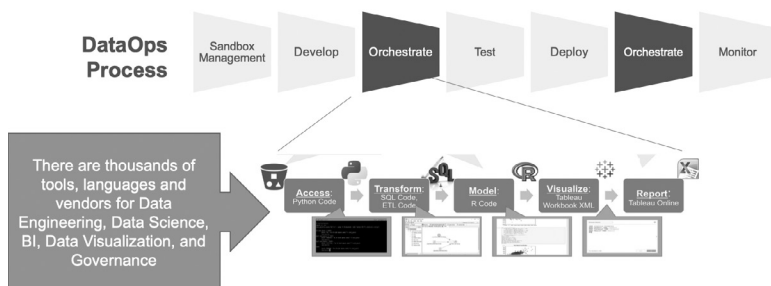


Figure 16: DataOps orchestration controls the numerous tools that access, transform, model, visualize and report data

## The Duality of Testing in DataOps

Tests in [DataOps](#) have a role in both the Value and Innovation Pipelines. In the [Value Pipeline](#), tests monitor the data values flowing through the data factory to catch anomalies or flag data values outside statistical norms. In the Innovation Pipeline, tests validate new analytics before deploying them.

In DataOps, tests target either data or code. Figure 17 below illustrates this concept. Data that flows through the Value Pipeline is variable and subject to [statistical process control](#) and monitoring. Tests target the data which is continuously changing. Analytics in the Value Pipeline, on the other hand, are fixed and change only using a formal release process. In the Value Pipeline, analytics are revision controlled to minimize any disruptions in service that could affect the data factory.

In the [Innovation Pipeline](#) code is variable and data is fixed. The analytics are revised and updated until complete. Once the sandbox (analytics [development environment](#)) is set-up, the data doesn't usually change. In the Innovation Pipeline, tests target the code (analytics), not the data. All tests must pass before promoting ([merging](#)) new code into production. A good test suite serves as an automated form of [impact analysis](#) that runs on any and every code change before deployment.

Some tests are aimed at both data and code. For example, a test that makes sure that a database has the right number of rows helps your data and code work together. Ultimately both data tests and code tests need to come together in an integrated pipeline as shown in Figure 13. DataOps enables code and data tests to work together so all around quality remains high.

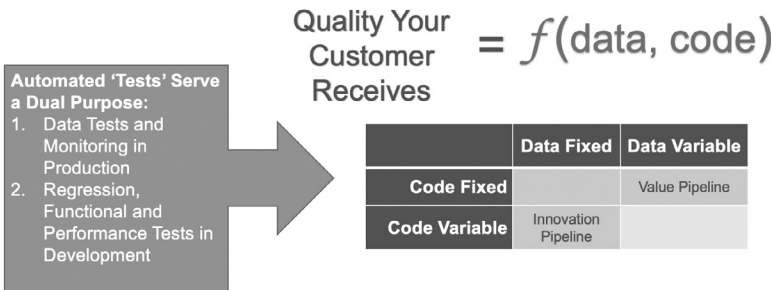


Figure 17: In DataOps, analytics quality is a function of data and code testing

### DataOps Complexity — Sandbox Management

When an engineer joins a software development team, one of their first steps is to create a "sandbox." A sandbox is an isolated development environment where the engineer can write and test new application features, without impacting teammates who are developing other features in parallel. Sandbox creation in software development is typically straightforward — the engineer usually receives a bunch of scripts from teammates and can configure a sandbox in a day or two. This is the typical mindset of a team using [DevOps](#).

Sandboxes in data analytics are often more challenging from a tools and data perspective. First of all, data teams collectively tend to use many more tools than typical software dev teams. There are literally thousands of tools, languages and vendors for [data engineering](#), data science, BI, data visualization, and governance. Without the centralization that is characteristic of most software development teams, data teams tend to naturally diverge with different tools and data islands scattered across the enterprise.

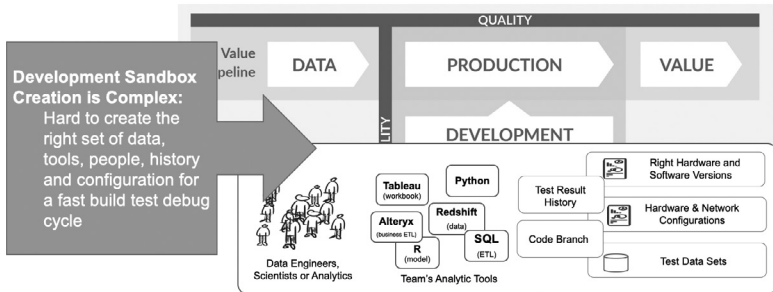


Figure 18: A “sandbox” is an isolated development environment where the data professional can write and test new analytics without impacting teammates.

### DataOps Complexity - Test Data Management

In order to create a dev environment for analytics, you have to create a copy of the data factory. This requires the data professional to replicate data which may have security, governance or licensing restrictions. It may be impractical or expensive to copy the entire data set, so some thought and care is required to construct a representative data set. Once a multi-terabyte data set is sampled or filtered, it may have to be cleaned or redacted (have sensitive information removed). The data also requires infrastructure which may not be easy to replicate due to technical obstacles or license restrictions.

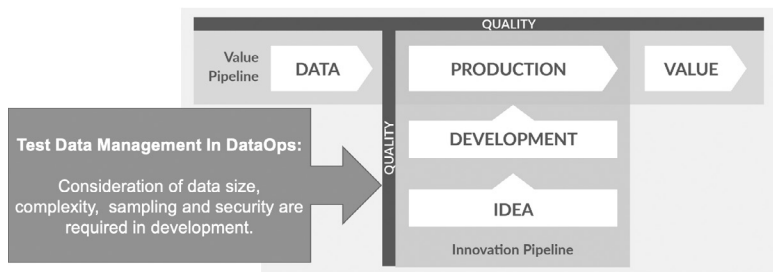


Figure 19: The concept of test data management is a first order problem in DataOps.

The concept of test data management is a first order problem in [DataOps](#) whereas in most DevOps environments, it is an afterthought. To accelerate analytics development, DataOps has to automate the creation of [development environments](#) with the needed data, software, hardware and libraries so innovation keeps pace with Agile iterations.

## DataOps Connects the Organization in Two Ways

[DevOps](#) strives to help development and operations (information technology) teams work together in an integrated fashion. In [DataOps](#), this concept is depicted in Figure 20. The development team are the analysts, scientists, engineers, architects and others who create data warehouses and analytics.

In data analytics, the operations team supports and monitors the data pipeline. This can be IT, but it also includes customers — the users who create and consume analytics. DataOps brings these groups together so they can work together more closely.

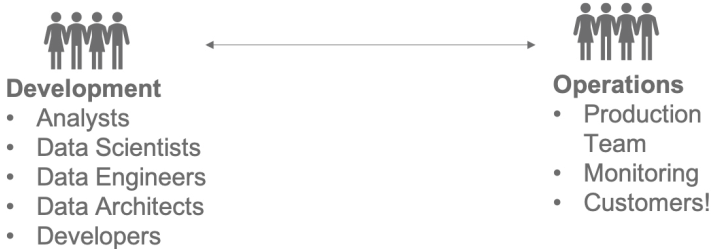


Figure 20: DataOps combines data analytics development and data operations

## Freedom vs. Centralization

DataOps also brings the organization together across another dimension. A great deal of data analytics development occurs in remote corners of the enterprise, close to business units, using self-service tools like Tableau, Alteryx, or Excel. These local teams, engaged in decentralized, distributed analytics creation play an essential role in delivering innovation to users. Empowering these pockets of creativity maintains the enterprise's competitiveness, but frankly, a lack of top-down control can lead to unmanaged chaos.

Centralizing analytics development under the control of one group, such as IT, enables the organization to standardize metrics, control [data quality](#), enforce security and governance, and eliminate islands of data. The issue is that too much centralization chokes creativity.

One important benefit of DataOps is its ability to harmonize the back-and-forth between the decentralized and centralized development of data analytics—the tension between [centralization and freedom](#). In a DataOps enterprise, new analytics

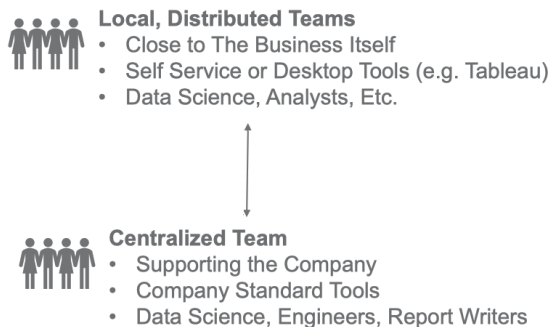


Figure 21: DataOps brings together centralized and distributed development

originate and undergo refinement in the local pockets of innovation. When an idea proves useful or is worthy of wider distribution, it is promoted to a centralized development group who can more efficiently and robustly implement it at scale.

DataOps brings localized and centralized development together enabling organizations to reap the efficiencies of centralization while preserving localized development—the tip of the innovation spear. DataOps brings the enterprise together across two dimensions as shown in Figure 22 — development/operations as well as distributed/centralized development.

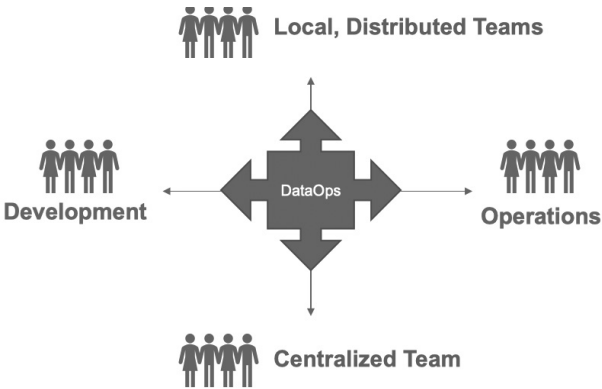


Figure 22: DataOps brings teams together across two dimensions — development/operations as well as distributed/centralized development.

DataOps brings three cycles of innovation between core groups in the organization: centralized production teams, centralized [data engineering](#)/analytics/science/governance development teams, and groups using self-service tools distributed into the lines business closest to the customer. Figure 23 shows the interlocking cycles of innovation.

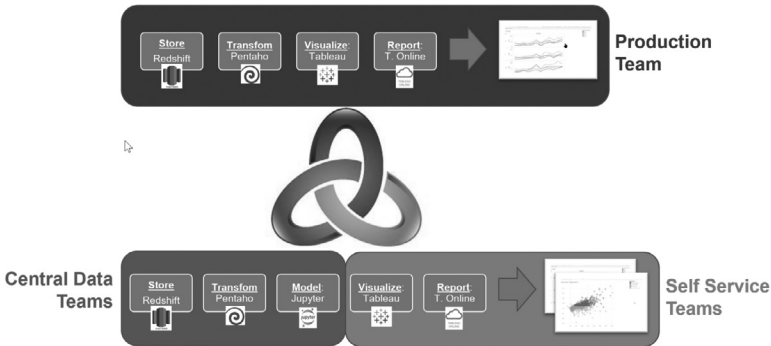


Figure 23: DataOps brings three cycles of innovation between production, central data, and self-service teams.

## Enterprise Example - Data Analytics Lifecycle Complexity

Having examined the [DataOps](#) development process at a high level, let's look at the development lifecycle in the enterprise context. Figure 24 illustrates the complexity of analytics progression from inception to production. Analytics are first created and developed by an individual and then [merged](#) into a team project. After completing [unit acceptance testing](#) (UAT), analytics move into production. The goal of DataOps is to create analytics in the individual [development environment](#), advance into production, receive feedback from users and then continuously improve through further iterations. This can be challenging due to the differences in personnel, tools, code, versions, manual procedures/automation, hardware, operating systems/libraries and target data. The columns in Figure 24 show the varied characteristics for each of these four environments.

The challenge of pushing analytics into production across these four quite different environments is daunting without DataOps. It requires a patchwork of manual operations and scripts that are in themselves complex to manage. Human processes are error-prone so data professionals compensate by working long hours, mistakenly relying on [hope and heroism](#) for success. All of this results in unnecessary complexity, confusion and a great deal of wasted time and energy. Slow progression through the lifecycle shown in Figure 24 coupled with high-severity errors finding their way into production can leave a data analytics team little time for innovation.

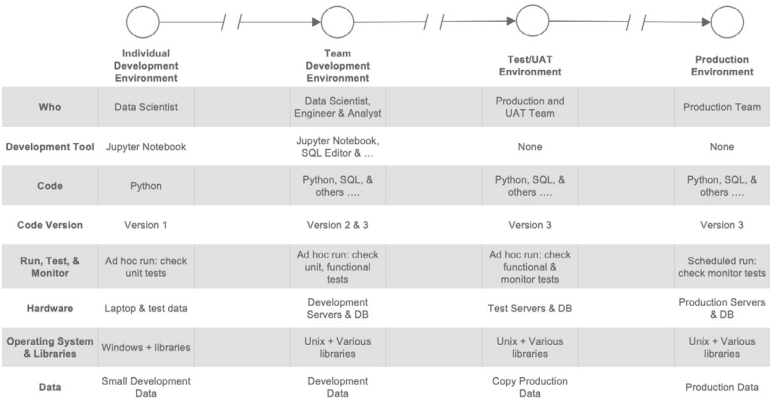


Figure 24: Data Analytics Development Lifecycle Complexities

## Implementing DataOps

DataOps simplifies the complexity of data analytics creation and operations. It aligns data analytics development with user priorities. It streamlines and automates the analytics development lifecycle — from the creation of sandboxes to deployment. DataOps controls and monitors the data factory so [data quality](#) remains high, keeping the data team focused on adding value.

You can get started with DataOps by implementing the [seven steps](#) in the last section. You can also adopt a [DataOps Platform](#) which will support DataOps methods within the context of your existing tools and infrastructure.

A DataOps Platform automates the steps and processes that comprise DataOps: sandbox management, orchestration, monitoring, testing, deployment, the data factory, dashboards, Agile, and more. A DataOps Platform is built for data professionals with the goal of simplifying all of the tools, steps and processes that they need into an easy-to-use, configurable, end-to-end system. This high degree of automation eliminates a great deal of manual work, freeing up the team to create new and innovative analytics that maximize the value of an organization's data.

# DataOps Resolves the Struggle Between Centralization and Freedom in Analytics

Freedom and employee empowerment are essential to innovation, but a lack of top-down control leads to chaos. Self-service tools enable data analysts to create new analytics very quickly, but they can drift in different directions. Imagine a team of analysts building reports that tally sales figures and each come up with a different result. One approach includes drop shipments and sales from distributors/subsidiaries. Another report might consist of product sales, but not services. These different approaches each have their use case, but from a manager's perspective inconsistency creates the appearance of inaccuracy. You can't establish a shared reality when everyone has different numbers.

Some managers respond to this challenge by centralizing analytics. With data and analytics under the control of one group, such as [IT](#), you can standardize metrics, control [data quality](#), enforce security and governance, and eliminate islands of data. All worthy endeavors, however forcing analytic updates through a heavyweight IT development process is a sure way to stifle innovation. It is one of the reasons that some companies take three months to deploy ten lines of SQL into production. Analytics have to be able to evolve and iterate quickly to keep up with user demands and fast-paced markets. Managers instinctively understand that data analytics teams must be free to innovate. The fast-growing self-service tools market (Tableau, Looker, etc.) addresses this market.

Centralizing analytics brings it under control but granting analysts free reign is necessary to stay competitive. How do you balance the need for centralization and freedom? How do you empower your analysts to be innovative without drowning in the chaos and inconsistency that a lack of centralized control inevitably produces? Visit any modern enterprise, and you will find this challenge playing out repeatedly in budget discussions and hiring decisions. You might say, it is a *struggle between centralization and freedom*.



# Roles in the Data-Analytics Organization

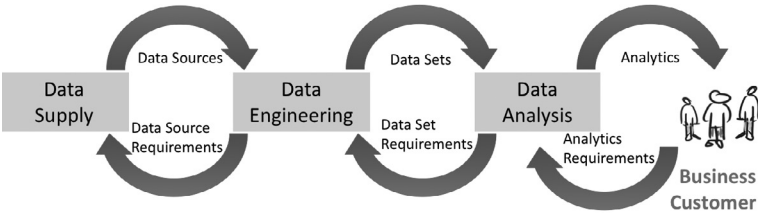


Figure 25: Data Supply, Data Engineering and Data Analysis work together in a supply chain to fulfill the analytics requirements of customers (business users).

## The Analytics Supply Chain

[DataOps](#) processes and tools offer you a way to harmonize these opposing forces, empowering data analysts, while exerting a measured amount of centralization and control on your end-to-end process. To explore these ideas further, we need to review the structure of the analytics supply chain, and how the various roles relate to each other. The analytics organization in a DataOps enterprise consists of three essential job functions: data analysts, data engineers and data suppliers. You can think of the three roles as groups forming a supply chain. Data suppliers extract data for data engineers who create targeted data sets. Data analysts consume these data sets and generate analytics for business use cases. As figure 25 shows, the three functions work together in an interlinked fashion with data and analytics flowing to the right and feedback and requirements flowing to the left. Each group focuses on its immediate customer (its right neighbor), but together they share the mission of delivering analytic insights to business users. While they share a common underlying mission, the three groups operate in different business contexts.

## ~Timing

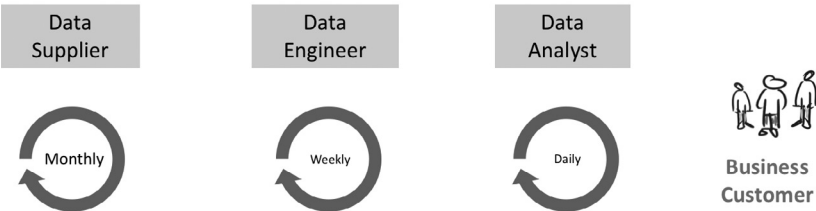


Figure 26: Data suppliers, engineers and analysts use different cycle times driven mainly by their tools, methods and proximity to demanding users.

## Data Analysts

Data analysts directly support business users who work in fast-paced markets, which continuously evolve. A successful analyst finds ways to respond quickly to user requests. If new analytics are needed, the analyst pulls that together. New charts and graphs, updates, changes to calculated fields, integrations of new data sets — top-performing analysts do whatever it takes to address user requirements.

Analysts choose tools and processes oriented toward this business context. They use powerful, self-service tools, such as Tableau, Alteryx, and Excel, to quickly create or iterate on charts, graphs, and dashboards. They organize their work into daily sprints (figure 26), so they can deliver value regularly and receive feedback from users immediately. Agile tools like Jira are an excellent way to manage the productivity of analyst daily sprints.

The data analyst is the tip of the innovation spear. Organizations must give data analysts maximum freedom to experiment. There are a lot more data in the world than companies can analyze. Not everything can be placed in data warehouses. Not all data *should* be operationalized. Companies need data analysts to play around with different data sets to establish what is predictive and relevant.

### When Freedom is Not Free

As their body of analytics grows, data analysts can get bogged down in non-value-add tasks. Self-service tools do not include mechanisms that promote and enable [reuse](#). The data analyst may end up copying a calculated field into many reports and then have to manage changes to that algorithm manually. It becomes a revision control nightmare. New data sets are often integrated using labor-intensive and error-prone manual steps. This becomes a heavy burden on data analysts, consuming more than 75% of their time. Help and support from [data engineering](#) addresses these challenges for the team.

Some companies mistakenly ask data engineering to create data sets for every idea. It is best to let analysts lead on implementing new analytic ideas and proving them out before considering how data engineering can help. For example, consider the following:

1. Have the new analytics proven to be useful to many users?
2. Have calculations been reused in many reports/charts/dashboards?
3. Can automation reduce duplication of effort or manual integration errors?
4. Does data quality need improvement?

By this standard, the organization focuses its data engineering resources on those items that give the most *bang for the buck*. Keep in mind that when analytics are moved into a data warehouse, some of the benefits of centralization come at the expense of reduced freedom — it is slower to update a data warehouse than a Tableau worksheet. It's important to wait until analytics have *earned the right* to make this transition. The value created by centralizing must outweigh the restriction of freedom.

## Data Engineers

[Data engineers](#) choose tools and processes which facilitate their objectives — to produce quality-checked data sets like data lakes, data warehouses and data marts for Data Analysts. These data sets include field calculations that analysts can leverage, promoting [reuse](#). Data engineers can also automate data integration and other processes, minimizing manual steps for the data analyst. With the added centralization offered by data engineering, analysts can mitigate non-value add tasks and keep innovating rapidly.

Data engineers utilize programmable platforms such as AWS, S3, EC2 and Redshift. These tools require programming in a high-level language and offer greater potential functionality than the tools used by analysts. The relative complexity of the tools and scope of projects in data engineering fit best in weekly Agile iterations (figure 26). [DataOps platforms](#) like [DataKitchen](#) enable the data engineer to streamline the quality control, orchestration and data operations aspects of their duties. With automated support for agile development, [impact analysis](#), and [data quality](#), the data engineer can stay focused on creating and improving data sets for analysts.

After data sets have proven their value, it's worth considering whether the benefits of further centralization outweigh the cost of a further reduction in freedom. Data suppliers fulfill the function of greater centralization by providing data sources or data extracts for data engineering.

There are several reasons that a project may have earned the right to transition to data suppliers. Analytics may provide functionality that executives wish to make available to the entire corporation, not just one business unit. It could also be a case of standardization — for example, the company wants to standardize on an algorithm for calculating market share. In another example, perhaps data engineering has implemented quality control on a data set and wishes to achieve efficiencies by pushing this functionality upstream to the data supplier. A data supplier may be an external third party or an internal group, such as an IT master data management (MDM) team.

## Master Data Management

In MDM, an enterprise links all of its critical data to a common reference. For example, in the pharmaceutical industry, there are 20-30 public data sets that describe physicians, payers and products. The initial [merging](#) and mastery of the data sets may, and in some cases should, be performed by the data engineering team, for example, for the purpose of business analytics.

After the usefulness of the mastered data is established, the company might decide that the data has broader uses. They may want the customer or partner list to be available for a portal or tied into a billing system. This use case requires a higher standard of accuracy for the mastered data than was necessary for the analytic data warehouse. It's appropriate at this point to consider moving the MDM to a data supplier, such as a corporate IT team, who are adept at tackling more extensive, development initiatives. Put another way, initial data mastery may have been good enough for analytic insights, but data must be perfect when it is being used in a billing system. The data supplier takes the MDM to the next level.

| Function                 | Data Supply                 | Data Engineering                     | Data Analysis                                   |
|--------------------------|-----------------------------|--------------------------------------|---|
| Iteration Cycle Time     | Monthly                     | Weekly                               | Daily   |
| Deliverables             | Data Extracts               | Organized, quality-checked data sets | Produce insights via charts, graphs, dashboards |
| Customer                 | Data Engineers              | Data Analysts                        | Business users                                  |
| Technical tools          | RDBS, MDM, Salesforce, etc. | AWS, <u>DataKitchen</u> , GIT        | Self-service tools                              |
| Process management tools | MS Office and others        | Jira                                 | Jira  |

Table 2: Data Supply, Data Engineering and Data Analysis prefer different tools and methods which influence their optimal cycle times

Data Suppliers

Projects transitioned to data suppliers tend to incorporate more process and tool complexity than those in [data engineering](#), leading to a more extended iteration period of one or more months (figure 26). These projects use tools such as RDBS, MDM, Salesforce, Excel, sFTP, etc., and rely upon waterfall project management and MS Project tracking. Table 2 summarizes tools and processes preferred by data suppliers as contrasted with engineers and analysts.

The Centralization-Freedom Spectrum

Data analysts, data engineers and data suppliers sit on a centralization-innovation (freedom) spectrum with data suppliers offering the most centralization capabilities, data analysts producing the fastest innovation and data engineering serving as a transition space in the center. The characteristic strengths and weaknesses of each of these groups are strongly influenced by their daily, weekly and monthly iteration periods. By organizing the various groups in this way, the company has access to the full spectrum of development; fast innovation, medium-scope development and longer-term, complex projects. For every need, the project has a home. The principle of granting analysts as much freedom as possible ensures that the innovation engine continues to turn. Waiting for analytics to stabilize and *earn the right* to be transitioned ensures that centralization adds value where it should and doesn't infringe on freedom and creativity.

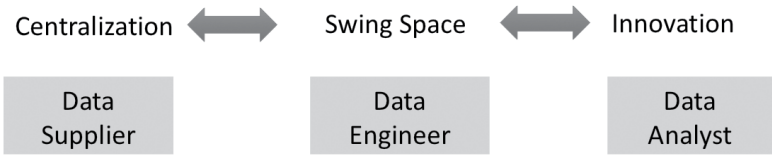


Figure 27: Data Suppliers, Data Engineers and Data Analysts sit on a spectrum of centralization and innovation/freedom.

# The DataOps Framework for Innovation Management

The supply chain model that we have discussed illustrates how [DataOps](#) processes and tools help enterprises empower data analysts while exerting a measured amount of centralization and control on the end-to-end data pipeline. The special sauce behind DataOps is automated orchestration, continuous deployment and testing/monitoring of the data pipeline. DataOps reduces manual effort, enforces [data quality](#) and streamlines the orchestration of the data pipeline.

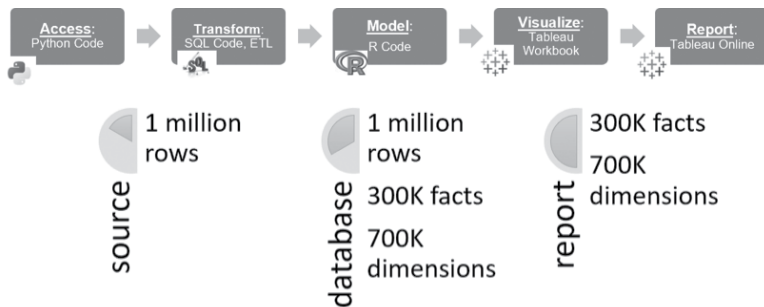


Figure 28: Tests verify that data rows, facts and dimensions match business logic throughout the data pipeline

For example, Figure 28 shows how the [DataOps platform](#) orchestrates, tests and monitors every step of the data operations pipeline, freeing up the team from significant manual effort. The test verifies that the quantity of data matches business logic at each stage of the data pipeline. If a problem occurs at any point in the pipeline, the analytics team is alerted and can resolve the issue before it develops into an emergency. With 24x7 monitoring of the data pipeline, the team can rest easy and focus on customer requirements for new/updated analytics.

## Superpower Mindset

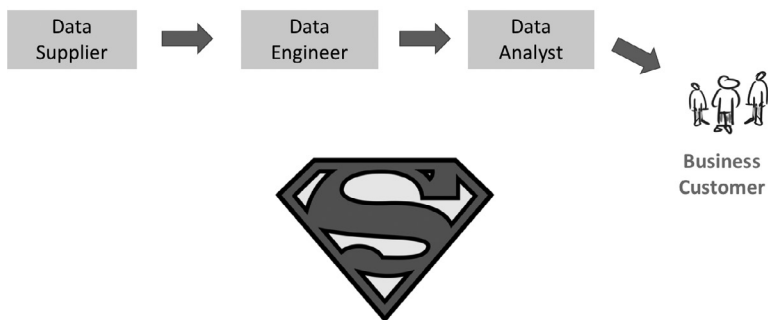


Figure 29: DataOps enables a superpower mindset —  
make your customer awesome

### Giving Your Team Superpowers

The goal of [DataOps](#) is to minimize overhead and free data engineers and analysts to focus on delivering analytics to customers. With automation, DataOps enables data professionals to improve their productivity by an order of magnitude. A single [data engineer](#) supports ten data analysts, who in turn support 100 business professionals. It's like gaining *superpowers*. Data suppliers, engineers and analysts then concentrate their energy on their primary objective — *making customers awesome*.



# DataOps Chicken Wings

by Nick Bracy

## INGREDIENTS AND TOOLS

- 1/4 cup of soy sauce
- 3 tbsp of sesame oil
- 1 1/2 tbsp siracha sauce
- Freshly grated ginger and garlic  
*(I use about 2-3 tsp of minced garlic — 2 cloves)*
- Ginger *(can be overpowering, a grated piece of ginger that is about a square inch is good, about the size of the top part of your thumb—from knuckle to the top of your thumb)*
- 1 tbsp of rice wine vinegar
- A generous tbsp of organic honey or agave
- Black pepper to taste
- 1/2 cup of ketchup (optional)

## INSTRUCTIONS

1. Place chicken drums and wings in a large zip-lock back, add marinade, seal zip-lock bag, mix contents of the bag around gently (you don't want to accidentally open the bag and marinate your kitchen floor or counter), make sure your chicken is well coated inside the bag.
2. Refrigerate your chicken in the marinade for 8-24 hours *(You can also just cook them right away if you don't have the time)*
3. Best slow cooked for 5-6 hours in a crockpot or on 225 degrees in a conventional oven—use all the contents in the bag. *(If you don't have that kind of time, bake at 400 degrees Fahrenheit.)* 3.5 lbs. of chicken should bake for 55-60 minutes; 4.5 lbs. of chicken requires 60-65 minutes.

*Recipe inspired by Joe DeFran*

# DataOps for the Chief Data Officer

## Warring Tribes into Winning Teams: Improving Teamwork in Your Data Organization

If the groups in your data-analytics organization don't work together, it can impact analytics-cycle time, data quality, governance, employee retention and more. A variety of factors contribute to poor teamwork. Sometimes geographical, cultural and language barriers hinder communication and trust. Technology-driven companies face additional barriers related to tools, technology integrations and workflows which tend to drive people into isolated silos.

### THE WARRING TRIBES OF THE TYPICAL DATA ORGANIZATION

The data organization shares a common objective; to create analytics for the (internal or external) customer. Execution of this mission requires the contribution of several groups shown in Figure 30. These groups might report to different management chains, compete for limited resources or reside in different locations. Sometimes they behave more like warring tribes than members of the same team.



Figure 30: Delivery of analytics (the value chain) to customers requires contributions from several groups in the data organization

Let's explore some of the factors that isolate the tribes from one another. For starters, the groups are often set apart from each other by the tools that they use. Figure 31 is the same value chain as above but reconstructed from the perspective of tools.

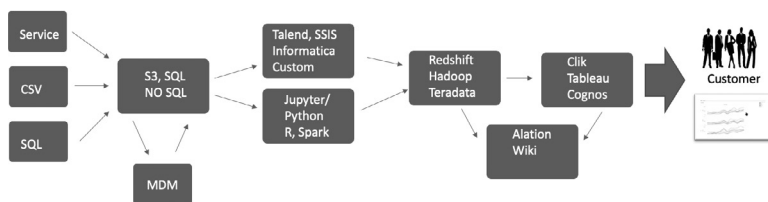


Figure 31: The value chain shown from a tools perspective

To be more specific, each of the roles mentioned above (figure 30) view the world through a preferred set of tools (figure 31):

- Data Center/IT — Servers, storage, software
- Data Science Workflow — Kubeflow, Python, R
- Data Engineering Workflow — Airflow, ETL
- Data visualization, Preparation — Self Service tools, Tableau, Alteryx
- Data Governance/Catalog (Metadata management) Workflow — Alation, Collibra, Wikis

The day-to-day existence of a data engineer working on a master data management (MDM) platform is quite different than a data analyst working in Tableau. Tools influence their optimal iteration cycle time, e.g., months/weeks/days. Tools determine their approach to solving problems. Tools affect their risk tolerance. In short, they view the world through the lens of the tools that they use.

The division of each function into a tools silo creates a sense of isolation which prevents the tribes from contemplating their role in the end-to-end data pipeline. The less they understand about each other, the less compelling the need to communicate about actions taken which impact others. Communication between teams (people in roles) is critical to the organization's success. Most analytics requires contributions from all the teams. The work output of one team may be an input to another team. In the figure below, the data (and metadata) build as the work products compound through the value chain.

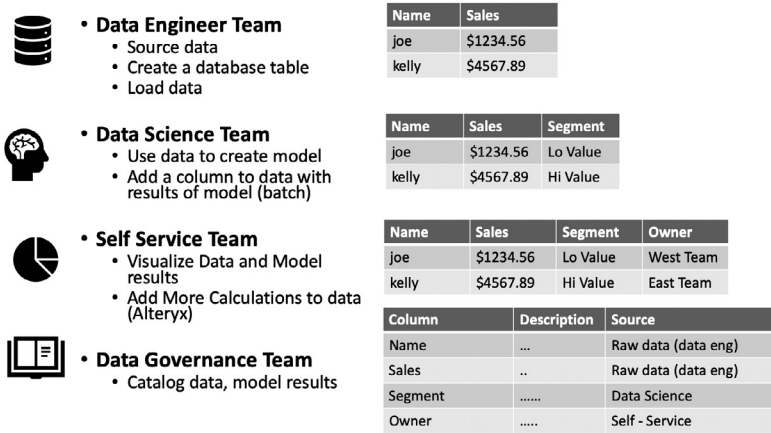


Figure 32: Each group adds unique value to analytics. In most cases, the work of one group is an input to the next group.

In many enterprises, there is a natural tendency for the groups to retreat into the complexity of their local workflow. In figure 33, we represent the local workflow of each tribe with a [directed-acyclic graph](#) (DAG).

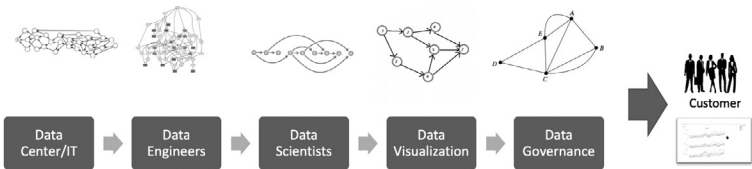


Figure 33: Work groups tend to focus on the complexity of their local workflow

It is too easy to overlook the fact that the shared purpose of these local workflows is to work together to publish analytics for end-customers.

## OTHER FACTORS THAT INCREASE GROUP ISOLATION

Group isolation is also induced by platforms, release cadence and geographic locations. The example below shows a multi-cloud or multi-data center pipeline with integration challenges.

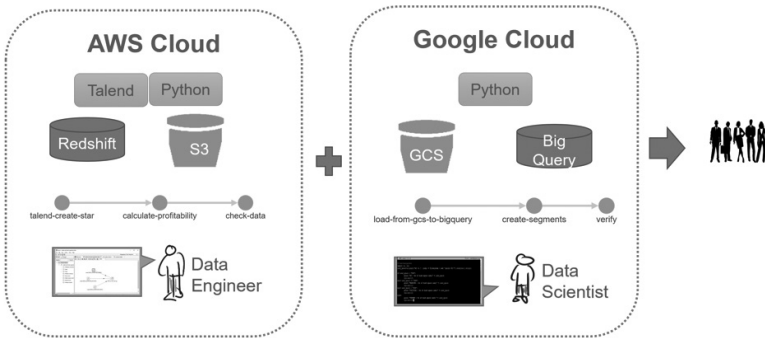


Figure 34: Multi-cloud or multi-data center pipelines with integration challenges

The two groups managing the two halves of the solution have difficulty maintaining quality, coordinating their processes and maintaining independence (modularity). Group one tests part one of the system (figure 35). Group two validates part two. Do the part one and two tests deliver a unified set of results (and alerts) to all stakeholders? Can tests one and two evolve independently without breaking each other? These issues repeatedly surface in data organizations.

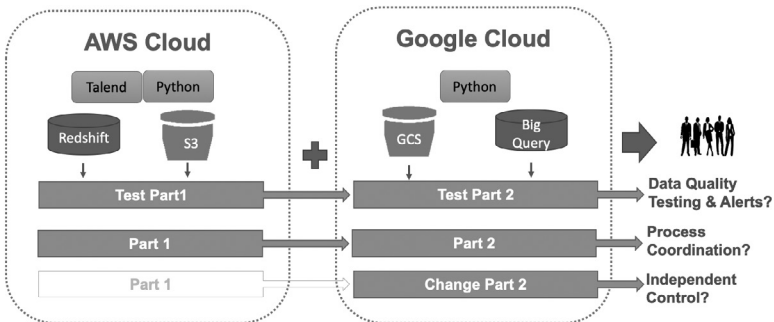


Figure 35: Integration challenges of multi-cloud or multi-data center solutions

In another example, assume that two groups are required to work together to deliver analytics to the VP of marketing. The home office in Boston handles data engineering and creates data marts. Their iteration period is weekly. The local team in New Jersey uses the data marts to create analytics for the VP of Marketing. Their iteration is daily (or hourly).

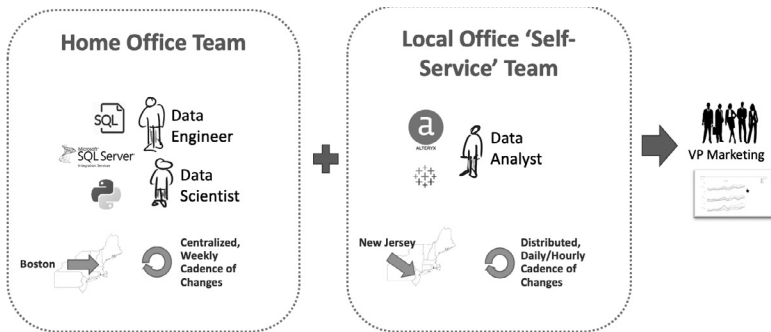


Figure 36: Issues related to multi-team workflows

One day, the VP of Marketing requests new analytics (deadline ASAP) from the data analysts for a meeting later that day. The analysts jump into action, but face obstacles when they try to add a new data set. They contact data engineering in Boston. Boston has its own pressures and priorities and their workflow, organized around a weekly cadence, can't respond to these requests on an "ASAP" basis.

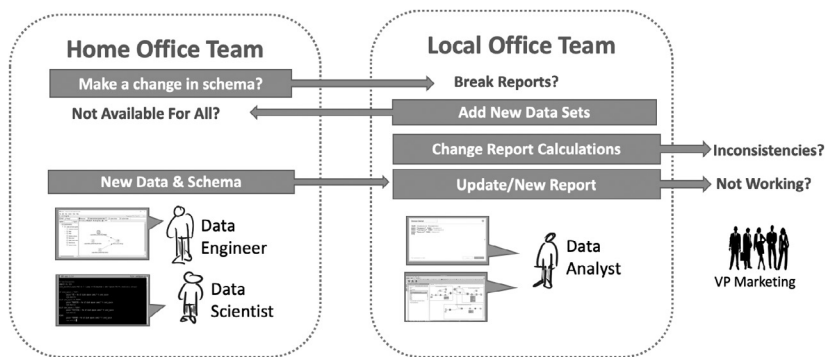


Figure 37: Challenges with multi-team coordination

The home office team in Boston finally makes the needed changes, but they inadvertently break other critical reports (figure 37). Meanwhile, out of desperation, the New Jersey team adds the required data sets and updates their analytics. The new data sets are only available to New Jersey, so other sites are now a revision behind. New Jersey's reports are inconsistent with everyone else's. Misunderstandings ensue. It's not hard to imagine why the relationship between these groups could be strained.

These challenges may seem specific to data organizations, but at a high level, everything that we have discussed boils down to poor communication and lack of coordination between individuals and groups. As such, we can turn to management science to better understand the problem and explore solutions.

# RELATIONAL COORDINATION

Strip away the technological artifacts from the situations described above and you are left with an organization that cannot foster strong role relationships and communication between employees. These challenges are not unique to technology-driven organizations. Many enterprises across a wide variety of industries face similar issues.

For those who don't remember, the airline business in the 1980s and 1990s was brutally competitive, but during this same period, Southwest Airlines revolutionized air travel. By the early 2000s, they had experienced 31 straight years of profitability and had a market capitalization greater than all the other major US airlines combined. Brandeis management professor [Jody Hoffer Gittel](#) investigated the factors in Southwest Airlines' performance and, back in 2003, published a quantitative, data-driven analysis shedding light on Southwest's success.

Dr. Gittel surveyed the major players in the airline industry and found a correlation between key performance parameters (KPP) and something that she termed [Relational Coordination](#) (RC), the way that relationships influence task coordination, for better or worse. "Relational coordination is communicating and relating for the purpose of task integration — a powerful driver of performance when work is interdependent, uncertain and time constrained."

In her study, higher RC levels correlated with better performance on KPPs, even when comparing two sites within the same company. Since that time RC has been applied in industries ranging from [healthcare to manufacturing](#) across 22 countries.

One common misconception is that RC focuses on personal relationships. While personal relationships are important, RC is more concerned with the relationship of roles and work-flows within the organization. RC studies how people interact and exchange information in executing their role-based relationships.

Relational Coordination can be expressed as characteristics of relationships and communication:

| Dimensions of RC     | Low RC   | High RC  |
|----------------------|--|--|
| <b>Relationships</b> | Functional Goals<br>Exclusive Knowledge<br>Lack of Respect | Shared Goals<br>Shared Knowledge<br>Mutual Respect |
| <b>Communication</b> | Infrequent<br>Delayed<br>Inaccurate<br>Finger-pointing     | Frequent<br>Timely<br>Accurate<br>Problem-solving  |

Members of the "Low-RC" organization express their goals solely in terms of their own function. They keep knowledge to themselves and there may be a tendency for one group to look down upon another group. Inter-group communication is inadequate, inaccurate and might be more concerned with finding blame than finding solutions. As expected, the "High-RC" organization embodies the exact opposite end of this spectrum.

“High-RC” team members understand the organization’s collective goal. They not only know what to do but why, based on a shared knowledge of the overall workflow. Everyone’s contribution is valued, and no one is taken for granted. There is constant communication, especially when a problem arises.

At this point you may be thinking: “OK fine, this is all *touchy-feely* stuff. I’ll try to smile more and I’ll organize a pizza party so everyone can get to know each other.” Maybe you should (smiling will make you feel good and parties are fun after all), but our experience is that the good feeling wears off once the last cupcake is gone and the mission-critical analytics are offline.

How do you keep people working *independently* and *efficiently* when their work product is a *dependency* for another team? How can one team *reuse* the data or artifacts or code that another team produces?

For most enterprises, improving RC requires foundational change. You need to examine your end-to-end data operations and analytics-creation workflow. Is it building up or tearing down the communication and relationships that are critical to your mission? Instead of allowing technology to be a barrier to Relational Coordination, how about utilizing automation and designing processes to improve and facilitate communication and coordination between the groups? In other words, you need to restructure your data analytics pipelines as services (or microservices) that create a robust, transparent, efficient, repeatable analytics process that unifies all your workflows.

## BUILDING A HIGH-RC ENTERPRISE USING DATAOPS

[DataOps](#) is a new approach to data analytics that applies [lean manufacturing](#), [DevOps](#) and [Agile development](#) methods to data analytics. DataOps unifies your data operations pipeline with the publication of new analytics under one orchestrated workflow.

**Robust** – Statistical process control (lean manufacturing) calls for [tests](#) at the inputs and outputs of each stage of the data operations pipeline. Tests also vet analytics deployments, like an [impact review board](#), so new analytics don’t disrupt critical operations.

**Transparent** – [Dashboards](#) display the status of new analytics development and the operational status of the data operations pipeline. Automated alerts communicate issues immediately to appropriate response teams. Team members can see a birds-eye-view of the end-to-end workflow as well as local workflows.

**Efficient** – Automated orchestration of the end-to-end data pipeline (from data sources to published analytics) minimizes manual steps that tie up resources and introduce human error. Balance is maintained between [centralization and decentralization](#); the need for fast-moving innovation, while supporting standardization of metrics, quality and governance.

**Repeatable** – [Revision control](#) with built-in error detection and fault resilience is applied to the data operations pipeline.

**Sharable and Chunkable** – Encourage reuse, by creating a [services oriented architecture](#) (SOA) for your team to use together.

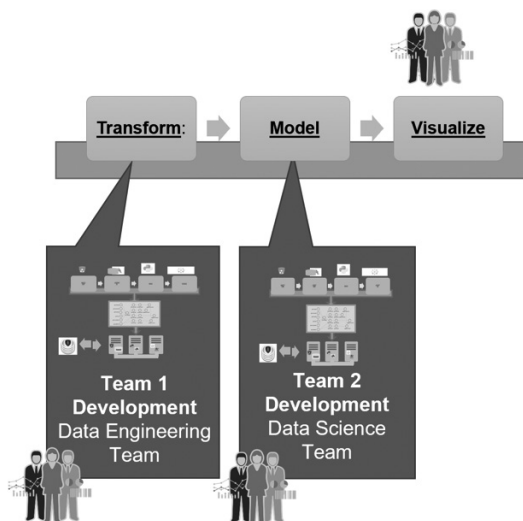


Figure 38: DataOps is a task coordination and communication framework that uses technology to break down the barriers between the groups in the data organization.

It may help to provide further concrete examples of a DataOps implementation and how it impacts productivity. Some of these points are further explained in our blog [DataOps in Seven Steps](#):

- **Data Sharing** – data sources flow into a [data lake](#) which is used to create data warehouses and data marts. Bringing data under the control of the data organization decouples it from IT operations and enables it to be shared more easily.
- **Deployment** of code into an existing system – [continuous integration](#) and [continuous delivery](#) of new analytics, leveraging [on-demand IT resources](#) and automated orchestration of integration, test and deployment.
- **Environment** startup, shutdown – With computing and storage on-demand from cloud services ([infrastructure as code](#)), large data sets and applications (test [environments](#)) can be quickly and inexpensively copied or provisioned to reduce conflicts and dependencies.
- **Testing** of data and other artifacts – [Testing](#) of inputs, outputs, and business logic are applied at each stage of the data analytics pipeline. Tests catch potential errors and warnings before they are released so the quality remains high. Test alerts immediately

inform team members of errors. Dashboards show the status of tests across the data pipeline. Manual testing is time-consuming and laborious so it can't be done in a timely way. A robust, automated test suite is a key element in continuous delivery.

- **Reuse** of a set of steps across multiple pipelines – Analytics [reuse](#) is a vast topic, but the basic idea is to componentize functionalities as services in ways that can be shared. Complex functions, with lots of individual parts, can be containerized using a container technology (like Docker).

We have seen marked improvements in analytics cycle time and quality with DataOps. It unlocks an organization's creativity by forging trust and close working relationships between data engineers, scientists, analysts and most importantly, users. DataOps is a task coordination and communication framework that uses technology to break down the barriers between the groups in the data organization. Let's look at the DataOps enterprise from the perspective of Relational Coordination.

| Dimension                                | Warring Tribes (Weak RC)   | The DataOps Enterprise (Strong RC)   |
|--|--|--|
| <b>Shared Goals</b>                      | -Separation into isolated tools silos with little regard for or understanding of others  | -Visibility into how analytics builds in stages until delivery to the customers or users   |
| <b>Shared Knowledge</b>                  | -System knowledge concentrated in the <a href="#">Impact Review Board</a><br>-Little reuse of analytics components<br>-Bureaucratic processes govern change<br>-Little visibility into the end-to-end pipeline | -System knowledge implemented in tests that anyone can view<br>-Reusable analytics components are maintained in <a href="#">source control</a><br>-Rapid <a href="#">cycle time</a> for new analytics<br>-Complete visibility into the global and local workflows of the data pipeline |
| <b>Mutual Respect</b>                    | -Each tribe in the data-analytics organization thinks it is better than the others   | -No one is taken for granted. The workflow shows how everyone's contribution is important  |
| <b>Frequent and Timely Communication</b> | -Communication is limited and definitely not a priority during frequent high-severity outages  | -Dashboards and automated alerts keep everyone informed 24x7   |
| <b>Problem-solving Communication</b>     | -When something goes wrong, the tribes focus on finding someone to blame   | -Discussion centers on tests that caught or will catch problems  |

## CONCLUSION

Technology companies face unique challenges in fostering positive interaction and communication due to tools and workflows which tend to promote isolation. This natural distance and differentiation can lead the groups in a data organization to act more like warring tribes than partners. These challenges can be understood through the lens of Relational Coordination; a management theory that has helped explain how some organizations achieve extraordinary levels of performance as measured by KPPs. DataOps is a tools and methodological approach to data analytics which raises the Relational Coordination between teams. It breaks down the barriers between the warring tribes of data organizations. With faster cycle time, automated orchestration, higher quality and better end-to-end data pipeline visibility, DataOps enables data analytics groups to better communicate and coordinate their activities, transforming warring tribes into winning teams.

# Improving Teamwork in Data Analytics with DataOps

*Previously, we wrote about how members of large data organizations sometimes behave more like [warring tribes](#) than members of the same team. We discussed how DataOps facilitates communication and task coordination between groups. Today we move from the macro to the micro-level. We look at how DataOps operates, within a team, to ease the flow of work from one team member to the next.*

Whether celebrating a team's success or contemplating its failure, people tend to focus on team leadership as the most crucial factor in team performance. Richard Hackman, a pioneer in the field of organizational behavior who studied teams for more than 40 years, called this the "leader attribution error." People generally pay more attention to factors that they can see (like leaders) than to the background structural and contextual factors that actually determine team performance. Hackman's groundbreaking insight was to look beyond personalities, attitudes, or behavioral styles. (Put down your Meyers-Briggs assessments.) What matters most to high-performance teams is the presence of "enabling conditions."

As W. Edwards Deming famously said, "A bad system will beat a good person every time." DataOps applies this point of view to data analytics by taking a process-oriented approach to improving analytics quality and reducing cycle-time. It seeks to uncover the specific factors that best contribute to team success.

We live in a world where the [average tenure of a CDO or CAO is about 2.5 years](#). A couple of years ago, Gartner predicted that 85 percent of AI projects would not deliver for CIOs. Forrester affirmed this unacceptable situation by stating that 75% of AI projects underwhelm. Clearly, data-analytics teams need a "tune-up."

After conducting 300 interviews and 4,200 surveys over 15 years, Haas and Mortensen (HBR, June 2016) built upon Richard Hackman's work by identifying four specific conditions most critical for team success:

1. **Compelling direction** – explicit and consequential goals that the team is working toward
2. **Strong structure** – includes optimally designed tasks and processes, and norms that promote positive dynamics.
3. **Supportive context** – includes an information system that provides access to the data needed for the work, and the material resources required to do the job
4. **Shared mindset** – collective identity and a shared reality

Per Haas and Mortensen, teams are more diverse, dispersed, digital, and dynamic than ever before. Modern organizations suffer from two corrosive problems – "*us versus them*" thinking and *incomplete information*. The four critical enabling conditions above help teams overcome these two pervasive problems and can raise overall team productivity while improving the quality of their work product.

## INSIGHTS UNLIMITED

When enterprises invite us in to talk to them about DataOps, we generally encounter dedicated and competent people struggling with conflicting goals/priorities, weak process design, insufficient resources, clashing mindsets and differing views of reality. We would love to bring you inside these meetings, behind the closed doors, so you could see and experience this for yourself.

Imagine a typical enterprise. We'll call them.... "*Insights Unlimited.*" Let's peek inside the data team's weekly staff meeting:

**Manager:** *Good morning, everyone. As you know, our new Chief Data Officer has been asking questions about the large and growing list of work items on Jira. The backlog has grown steadily and...*

**Eric (Production Engineer):** *You're kidding me right. I lost most of last week chasing down data errors that originated upstream from one of our data sources. And the new reports that the development team gave me last week took 20 hours to install and then broke the weekly sales report. I thought Bill's (VP of Sales) head was going to explode.*

**Padma (Data Engineer):** *Hey, if you had let me test the changes in the "real" environment, I could have caught those problems upfront.*

**Eric (Production Engineer):** *As I have said before, the operational systems are not a sandbox. Plus, we have to control access to private HIPAA data.*

A typical data analytics team has many key players, with distinct skill sets and tool preferences. There may be production engineers, data engineers, data analysts, data scientists, BI analysts, QA engineers, test engineers, ETL engineers, DBAs, governance and more. In our little anecdote, we could have filled a room full of grumpy and frustrated data professionals and business colleagues. For the sake of simplicity, we pared the team down to two members, Eric and Padma, who could each represent many people. To further explore the teamwork issues at *Insights Unlimited*, let's get to know Eric and Padma a little better. Note, that we'll be meeting a third key player on our data team as the exploration of *Insights Unlimited* continues.

### ERIC, PRODUCTION ENGINEER AT INSIGHTS UNLIMITED

**Role:** Production Perfectionist

**Goals:** Protect and perfect the daily grind of delivering data; *minimize* errors and chaos

**Skills:** Taskmaster, *little-bit-of-everything* tech skill



**Eric – Production Engineer**

Eric is the backbone of data operations and keeps the operational systems running. This is a mission-critical role, and every error has visibility. Eric regularly takes calls at odd hours. We have to forgive Eric for being a little blunt and insensitive to the needs of the data science team. He has learned the hard way never to let anyone in the development team anywhere near the production systems. He gets a little impatient debugging other people's analytics. He wastes a lot of time responding to high-severity

alerts generated by poor data quality. He has the unenviable job of standing in front of directors and VPs and explaining what broke their charts and graphs. Every organization needs an Eric, and sometimes he is an oversubscribed resource (a [bottleneck](#)). If deploying new analytics into production requires Eric's time, then his backlog of tasks will grow, increasing cycle time. It's a multi-language, multi-tool, multi-platform world and Eric has to manage all of that. Eric is also a gatekeeper. Complex, risky changes are only allowed when there is scheduled downtime, like a long weekend.

## **PADMA, DATA ENGINEER AT INSIGHTS UNLIMITED**

**Role:** Data Doer (or perhaps data scientist, BI/Analysts, etc.)

**Goals:** Create cool features for customers, flexible data architecture

**Skills:** SQL, ETL Tools, databases, machine learning



**Padma – Data Engineer**

Padma is the star that turns ideas into analytics that serve the business. She's an expert in analytics and machine learning tools. What motivates Padma is creating exciting new analytics. Whereas Eric wants to control change to reduce errors, Padma values a flexible data architecture that can be adapted quickly to new requirements. She wants to add new data sources and update schemas easily. Padma is a thought leader in AI and data science. She's less interested in the IT infrastructure that powers the data pipeline. When a new project is assigned, Padma

sometimes has to wait months for the IT department to order and configure a new development system or give her access to new data. She also waits weeks or months for the production team to deploy her new analytics. With the company's inefficient processes, Padma has trouble keeping up with user demands for new analytics, and colleagues sometimes think "she" is the bottleneck. Padma puts a lot of effort into quality, but because her test environment is different from production, there are always issues that surface during the production integration. Sometimes erroneous data flows into Padma's analytics, distorting results. That isn't something she can easily anticipate or address because operations lie outside her domain.

## **WITHOUT DATAOPS, A BAD "SYSTEM" OVERWHELMS GOOD PEOPLE**

The situation that we see playing out with Eric and Padma repeats in many organizations we encounter. Padma and Eric have different pressures and priorities. Inadequate workflow processes prevent them from doing their best work. The team lacks the structural and contextual support necessary to enable successful teamwork.

Imagine that the Vice President of Marketing makes an urgent request to the data analytics team: "I need new data on profitability ASAP." At *Insights Unlimited*, the process for creating and deploying these new analytics would go something like this:

1. The new requirement falls outside the scope of the development “plan of record” for the analytics team. Changing the plan requires departmental meetings and the approval of a new budget and schedule. Meetings ensue.
2. Padma requests access to new data. The request goes on the IT backlog. IT grants access after several weeks.
3. Padma writes a functional specification and submits the proposed change to the [Impact Review Board](#) (IRB), which meets monthly. A key-person is on vacation, so the proposed feature waits another month.
4. Padma begins implementation. The change that she is making is similar to another recently developed report. Not knowing that, she writes the new analytics from scratch. Padma’s test environment does not match “production.” so her testing misses some corner cases.
5. Testing on the target environment begins. High-severity errors pull Eric into an “all-hands-on-deck” situation, putting testing temporarily on hold.
6. Once the fires are extinguished, Eric returns to testing on the target and uncovers some issues in the analytics. Eric feeds error reports back to Padma. She can’t easily reproduce the issues because the code doesn’t fail in the “dev” environment. She spends significant effort replicating the errors so she can address them. The cycle is repeated a few times until the analytics are debugged.
7. Analytics are finally ready for deployment. Production schedules the update. The next deployment window available is in three weeks.
8. After several months have elapsed (total cycle time), the VP of Marketing receives the new analytics, wondering why it took so long. This information could have boosted sales for the current quarter if it had been delivered when she had initially asked.

Every organization faces unique challenges, but the issues above are ubiquitous. The situation we described is not meeting anyone’s needs. Data engineers went to school to learn how to create analytic insights. They didn’t expect that it would take six months to deploy twenty lines of SQL. The process is a complete hassle for IT. They have to worry about governance and access control and their backlog is entirely unmanageable. Users are frustrated because they wait far too long for new analytics. We could go on and on. No one here is enjoying themselves.

The frustration sometimes expresses itself as conflict and stress. From the outside, it looks like a teamwork problem. No one gets along. People are rowing the boat in different directions. If managers want to blame someone, they will point at the team leader.

At this point, a manager might try beer, donuts and trust exercises (hopefully not in that order) to solve the “teamwork issues” in the group. Another common mistake is to coach the group to work more [slowly and carefully](#). This thinking stems from the fallacy that you have to *choose between quality and cycle time*. In reality, you can have both.

This team lacks the critical enabling conditions of success that Haas and Mortensen identified. We recommend a process-oriented solution that addresses everyone's goals and priorities, coordinates tasks, provisions resources and creates a shared reality. Our method for turning a *band of squabbling data professionals* into a high-performance team is called [DataOps](#). (By the way, if you are new to the term, DataOps is the [hottest thing](#) in data science.)

## DATAOPS IMPROVES TEAMWORK

DataOps shortens the [cycle time](#) and improves the quality of data analytics. Data teams that do not use DataOps may try to reduce the number of errors by being more [cautious and careful](#). In other words, *slowing down*. DataOps helps organizations improve data quality while going faster. This might seem impossible until you learn more about how DataOps approaches analytics development and deployment.

DataOps is a set of methodologies supported by tools and automation. To say it in one breath; think [Agile development](#), [DevOps](#) and [lean manufacturing](#) (i.e., statistical process controls) applied to data analytics. DataOps comprehends that enterprises live in a multi-language, multi-tool, heterogeneous environment with complex workflows. To implement DataOps, extend your existing environment to align with DataOps principles. As we have written extensively, you can implement DataOps by yourself in [seven steps](#), or you can adopt a [DataOps Platform](#) from a third party vendor. In our example case, we are going to assume that *Insights Unlimited* begins using the DataOps Platform from [DataKitchen](#). First, we'll describe how DataOps addresses the process, resource and information sharing challenges at *Insights Unlimited*. Next, we'll describe an analytics development project at *Insights Unlimited* using DataOps.

## DATAOPS JOB #1: ABSTRACTING, SEPARATING AND ALIGNING RELEASE ENVIRONMENTS

Enterprises that collocate development and production on the same system face a number of issues. Analytics developers sometimes make changes that create side effects or break analytics. Development can also be processor intensive, impacting production performance and query response time.

DataOps provides production and development with dedicated system [environments](#). Some enterprises take this step but fail to align these environments. Development uses cloud platforms while production uses on-prem. Development uses clean data while production uses raw data. The list of opportunities for misalignment are endless. DataOps requires that system environments be aligned. In other words, as close as possible to identical. The more similar, the easier it will be to migrate code and replicate errors. Some divergence is necessary. For example, data given to developers may have to be sampled or masked for practical or governance reasons.

Figure 39 below shows a simplified production environment. The system transfers files securely using SFTP. It stores files in S3 and utilizes a Redshift cluster. It also uses Docker containers and runs some Python. Production alerts are forwarded to a Slack channel in real-time. Note that we chose an example based on Amazon Web Services, but we could have selected any other tools. Our example applies whether the technology is Azure, GCP, on-prem or anything else.

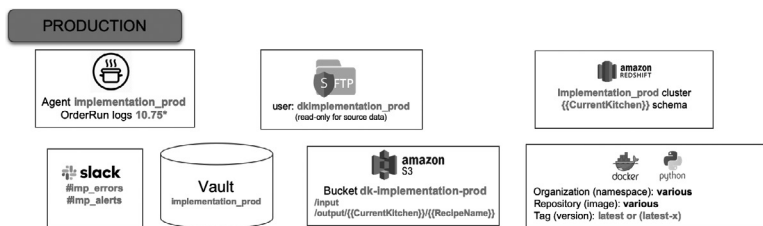


Figure 39: Simplified Production Technical Environment

DataOps segments production and development into separate release environments — see Figure 40. In our parlance, a release environment includes a set of hardware resources, a software toolchain, data, and a security Vault which stores, encrypted, sensitive access control information like usernames and passwords for tools. Our production engineer, Eric, manages the production release environment. Production has dedicated hardware and software resources so Eric can control performance, quality, governance and manage change. The production release environment is secure — the developers do not have access to it.

The development team receives its own separate but equivalent release environment, managed by the third important member of our team; Chris, Insight Unlimited's [DataOps Engineer](#). Chris also implements the infrastructure that abstracts the release environments so that analytics move easily between dev and production. We'll describe this further down below. Any existing team member, with DataOps skills, can perform the DataOps engineering function, but in our simplified case study, adding a person will better illustrate how the roles fit together.

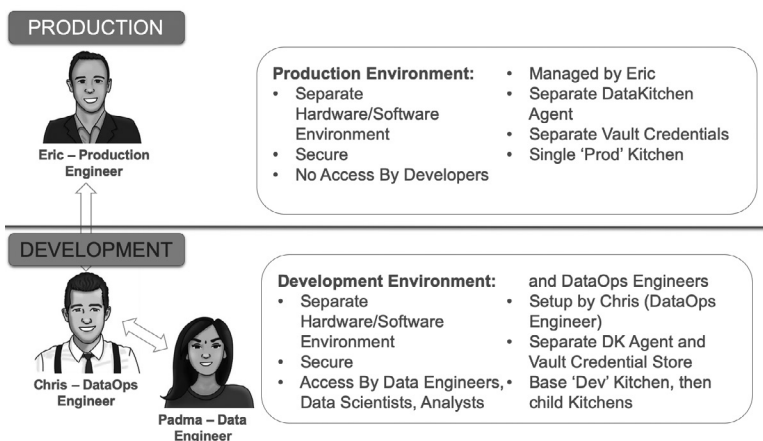


Figure 40: Production and development maintain separate but equivalent environments. The production engineer manages the production release environment and the DataOps Engineer manages the development release environment.

Chris creates a development release environment that matches the production release environment. This alignment reduces issues when migrating analytics from development to production. Per Figure 40, the development environment has an associated security Vault, just like the production environment. When a developer logs into a development workspace, the security Vault provides credentials for the tools in the development release environment. When the code seamlessly moves to production, the production Vault supplies credentials for the production release environment.

Figure 41 below illustrates the separate but equivalent production and development release environments. If you aren't familiar with "environments," think of these as discrete software and hardware systems with equivalent configuration, tools and data.

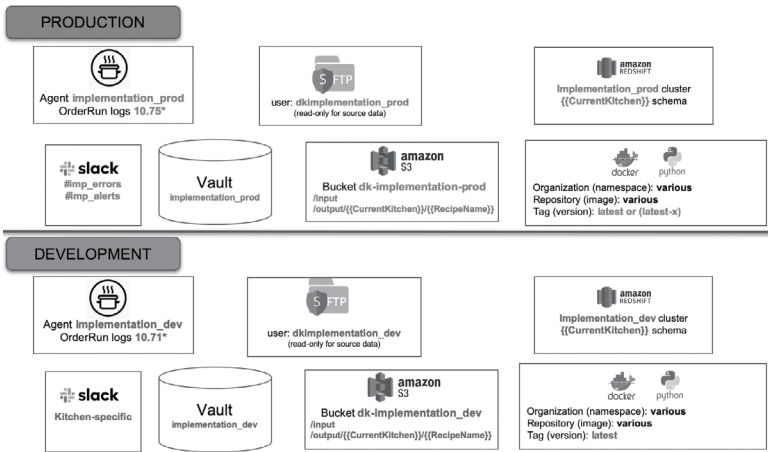


Figure 41: DataOps segments the production and development workspaces into separate but equivalent release environments.

Before we continue any further, let's formally add Chris to the team.

## CHRIS, DATAOPS ENGINEER

**Role:** Cycle Time and Quality Optimizer

**Goals:** Setup and maintain development environments; accelerate and ease deployment

**Skills:** DevOps, cloud/IT, DataKitchen, security



**Chris – DataOps Engineer**

Chris uses DataOps to create and implement the processes that enable successful teamwork. This activity puts him right at the nexus between data-analytics development and operations. At *Insights Unlimited*, Chris is one of the most [important and respected](#) members of the data team. He creates the mechanisms that enable work to flow seamlessly from development to production. Chris makes sure that environments are aligned and that everyone has the hardware, software, data, network and other resources that they need. He also makes available

software components, created by team members, to promote [reuse](#) — a considerable multiplier of productivity. In our simple example, Chris manages the tasks that comprise the pre-release process. Padma appreciates having Chris on the team because now she has everything that she needs to create analytics efficiently on a self-service basis. Eric is happy because DataOps has streamlined deployment, and expanded testing has raised both data and analytics quality. Additionally, there is much greater visibility into the artifacts and logs related to analytics, whether in development, pre-release or in production. It's clear that Chris is a key player in implementing DataOps. Let's dive deeper into how it really works.

## A DATAOPS “KITCHEN”: A RELEASE ENVIRONMENT, WORKSPACE AND PIPELINE BRANCH

Our development team in Figure 40 consists of Chris and Padma. In a real-world enterprise, there could be dozens or hundreds of developers. DataOps helps everyone work as a team by minimizing the amount of rekeying required so that analytics move seamlessly from developer to developer and into production. DataOps also organizes activities so that tasks remain coordinated and team members stay aligned. The foundation of these synchronized activities is a virtual workspace called a “Kitchen.”

A Kitchen is a development workspace with everything that an analytics developer requires. It contains hardware, software, tools, code (with [version control](#)) and data. A Kitchen [points](#) to a release environment which gives it access to all of the resources associated with that environment. A Kitchen also enforces workflow and coordinates tasks.

The processing pipelines for analytics consist of a series of steps that operate on data and produce a result. We use the term “Pipeline” to encompass all of these tasks. A DataOps Pipeline encapsulates all the complexity of these sequences, performs the orchestration work, and tests the results. The idea is that any analytic tool that is invocable under software control can be [orchestrated](#) by a DataOps Pipeline. Kitchens enable team members to access, modify and execute workflow Pipelines. A simple Pipeline is shown in Figure 42.

Pipelines, and the components that comprise them, are made visible within a Kitchen. This encourages reuse of previously developed analytics or services. Code reuse can be a significant factor in reducing cycle time.

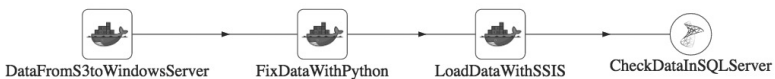


Figure 42: A simple DataOps Pipeline is represented by a directed acyclic graph (DAG). Each node in the graph is a sequence of orchestrated operations.

Kitchens should also tightly couple to version control (*Insights Unlimited* uses Git). When the development team wants to start work on a new feature, they instantiate a new child Kitchen which creates a corresponding Git [branch](#). When the feature is complete, the Kitchen is merged back into its parent Kitchen, initiating a Git merge. The Kitchen hierarchy aligns with the source control branch tree. Figure 43 shows how Kitchen creation/deletion corresponds to a version control branch and [merge](#).

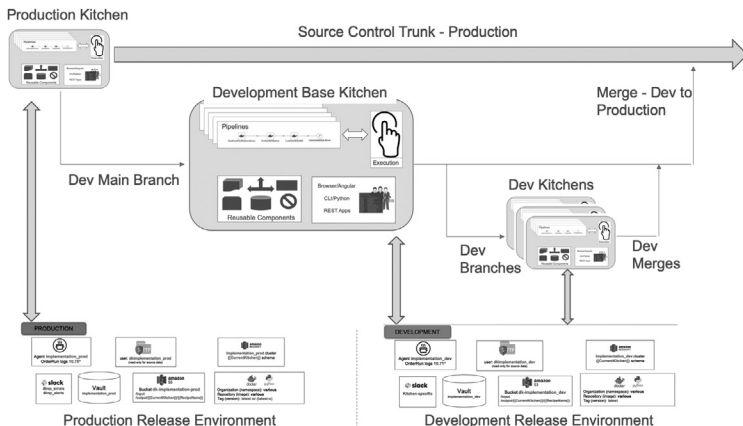


Figure 43: Kitchens point to a release environment. They represent source control branches and merges, and also serve as development, test and release workspaces.

Kitchens may be persistent or temporary; they may be private or shared, depending on the needs of a project. Access to a Kitchen is limited to a designated set of users or “Kitchen staff.” The Vault in a release environment supplies a Kitchen with the set of usernames and passwords needed to access the environment toolchain.

DataOps empowers an enterprise to provide people access to data, eliminating gatekeepers. As mentioned above, developers access test data from within a Kitchen. In another example, a Pipeline could extract data from a data lake and create a data mart or flat file that serves Alteryx, Tableau and Excel users in the business units. DataOps promotes and enables *data democratization*, providing everyone access to the data relevant to their job. When “self-service” replaces “gatekeepers,” more work gets done in parallel and analytics development cycle-time accelerates significantly.

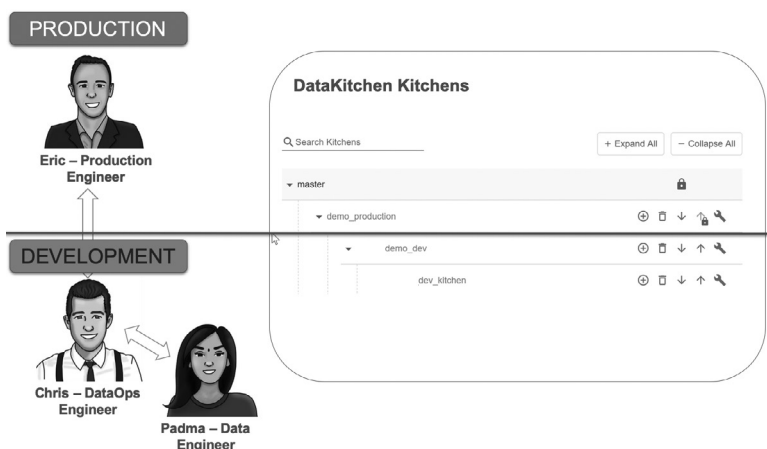


Figure 44: Eric, Chris and Padma each have personal Kitchens, organized in a hierarchy that aligns with their workflow.

Figure 44 above shows a Kitchen hierarchy. The base Kitchen is “demo\_production,” which points to the production release environment described earlier. This Kitchen is Eric’s workspace, and it enables him to coordinate his interactions with the development team. There is only one Kitchen corresponding to Eric’s production release environment. No iterative work takes place in production. Instead, think of “demo\_production” as a manufacturing flow where assembly lines run on a tight schedule.

Chris’ workspace is a Kitchen called “demo\_dev.” The “demo\_dev” Kitchen is the baseline development workspace, and it [points](#) to the development release environment introduced above, at the bottom of Figure 41. In our example, Chris’ Kitchen serves as a pre-release staging area where merges from numerous child development Kitchens consolidate and integrate before being deployed to production. With release [environments](#) aligned, Kitchens don’t have to do anything different or special for merges across release environments versus merges within a release environment.

Every developer needs a workspace so they may work productively without impacting or being impacted by others. A Kitchen can be persistent, like a personal workspace, or temporary, tied to a specific project. Once Kitchen creation is set-up, team members create workspaces as needed. This “self-service” aspect of DataOps eliminates the time that developers used to wait for systems, data, or approvals. DataOps empowers developers to *hit the ground running*. In Figure 44, Padma has created the Kitchen “dev\_kitchen.” Padma’s Kitchen can leverage Pipelines and other services created by the dev team.

## DATAOPS SEGREGATES USER ACTIVITY

With multiple developers sharing a release environment, the DataOps Platform segregates developer activity. For example, all of the developer Kitchens share the Redshift cluster shown in Figure 41. Note the notation “{{CurrentKitchen}}” associated with Redshift in Figure 41. Each developer has a Redshift schema within the cluster identified by their Kitchen name. For example, an access by Padma would target a schema identified by her unique Kitchen name “dev\_kitchen.” The DataOps Platform uses Kitchen names and other identifiers to segregate user activity within a shared release environment. Segregation helps keep everyone’s work isolated while sharing development resources.

Slack messages are similarly segregated by Kitchen, in our Figure 41 example. Note how production alerts are directed to the Slack channels “#imp\_errors” and “#imp\_alerts,” while dev alerts are sent to a kitchen-specific Slack channel. This prevents production from seeing any dev-related slack messages. It also prevents the developers from receiving each other’s Slack messages. Alerts could easily be managed on a much finer-grained level if required.

## DATAOPS AT INSIGHTS UNLIMITED

Let’s look at how *Insights Unlimited* is now utilizing a DataOps Platform to develop and deliver analytics with minimal cycle time and unsurpassed quality. We’ll walk through an example of how DataOps helps team members work together to deploy analytics into production.

Think back to the earlier request by the VP of Marketing for “new analytics.” DataOps coordinates this multi-step, multi-person and multi-environment workflow and manages it from

inception to deployment. The *Insights Unlimited* DataOps process addresses this requirement in six steps.

## STEP 1 — STARTING FROM A TICKET

The Agile Sprint meeting commits to the new feature for the VP of Marketing in the upcoming iteration. The project manager creates a JIRA ticket.

## STEP 2 — CREATION OF THE DEVELOPMENT KITCHEN

In a few minutes, Padma creates a development Kitchen for herself and gets to work. Chris has automated the creation of Kitchens to provide developers with the test data, resources and Git branch that they need. Padma's Kitchen is called "dev\_Kitchen" (see Figure 44). If Padma takes a technical risk that doesn't work out, she can abandon this Kitchen and start over with a new one. That effectively deletes the first Git branch and starts again with a new one.

## STEP 3 — IMPLEMENTATION

Padma's Kitchen provides her with Pipelines that serve as a significant head start on the new profitability analytics. Padma procures the test data she needs (de-identified) and configures toolchain access (SFTP, S3, Redshift, ...) for her Kitchen. Padma implements the new analytics by modifying an existing Pipeline. She adds additional [tests](#) to the existing suite, checking that incoming data is clean and valid. She writes tests for each stage of ETL/processing to ensure that the analytics are working from end to end. The tests verify her work and will also run as part of the production flow. Her new Pipelines include orchestration of the data and analytics as well as all tests. The tests direct messages and alerts to her Kitchen-specific Slack channel. With the extensive testing, Padma knows that her work will migrate seamlessly into production with minimal effort on Eric's part. Now that release environments have been aligned, she's confident that her analytics work in the target environment.

Before she hands off her code for pre-production staging, Padma first has to [merge](#) down from "demo\_dev" Kitchen so that she can integrate any relevant changes her coworkers have made since her [branch](#). She reruns all her tests to ensure a clean merge. If there is a conflict in the code merge, the DataOps Platform will pop-up a three panel UI to enable further investigation and resolution. When Padma is ready, she updates and reassigns the JIRA ticket. If the data team were larger, the new analytics could be handed off from person to person, in a line, with each person adding their piece or performing their step in the process.

## STEP 4 — PRE-RELEASE

In our simple example, Chris serves as the pre-release engineer. With a few clicks, Chris merges Padma's Kitchen "dev\_Kitchen" back into the main development Kitchen "demo\_dev," initiating a Git merge. After the merge, the Pipelines that Padma updated are visible in Chris' Kitchen. If Chris is hands-on, he can review Padma's work, check artifacts, rerun her tests, or even add a few tests of his own, providing one last step of QA or governance. Chris creates a schedule that, once enabled, will automatically run the new Pipeline every Monday at 6 am. When Chris is satisfied, he updates and reassigns the JIRA ticket, letting Eric know that the feature is ready for deployment.

## STEP 5 – PRODUCTION DEPLOYMENT

Eric easily merges the main development Kitchen “demo\_dev” into the production Kitchen, “demo\_production,” corresponding to a Git merge. Eric can now see the new Pipelines that Padma created. He inspects the test logs and reruns the new analytics and tests to be 100% sure. The release environments match so the new Pipelines work perfectly. He’s also happy to see tests verifying the input data using DataOps statistical process control. Tests will detect erroneous data, before it enters the production pipeline. When he’s ready, Eric enables the schedule that Chris created, integrating the new analytics into the operations pipeline. DataOps redirects any Slack messages generated by the new analytics to the production Slack channels.

## STEP 6 – CUSTOMER SEES RESULTS

The VP of Marketing sees the new customer segmentation and she’s delighted. She then has an epiphany. If she could see this new data combined with a report that Padma delivered last week, it could open up a whole new approach to marketing for *Insights Unlimited* – something that she is sure the competitors haven’t discovered. She calls the analytics team and... back to Step 1.

## DATAOPS BENEFITS

As our short example demonstrated, the DataOps Teamwork Process delivers these benefits:

- **Ease movement between team members with many tools and environments** – Kitchens align the production and development environment(s) and abstract the machine, tools, security and networking resources underlying analytics. Analytics easily migrate from one team member to another or from dev to production. Kitchens also bind changes to source control.
- **Collaborate and coordinate work** – DataOps provides teams with the compelling direction, strong structure, supportive context and shared mindset that are necessary for effective teamwork.
- **Automate Work and Reduce errors** – Automated orchestration reduces process variability and errors resulting from manual steps. Input, output and business logic tests at each stage of the workflow ensure that analytics are working correctly, and that data is within statistical limits. DataOps runs tests both in development and production, continuously monitoring quality. Warnings and errors are forwarded to the right person/channel for follow up.
- **Maintain security** – Kitchens are secured with access control. Kitchens then access a release environment toolchain using a security Vault which stores unique usernames/passwords.
- **Leverage best practices and re-use** – Kitchens include Pipelines and other reusable components which data engineers can leverage when developing new features.
- **Self-service** – Data professionals can move forward without waiting for resources or committee approval.

- **Data democratization** – Data can be made available to more people, even users outside the data team, who bring contextual knowledge and domain expertise to data-analytics initiatives. “Self-service” replaces “gatekeepers” and everyone can have access to the data that they need.
- **Transparency** – Pipeline status and statistics are available in messages, reports and dashboards.

## SMOOTH TEAMWORK WITH DATAOPS

DataOps addressed several technical and process-oriented bottlenecks at *Insights Unlimited* that previously delayed the creation of new analytics for months. Their processes can improve further, but they are now an order of magnitude faster and more reliable. At the next staff meeting, the mood of the team is considerably improved:

***Manager:** Good morning, everyone. I'm pleased to report that the VP of Marketing called the CDO thanking him for a great job on the analytics last week.*

***Padma** (Data Engineer): Fortunately, I was able to leverage a Pipeline developed a few months ago by the MDM team. We were even able to reuse most of their tests.*

***Chris** (DataOps Engineer): Once I set-up Kitchen creation, Padma was able to start being productive immediately. With matching release environments, we quickly migrated the new analytics from dev to production.*

***Eric** (Production Engineer): The tests are showing that all data remains within statistical limits. The dashboard indicators are all green.*

DataOps helps our band of frustrated and squabbling data professionals achieve a much higher level of overall team productivity by establishing processes and providing resources that support teamwork. With DataOps, two key performance parameters improve dramatically — the development cycle time of new analytics and quality of data and analytics code. We've seen it happen time and time again.

What's even more exciting is the business impact of DataOps. When users request new analytics and receive them in a timely fashion, it initiates new ideas and uncharted areas of exploration. This tight feedback loop can help analytics achieve its true aim, stimulating creative solutions to an enterprise's greatest challenges. Now that's teamwork!

# Eliminate Your Analytics Development Bottlenecks

## APPLYING THE THEORY OF CONSTRAINTS TO DATA ANALYTICS

Business users often have no concept of what it takes to design and deploy robust data analytics. The gap between expectations and execution is one of the main obstacles holding the analytics team back from delighting its users. Managers may ask for a simple change to a report. They don't expect it to take weeks or months.

Analytics teams need to move faster, but cutting corners invites problems in quality and governance. How can you reduce cycle time to create and deploy new data analytics (data, models, transformation, visualizations, etc.) without introducing errors? The answer relates to finding and eliminating the bottlenecks that slow down analytics development.



Figure 45: The creation of analytics in a large data organization requires the contribution of many groups.

## YOUR DEPLOYMENT PIPELINE

Analytics development in a large data organization typically involves the contribution of several groups. Figure 45 shows how multiple teams work together to produce analytics for the internal or external customer.

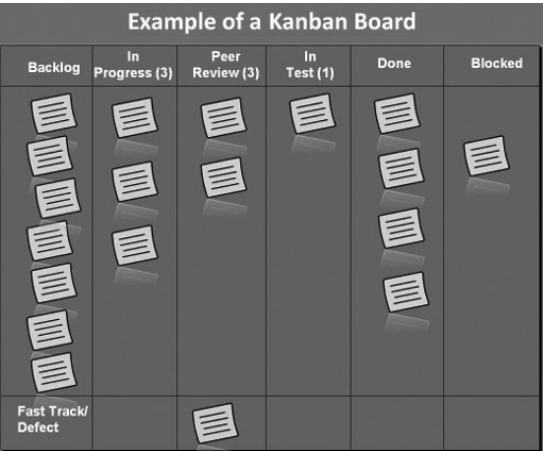


Figure 46: Example Kanban Board

Tasks in development organizations are often tracked using Kanban boards, tickets or project tracking tools. Figure 46 is a Kanban board, representing a project, with a yellow sticky note for each task. As tasks progress through milestones, they move from left to right until they reach the “Done” column.

Each of the groups shown in figure 45 tracks their own

projects. Figure 47 shows the data-analytics groups again, but each with their own Kanban boards to track the progress of work items. To serve the end goal of creating analytics for users, the data teams are desperately trying to move work items from the backlog (left column) to the done column at the right, and then pass it off to the next group in line.

Data professionals are smart and talented. They work hard. Why does it take so long to move work tickets to the right? Why does the system become overloaded with so many unfinished work items forcing the team to waste cycles context switching?



Figure 47: The development pipeline with Kanban boards

To address these questions, we need to think about the creation and deployment of analytics like a manufacturing process. The collective workflows of all of the data teams are a linked sequence of steps, not unlike what you would see in a manufacturing operation. When we conceptualize the development of new analytics in this way, it offers the possibility of applying manufacturing management tools that uncover and implement process improvements.

## THE THEORY OF CONSTRAINTS

One of the most influential methodologies for ongoing improvement in manufacturing operations is the [Theory of Constraints](#) (ToC), introduced by Dr. Eliyahu Goldratt in a business novel called “The Goal,” in 1984. The book chronicles the adventures of the fictional plant manager Alex Rogo who has 90 days to turn around his failing production facility. The plant can’t seem to ship anything on time, even after installing robots and investing in other improvements dictated by conventional wisdom. As the story progresses, our hero learns why none of his improvements have made any difference.

## THE BOTTLENECK

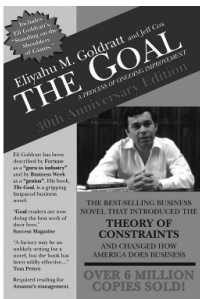


Figure 48

The plant’s complex manufacturing process, with its long sequence of interdependent stages, was throughput limited by one particular operation — a certain machine with limited capacity. This machine was the “*constraint*” or bottleneck. The Theory of Constraints views every process as a series of linked activities, one of which acts as a constraint on the overall throughput of the entire system. The constraint could be a human resource, a process, or a tool/technology.

In “The Goal,” Alex learned that “*an improvement at any point in the system, not at the constraint, is an illusion.*” An improvement made at a stage that feeds work to the bottleneck just increases the queue

of work waiting for the bottleneck. Improvements after the bottleneck will always remain starved. Every loss of productivity at the bottleneck is a loss in the throughput of the entire system. Losses in productivity in any other step in the process don't matter as long as that step still produces faster than the bottleneck.

Even though Alex's robots improved efficiency at one stage of his manufacturing process, they didn't alleviate the true system constraint. When Alex's team focused improvement efforts on raising the throughput of the bottleneck, they were finally able to increase the throughput of the overall manufacturing process. True, some of their metrics looked worse (the robot station efficiency declined), but they were able to reduce cycle time, ship product on time and make a lot more money for the company. That is, after all, the real "goal" of a manufacturing facility.

## FINDING YOUR BOTTLENECK

To improve the speed (and minimize the cycle time) of analytics development, you need to find and alleviate the bottleneck. This bottleneck is what is holding back your people from producing analytics at a peak level of performance. The bottleneck can often be identified using these simple indications:

- **Work in Progress (WIP)** – In a manufacturing flow, work-in-progress usually accumulates before a constraint. In data analytics, you may notice a growing list of requests for a scarce resource. For example, if it takes 40 weeks to [provision a development system](#), your list of requests for them is likely to be long.
- **Expedite** – Look for areas where you are regularly being asked to divert resources to ensure that critical analytics reach users. In data analytics, [data errors](#) are a common source of unplanned work.
- **Cycle Time** – Pay attention to the steps in your process with the longest cycle time. For example, some organizations take 6 months to shepherd 20 lines of SQL through the [impact review board](#). Naturally, if a step is starved or blocked by a dependency, the bottleneck is the external factor.
- **Demand** – Note steps in your pipeline or process that are simply not keeping up with demand. For example, often less time is required to create new analytics than to test and validate them in preparation for deployment.

## EXAMPLE BOTTLENECKS IN DATA ANALYTICS

You may notice a common theme in each of the example bottlenecks above. A bottleneck is especially problematic because it prevents people on the analytics team (analysts, scientists, engineers, ...) from fulfilling their primary function — creating new analytics. Bottlenecks distract them from high priority work. Bottlenecks redirect their energy to non-value add activities. Bottlenecks prevent them from implementing new ideas quickly.

When managers talk to data analysts, scientists and engineers, they can quickly discover the issues that slow them down. Figure 49 shows some common constraints. For example, data errors in analytics cause unplanned work that upsets a carefully crafted Kanban board. Work-in-progress (WIP) is placed on hold and key personnel context switch to address

the high-severity outages. Data errors cause the Kanban boards to be flooded with new tasks which can overwhelm the system. Formerly high priority tasks are put on hold, and management is burdened, having to manage the complexity of many more work items. Data errors also affect the culture of the organization. After a series of interruptions from data errors, the team becomes accustomed to moving more slowly and cautiously. From a Theory of Constraints perspective, data errors severely impact the overall throughput of the data organization.

A related problem, also shown in figure 49, occurs when deployment of new analytics breaks something unexpectedly. Unsuccessful deployments can be another cause of unplanned work which can lead to excessive caution, and burdensome manual operations and testing.

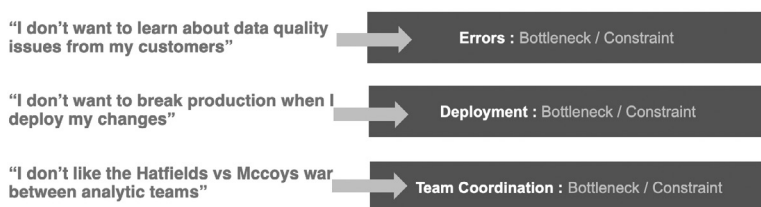


Figure 49: Translating problems to constraints

Another common constraint is [team coordination](#). The teams may all be furiously rowing the boat, but perhaps not in the same direction. In a large organization, each team's work is usually dependent on each other. The result can be a serialized pipeline. Tasks could be parallelized if the teams collaborated better. New analytics wouldn't break existing data operations with proper coordination between and among teams.

A wide variety of constraints potentially slow down analytics development cycle time. In development organizations, there are sometimes multiple constraints in effect. There is also variation in the way that constraints impact different projects. The following are some potential rate-limiting bottlenecks to rapidly deploying analytics:

- [Dependency on IT](#) to make schema changes or to integrate new data sets
- [Impact Review Board](#)
- Provisioning of development systems and [environments](#)
- Long [test](#) cycles
- Data [errors](#) causing unplanned work
- [Manual orchestration](#)
- Fear of breaking existing analytics
- Lack of [teamwork](#) among data engineers, scientists, analysts, and users
- [Long project cycles — deferred value](#)

When you have identified a bottleneck, the Theory of Constraints offers a methodology called the Process Of On-Going Improvement (POOGI) to address it. If you have many active bottlenecks that all need to be addressed, it may be more effective to focus on them one at a time. Below, we will suggest a method that we have found particularly effective in prioritizing projects.

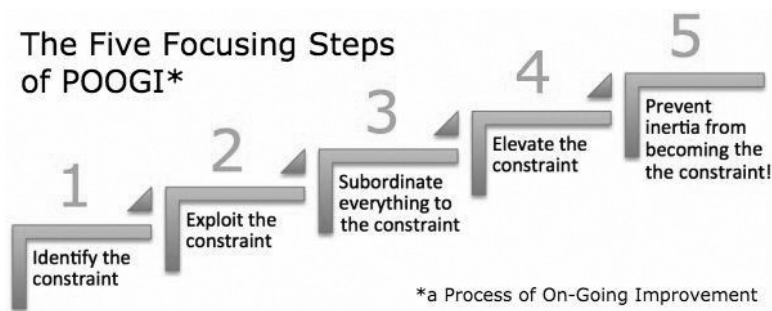


Figure 50: Source: Theory of Constraints Institute, Process of On-Going Improvement (POOGI)

### ALLEVIATING THE BOTTLENECK

Once identified, the Theory of Constraints recommends a five-step methodology to address the constraint:

1. Identify the constraint
2. Exploit the constraint – Make improvements to the throughput of the constraint using existing resources
3. Subordinate everything to the constraint – Review all activities and make sure that they benefit (or do not negatively impact) the constraint. Remember, any loss in productivity at the constraint is a loss in throughput for the entire system.

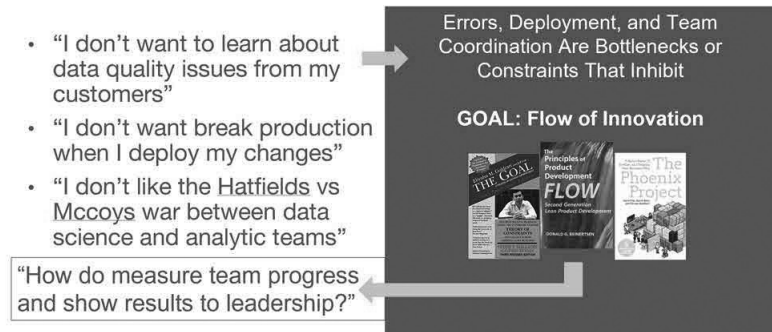


Figure 51: Errors, deployment and team coordination are bottlenecks that inhibit the flow of analytics innovation

4. Elevate the constraint – If after steps 2–3, the constraint remains in the same place, consider what other steps, such as investing resources, will help alleviate this step as a bottleneck
5. Prevent inertia from becoming a constraint by returning to step 1.

## THE THEORY OF CONSTRAINTS APPLIED TO IT



Figure 52

A leading book on DevOps, called “The Phoenix Project,” was explained by author Gene Kim to be essentially an adaptation of “The Goal” to IT operations. To alleviate their bottleneck, the team in the book implements [Agile development](#) (small lot sizes) and [DevOps](#) (automation). One important bottleneck was a bright programmer named Brent who was needed for every system enhancement and was constantly being pulled into unplanned work. When the team got better at relieving and managing their constraints, the output of the whole department dramatically improved.

## PRIORITIZING DATAOPS PROJECTS BASED ON DESIRED OUTCOMES

If you have identified multiple bottlenecks in your development process, it may be difficult to decide which one to tackle first. [DataOps](#) is a methodology that applies Agile, DevOps and [lean manufacturing](#) to data analytics. That’s a lot of ground to cover. One way to approach this question is to think like a product or services company.

The data organization creates analytics for its consumers (users, colleagues, business units, managers, ...). Think of analytics as your *product* and data consumers as your customers. Like any product or service organization, perhaps you should simply ask your *customers* what they want?



Figure 53: Many data professionals can relate to the experience of working diligently to deliver what customers say they want only to receive a lukewarm response.

The problem is that customers don’t actually know what products or services they want. What customer would have asked for Velcro or Post-It notes or Twitter? Many data professionals can relate to the experience of working diligently to deliver what customers say they want only to receive a lukewarm response.

There is much debate about how to listen to the voice of the customer ([Dorothy Leonard, Harvard Business School, The Limitations of Listening](#)). Customer preferences are reliable when you ask

them to make selections within a familiar product category. If you venture outside of the customer's experience, you tend to encounter two blocks. People fixate on the way that products are normally used, preventing them from thinking *outside the box*. Second, customers have seemingly contradictory needs. Your data-analytics customers want analytics to be error-free, which requires a lot of testing, but they dislike waiting for lengthy QA activities to complete. Data professionals might feel like they are in a no-win situation.

Management consultant [Anthony Ulwick](#) contends ([Harvard Business Review](#)) that you should not expect your customers to recommend solutions to their problems. They aren't expert enough for that. Instead, ask about desired *outcomes*. What do they want analytics to do for them? The customers might say that they want changes to analytics to be completed very fast so they can play with ideas. They won't tell you to implement automated orchestration or a data warehouse which can both contribute to that outcome.

The outcome-based methodology for gathering customer input breaks down into five steps.

## STEP 1 – PLAN OUTCOME-BASED CUSTOMERS INTERVIEWS

Deconstruct, step by step, the underlying processes behind your delivery of data analytics. It may make sense to interview users like data analysts who leverage data to create analytics for business colleagues.

## STEP 2 – CONDUCT INTERVIEWS

Pay attention to desired outcomes not recommended solutions. Translate solutions to outcomes by asking what benefit the suggested feature/solution provides. Participants should consider every aspect of the process or activity they go through when creating or consuming analytics. A good way to phrase desired outcomes is in terms of the type (minimize, increase) and quantity (time, number, frequency) of improvement required. Experts in this method report that 75% of the customers' desired feedback is usually captured in the first two-hour session.

## STEP 3 – ORGANIZE THE DATA

Collect a master list of outcomes, removing duplicates and categorize outcomes into groups that correspond to each step in the process

## STEP 4 – RATE THE OUTCOMES

Conduct a quantitative survey to determine the importance of each desired outcome and the degree to which the outcome is satisfied by the current solution. Ask customers to rate, on a scale of 1–10, the importance of each desired outcome (Importance) and the degree to which it is currently satisfied (Satisfaction). These factors are input into the [opportunity algorithm](#) below which helps rate outcomes based on potential.

The opportunity algorithm makes use of a simple mathematical formula to estimate the potential opportunity associated with a particular outcome:

**Opportunity = Importance + (Importance - Satisfaction)**

*Note that if Satisfaction is greater than Importance, then the term (Importance - Satisfaction) is zero not negative.*

When you are done, you should have produced something like the below example.

| Desired Outcome  | Importance | Satisfaction | Opportunity |
|--|------------|--------------|-------------|
| Minimize data errors                                     | 9.5        | 3.2          | 15.8        |
| Release new analytics (iterations) weekly                | 8.3        | 4.2          | 12.4        |
| Provision development environments within 2 days         | 9.5        | 7.5          | 11.5        |
| Test new analytics and approve deployment within 8 hours | 9.1        | 8.4          | 9.8         |
| Minimize time of impact review of new analytics          | 5.1        | 1.0          | 9.2         |
| Minimize time to make schema changes                     | 7.7        | 6.6          | 8.8         |

Table 3: Desired outcomes ranked by opportunity strength

## STEP 5 – GUIDE INNOVATION

The table above reveals which outcomes are important to users and deprecates those outcomes that are already well served by the existing analytics development process. The outcomes which are both important and unsatisfied will rise to the top of the priority list. This data can be used as a guide to prioritize process improvements in the data analytics development pipeline and process.

## THE PATH FORWARD FOR DATAOPS

DataOps applies manufacturing management methods to data analytics. One leading method, the Theory of Constraints, focuses on identifying and alleviating bottlenecks. Data analytics can apply this method to address the constraints that prevent the data analytics organization from achieving its peak levels of productivity. Bottlenecks lengthen the cycle time of developing new analytics and prevent the team from responding quickly to requests for new analytics. If these bottlenecks can be improved or eliminated, the team can move faster, developing and deploying with a high level of quality in record time.

If you have multiple bottlenecks, you can't address them all at once. The opportunity algorithm enables the data organization to prioritize process improvements that produce outcomes that are recognized as valued by users. It avoids the requirement for users to understand the technology, tools, and processes behind the data analytics pipeline. For DataOps proponents, it can provide a clear path forward for analytics projects that are both important and appreciated by users.

# Prove Your Awesomeness with Data: The CDO DataOps Dashboard

Do you deserve a promotion? You may think to yourself that your work is exceptional. Could you prove it?

As a Chief Data Officer (CDO) or Chief Analytics Officer (CAO), you serve as an advocate for the benefits of [data-driven decision making](#). Yet, many CDO's are surprisingly unanalytical about the activities relating to their own department. Why not use analytics to shine a light on yourself?

Internal analytics could help you pinpoint areas of concern or provide a big-picture assessment of the state of the [analytics team](#). We call this set of analytics the *CDO Dashboard*. If you are as good as you think you are, the CDO Dashboard will show how simply awesome you are at what you do. You might find it helpful to share this information with your boss when discussing the data analytics department and your plans to take it to the next level. Below are some reports that you might consider including in your *CDO dashboard*:



## BURNDOWN CHART

The [burndown chart](#) graphically represents the completion of backlog tasks over time. It shows whether a team is on schedule and sheds light on the productivity achieved in each development [iteration](#). It can also show a team's accuracy in forecasting its own schedule.

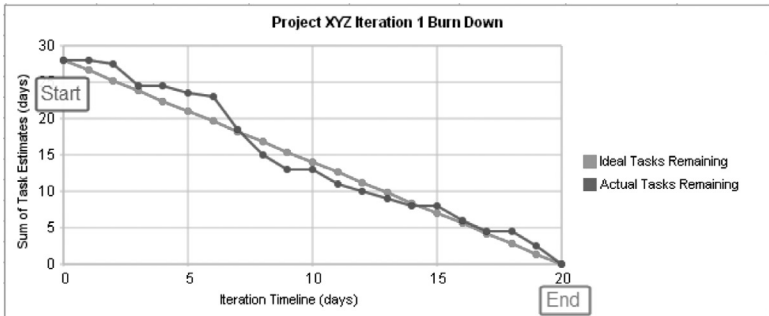


Figure 54: Sample Burndown chart

## VELOCITY CHART

The [velocity chart](#) shows the amount of work completed during each sprint — it displays how much work the team is doing week in and week out. This chart can illustrate how improved processes and indirect investments (training, tools, process improvements, ...) increase velocity over time.

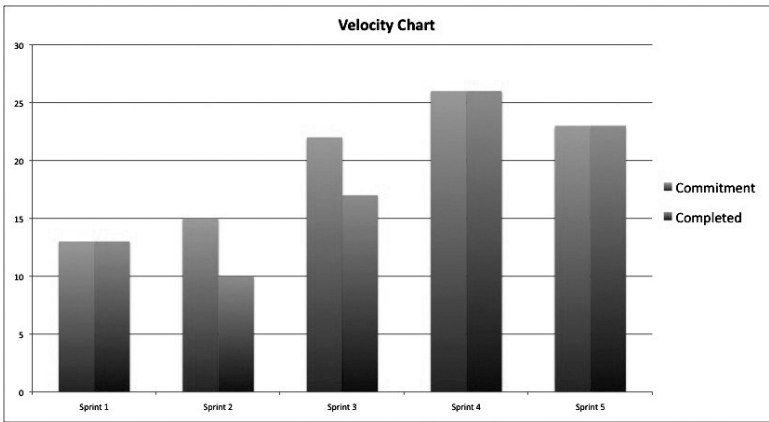


Figure 55: Sample Velocity chart

# TORNADO REPORT

The [Tornado Report](#) is a stacked bar chart that displays a weekly representation of the operational impact of production issues and the time required to resolve them. The Tornado Report provides an easy way to see how issues impacted projects and development resources.

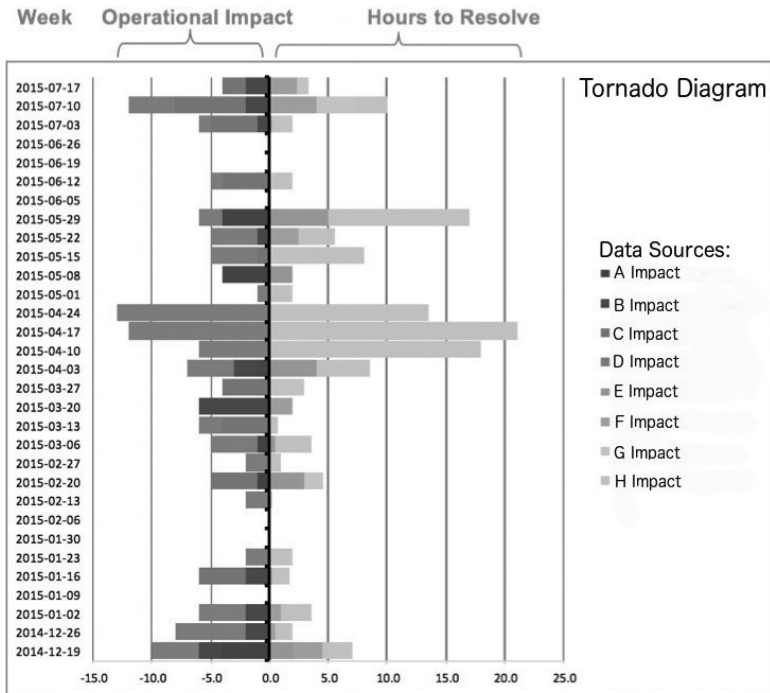


Figure 56: Sample Tornado report

# DATA ARRIVAL REPORT

A large organization might receive hundreds of data sets from suppliers and each one could represent dozens of files. All of the data has to arrive error-free in order to, for example, build the critical Friday afternoon report. The Data Arrival report tracks how vendors perform relative to their respective service level agreements (SLA).

The Data Arrival report enables you to track data suppliers and quickly spot delivery issues. Any partner that causes repeated delays can be targeted for coaching and management. The Tornado Report mentioned above can help quantify how much time is spent managing these issues in order to articulate impact. These numbers are quite useful when coaching a peer organization or vendor to improve its quality.

|         | Source 1 | Source 2 | Source 3 | Source 4 | Source 5 |
|---------|----------|----------|----------|----------|----------|
| 3/13/16 |          |          |          |          |          |
| 3/12/16 |          |          |          |          |          |
| 3/11/16 |          |          |          |          |          |
| 3/10/16 |          |          |          |          |          |
| 3/9/16  |          |          |          |          |          |
| 3/8/16  |          |          |          |          |          |
| 3/7/16  |          |          |          |          |          |
| 3/6/16  |          |          |          |          |          |
| 3/5/16  |          |          |          |          |          |
| 3/4/16  |          |          |          |          |          |
| 3/3/16  |          |          |          |          |          |
|         |          |          |          |          |          |
|         | Key:     |          | missing  |          |          |
|         |          |          | late     |          |          |
|         |          |          | on time  |          |          |

Figure 57: Sample Data Arrival Report

### TEST COVERAGE AND INVENTORY

The Test Coverage and Inventory Reports show the degree of test coverage of the data analytics pipeline. It shows the percent of tables and data covered by tests and how test coverage improves over time. The report can also provide details on each test. In a [DataOps](#) enterprise, results from tests run on the production pipeline are linked to real-time alerts. If a process fails with an error, the analytics team can troubleshoot the problem by examining test coverage before or after the point of interest.

# STATISTICAL PROCESS CONTROLS

The data analytics pipeline is a complex process with steps often too numerous to be monitored manually. [Statistical Process Control](#) (SPC) allows the data analytics team to monitor the pipeline end-to-end from a big-picture perspective, ensuring that everything is operating as expected.

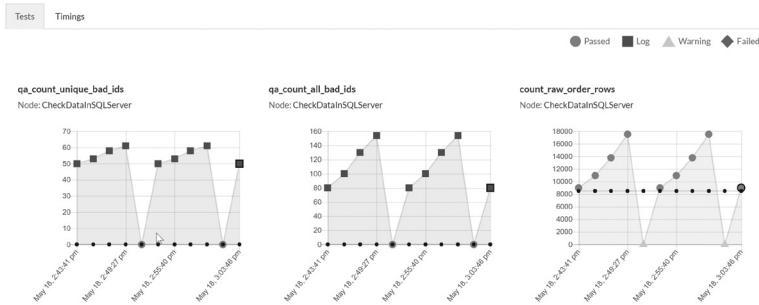


Figure 58: Sample Statistical Process Controls

# NET PROMOTER SCORE

A [Net Promoter](#) Score is a customer satisfaction metric that gauges a team's effectiveness. For a data team, this is often a survey of internal users who are served by analytics. The Net Promoter Score can show that the data analytics team is effective at meeting the needs of its internal customer constituency or that satisfaction is improving.

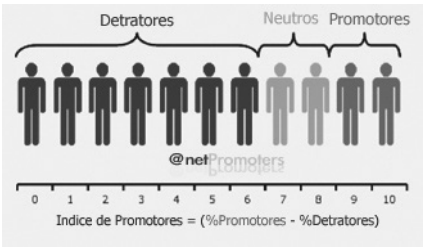


Figure 59: Net Promoter Score

One of the main goals of analytics is to improve decision-making. The CDO Dashboard puts information at the fingertips of executives, so they have a complete picture of what is happening in the data analytics domain. When it's time to review performance, the CDO Dashboard can help you show others that the analytics department is a well-oiled machine. *Now, about that promotion...*

# Surviving Your Second Year as CDO

As the Chief Data Officer (or Chief Analytics Officer) of your company, you manage a team, oversee a budget and hold a mandate to set priorities and lead organizational change. The bad news is that everything that could possibly go wrong from a security, governance and risk perspective is your responsibility. If you do a perfect job, then no one on the management team ever hears your name.

The average tenure of a CDO or CAO is about 2.5 years. In our conversations with data and analytics executives, we find that CDOs and CAOs often fall short of expectations because they fail to add sufficient value in an acceptable time frame. If you are a CDO looking to survive well beyond year two, we recommend avoiding three common *traps* that we have seen ensnare even the best and brightest.



## 1) THE TRAP OF DATA DEFENSE

Babson College professor Tom Davenport classifies data and analytics projects as either [defense or offense](#). Data defense seeks to resolve issues, improve efficiency or mitigate risks. [Data quality](#), security, privacy, governance, compliance — these are all critically important endeavors, but they are in essence, just enabling activities. You could think of data defense as providing *indirect* value.

Data offense expands top-line revenue, builds the brand, grows the company and in general puts *points on the board*. Using data analytics to help marketing and sales is data offense. Companies may acknowledge the importance of defense, but they care passionately about offense and focus on it daily. Data offense provides the organization with *direct* value and it is *what gets CDOs and CAOs promoted*.

The challenge for a CDO is that data defense is hard. A company's shortcomings in governance, security, privacy, or compliance may be glaringly obvious. In some cases, new regulations like [GDPR](#) (General Data Protection Regulation, EU 2016/679) demand immediate

action. Data defense has a way of consuming more than its fair share of attention and staff. If not put in perspective, data defense is a trap that can divert the CDO's attention and resources away from offensive activities that create value for the organization.

## 2) THE TRAP OF DEFERRED VALUE

Projects that implement new platforms and solutions can require months, if not years, of integration and oversight. If conceived as a waterfall project, with a *big-bang* deliverable at the end, these projects produce little to no value until they are complete. We call this the trap of *deferred value*, and it is possibly the main reason that many CDOs never make it past year three of their tenure.

In a fast-paced, competitive environment, an 18-month integration project can seem like the remote future. Also, success is uncertain until you deliver. Your C-level peers know that big software integration projects fail half the time. Projects frequently turn out to be more complex than anticipated, and they often miss the mark. For example, you may have thought you needed ten new capabilities, but your internal customers only really require seven, and two of them were not on your original list. The issue is that you won't know which seven features are critical until around the time of your second annual performance review and by then it might be too late to right the ship.

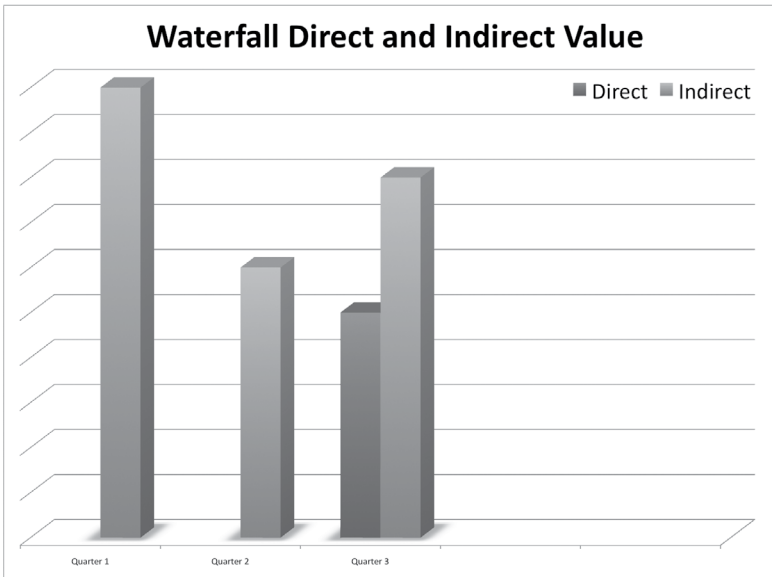


Figure 60: CDO's often make the dual mistake of (1) focusing too much on delivering indirect value (governance, security, privacy, or compliance, ...) and (2) using a waterfall project methodology which defers the delivery of value to the end of a long project cycle. In the case shown, it takes several months to deliver direct value

### 3) THE TRAP OF DATA VALUATION

Industry analysts and the media have long touted the strategic value of data. Following the advice of [analysts](#), a CDO may decide to embark on a project to quantify the monetary value of the company's data. This seems like a worthy endeavor that some say should attain a high level of visibility.

A data valuation project can take months of effort and consumes the attention of the CDO and her staff on what is essentially an internally-focused, intellectual exercise. In the end, you have a beautiful PowerPoint presentation with detailed spreadsheets to back it up. *Your data has tremendous value that can and should be carried on the balance sheet.* You tell everyone all about it — why don't they care?

Don't confuse data valuation with data offense. Knowing the theoretical value of data is not data offense. While data valuation may be useful and important in certain cases, it is often a distraction. All of the time and resources devoted to creating and populating the valuation model could have been spent on higher value-add activities.

#### DIRECT VS. INDIRECT VALUE

Investments in data analytics can create value either directly or indirectly. Sales growth is an example of direct value. Indirect value lays the foundation for future growth and productivity. In both cases, value is delivered either quickly or in a longer time frame. One common mistake is to focus too heavily on indirect-value, long-term projects.

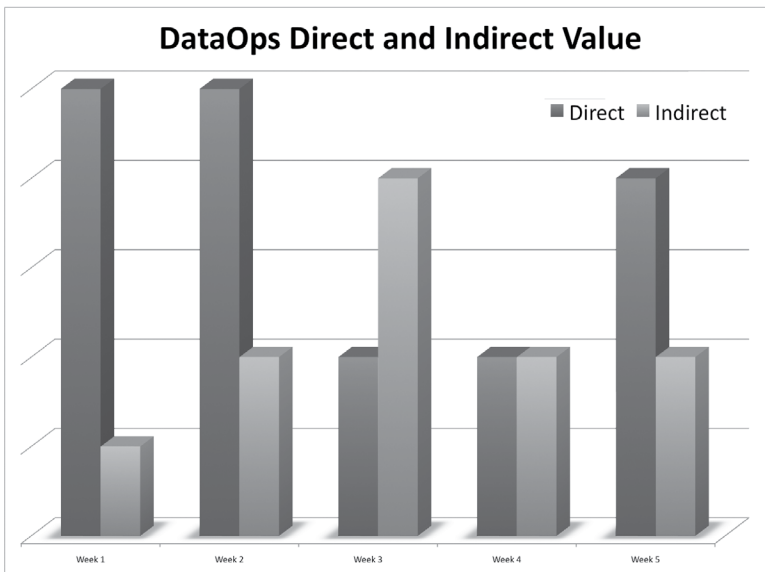


Figure 61: DataOps uses an iterative product management methodology (Agile development) that enables the CDO to rapidly deliver direct value (growing the top line).

That's not to say indirect or long-term projects don't have their place. They can be important and worthwhile. For example, a CDO may wisely invest in employee training or building technical infrastructure. It's essential to create the right mix, investing in enough indirect value-creators to support long-term growth and enough direct-value and short-term projects to maintain a high level of visibility.

## **DATAOPS ACCELERATES VALUE CREATION**

The trick is to reorganize the data and analytics teams to be responsive and adaptable to the needs of internal customers and users. [DataOps](#) can help here. DataOps subscribes to an Agile, iterative approach. Deliver something of value in a few weeks and build on that in successive intervals. DataOps combines Agile with [DevOps](#) and lean manufacturing methods to provide a data and analytics team with the processes and tools needed to accelerate the value-creation cycle. *Raise a glass to year two!*

# CAOs and CDOs: Earn the Trust of your CEO

One of the greatest challenges in analytics is earning the trust of your organization's CEO and management team. A lot of people in the business world make decisions with their gut. They rely on experience and intuition, but many companies would prefer to depend upon data. You cannot walk through an airport these days and not see some version of an advertisement saying *we are the company who is going to help you to be more data-driven*.

People do not always trust data. Imagine you are an executive and an employee walks into your office and shows you charts and graphs that contradict strongly held assumptions about your business. A lot of managers in this situation favor their own instincts. Data-analytics professionals, who tend to be doers, not talkers, are sometimes unable to convince an organization to trust its data.



## BUILDING TRUST IN THE DATA

We've discovered that the best way for data-analytics professionals to build trust with their management team is to deliver value consistently, quickly and accurately. To accomplish this, you need to create and publish analytics in a new way. We call this new approach [DataOps](#). DataOps is a combination of tools and methods, which streamline the development of new analytics while ensuring impeccable data quality. DataOps helps shorten the cycle time for producing analytic value and innovation, addressing some of the fundamental challenges that prevent organizations from trusting their data.

## EARN TRUST BY DELIVERING A JOURNEY OF VALUE

DataOps uses the [Agile Development](#) methodology to create valuable new analytics for the business or organization. Agile accepts that people don't necessarily know what they want until they see it. The data-analytics team delivers new analytics in a short time frame and receives immediate feedback from users. This tight feedback loop steers the development of new analytics to the features that are most valuable to the business. Users aren't expected to take a *leap of faith*. They are gradually introduced to their own data and direct the journey

of each future improvement in analytics through their feedback. This feedback both improves the analytics and draws them into the process as an active and invested stakeholder. When users grow to appreciate the value provided, it is time to operationalize the analytics and deliver them on a continuous basis.

## EARN TRUST BY DELIVERING QUICKLY

In [DataOps](#) automated tests enable the data-analytics team to deploy new analytics quickly and with confidence. Minimizing the cycle time of new analytics is critical to earning the trust of users. The relevance of a question asked, at any given moment, decays rapidly as the situation facing the organization quickly evolves. Customers and prospects do not stand still. If analytics take too long, frustration builds, and the answer to a question might be delivered long after the question has ceased to be relevant.

DataOps relies upon the [data lake design pattern](#), which enables data analytics teams to update schemas and transforms quickly to create new [data warehouses](#) that promptly address pressing business questions. DataOps incorporates the [continuous-deployment](#) methodology that is characteristic of [DevOps](#). This reduces the cycle time of new analytics by an order of magnitude. When users get used to quick answers, it builds trust in the data-analytics team, and stimulates the type of creativity and teamwork that leads to breakthroughs.

## EARN TRUST BY DELIVERING ACCURATELY

Trust is tough to earn and easy to squander. If bad data makes its way through your data pipeline, the users might not ever again trust the data. DataOps tests and monitors business logic and data validity by testing data at each stage of the data-analytics pipeline. We liken this testing to the statistical process control used in lean manufacturing. The tests can start simple, but over time they are expanded in breadth until they become a formidable check on quality. If a problem occurs with an internal or external dataset, or at any processing stage, the data-analytics team will be alerted immediately by the automated tests. DataOps protects the integrity of the data, so the data itself is worthy of user trust.

## THE BENEFIT OF TRUSTED ANALYTICS

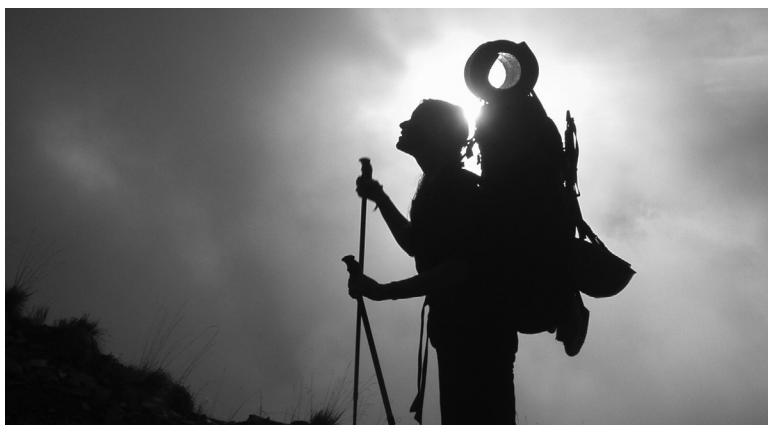
Earning your organization's trust makes the job of the data-analytics team a lot easier, but much more is at stake. Companies that don't trust their data will be outcompeted in the marketplace. Managers will make decisions based on instinct, past experience or preconceived notions. In cases like this, managers sometimes develop different versions of reality and can't agree on the facts, let alone strategic plans.

A company that trusts its data develops a unified view of reality and can formulate a shared vision of how to achieve its goals. Data-driven companies deliver higher growth and ultimately higher valuations than their peers. As a CAO or CDO, leading the organization to become more data-driven is your mission. DataOps makes that easier by helping the data-analytics team deliver quickly and robustly, creating value that is recognized and trusted by the organization.

# The Four Stage Journey to Analytics Excellence

First, we walk, then we run. The same is true in data analytics. In our many discussions, we have encountered companies that are just starting out with data analytics and others with substantial organizations handling petabytes of data. Everyone that we meet is somewhere along this spectrum of maturity. We've found that just because an enterprise's data analytics organization is large does not mean that it is *excellent*. In fact, the flaws in a process or methodology become particularly noticeable when a team grows beyond the initial stages.

We view every company as being somewhere on a journey towards achieving excellence. In our experience, the journey is divided into four stages. That said, some get there faster by taking a shortcut. We'll discuss the four-stage journey and the shortcut to excellence below.



## STAGE 1 - DATA DESERT

Companies generate data from a variety of enterprise applications. This data can help organizations gain a better understanding of customers, products, and markets. If your company is not reaping value from your data, then you live in a data desert. In a data desert, the data is underutilized or lays dormant. Like a mineral resource that remains in the ground, the data could have enormous potential, but without data analytics that potential goes unrealized.

This situation could have implications for the company's future. What if competitors have devised a way to use data analytics to garner a competitive advantage? Without a comprehensive data strategy, a company risks missing the market.

## STAGE 2- BOUTIQUE ANALYTICS

Some organizations are engaged in analytics but do so in a decentralized fashion or on a small scale. Some enterprises are just getting started in analytics. Whether or not a person has programming skills, it is possible to do a fair amount of analytics using everyday tools like spreadsheets. One can accomplish even more using data visualization software. We call this Boutique Analytics. In a boutique shop, data analytics professionals are akin to artisans.

Boutique Analytics tend to be ad hoc or create one-off reports that answer questions posed by a manager. For example, a global enterprise may wish to know how much of its revenue it derives from one customer. Data is exported from CRMs or operations systems and pulled into a spreadsheet for analysis. The term Boutique Analytics may make it sound small in scale, but some large enterprises are known to rely solely upon this approach. A large enterprise might run weekly reports exporting sales data into a flat file. The global sales and marketing team can then easily manipulate the data in a spreadsheet. The sharing of data using flat files can be used to complement an enterprise's operational analytics.

There is nothing inherently wrong with Boutique Analytics. It is a great way to explore the best ways to deliver value based on data. The eventual goal should be to operationalize the data and deliver that value on a regular basis. This can be time-consuming and error-prone if executed manually.

## STAGE 3- WATERFALL ANALYTICS

If an analytics initiative is successful and the team grows, a company will eventually begin to manage analytics more formally. Companies usually have a deeply entrenched project management culture based upon the methodology used by their research and development teams. Often project management is based on the [Waterfall](#) method so it is natural for these organizations to implement Waterfall Analytics.

In the Waterfall world, development cycles are long and rigidly controlled. Projects pass through a set of sequential phases: architecture, design, test, deployment, and maintenance. Changes in the project plan at any stage cause modifications to the scope, schedule or budget of the project. As a result, Waterfall projects are resistant to change. This is wholly appropriate when you are building a bridge or bringing a new drug to market, but in the field of data analytics, changes in requirements occur on a continuous basis. Teams that use Waterfall analytics often struggle with development cycle times that are much longer than their users expect and demand. Waterfall analytics also tends to be labor intensive, which makes every aspect of the process slow and susceptible to error. Most data-analytics teams today are in the Waterfall analytics stage and are often unaware that there is a better way.

| Stage | Name                       | Model   | Data Pipeline     | Cycle Time |
|-------|----------------------------|---|-------------------|------------|
| 1     | <b>Data Desert</b>         | N/A   | None              | N/A        |
| 2     | <b>Boutique Analytics</b>  | Individual Artisan                            | One-Off or Ad Hoc | Custom     |
| 3     | <b>Waterfall Analytics</b> | Waterfall Project Management                  | Manual Process    | Long       |
| 4     | <b>DataOps Analytics</b>   | Agile Development, DevOps, Lean Manufacturing | Automated Process | Short      |

Table 4: Four Stages of Analytics

## STAGE 4 - DATAOPS ANALYTICS

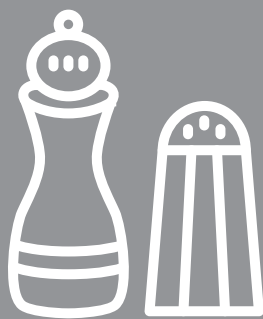
[DataOps](#) is a new approach to data analytics, which is superior to Waterfall Analytics in terms of flexibility, quality, and development cycle time. DataOps adopts key concepts from lean manufacturing. It views data analytics as a continuously operating pipeline, which can be automated, monitored and controlled. New analytics are created using [Agile Development](#), a methodology created in the software engineering field. Agile manages the development of new analytics by delivering valuable features in short increments. This allows an organization to quickly adapt to new requirements or change course based on the demands of the marketplace. Analytics are deployed using the [continuous deployment](#) methodology pioneered by DevOps. Automated orchestration replaces labor-intensive manual processes. This means that new analytics can be published continuously, on-demand with minimal human intervention. [Data quality](#) flowing through the data analytics pipeline is monitored using automated [data and logic tests](#) executed as part of the continuous deployment automation. These tests are inspired by the statistical process control widely used in modern manufacturing operations.

## TAKE THE SHORTCUT

The mistake that many companies make is that they languish in stage 3. The better approach is to take a shortcut, skip stage 3 entirely, and move directly to stage 4. If your organization is already in stage 3, then it's advantageous to advance as quickly as possible.

## THE JOURNEY TO EXCELLENCE

DataOps provides the foundation for data analytics excellence. It streamlines the development of new analytics, shortens cycle time, and automates the data-analytics pipeline, freeing the team to focus on value-adding activities. It also controls the quality of data flowing through the pipeline so users can trust their data. With DataOps in place, the team is productive, responsive and efficient. They will race far ahead of competitors whose analytics are less nimble and less impactful. DataOps shortens your journey to analytics excellence.



# DataOps Healthy Hearty Banana Oatmeal Bread

*by Andrew Sadoway*

## INGREDIENTS AND TOOLS

- 1 1/4 cups whole wheat flour
- 1 1/4 old-fashioned oats
- 1 teaspoon baking powder
- 1/2 teaspoon baking soda
- 1/4 teaspoon salt
- 1 teaspoon cinnamon
- Dash of nutmeg (approx. 1/8 teaspoon)
- 3 medium-*largeish* bananas, mashed (defrosted from freezer OK)
- 1/2 cup low fat plain yogurt + 1 teaspoon vanilla
- 2 tablespoons honey, 1/3 cup brown sugar
- 1 large egg
- Splash of milk, as needed
- 3/4 cup dried cranberries
- 3/4 cup chopped walnuts + 1/4 cup chopped walnuts

## INSTRUCTIONS

Preheat oven to 350. Combine dry ingredients (flour through nutmeg) in a small bowl. In a separate bowl, mix together yogurt, vanilla, brown sugar and honey. Add egg. Add mashed up bananas. Slowly fold dry ingredients into wet. Stir in cranberries and 3/4 cup walnuts gently. Pour mixture into buttered loaf pan. Sprinkle remaining walnuts on top of loaf. Bake about 45 minutes, or until lightly browned and knife comes out clean.

# DataOps for the Data Engineer and the Data Scientist

## A Great Model is Not Enough: Deploying AI Without Technical Debt

### DATAOPS IN DATA SCIENCE AND MACHINE LEARNING

AI and Data Science are *all the rage*, but there is a problem that no one talks about. Machine learning tools are evolving to make it faster and less costly to develop AI systems. But deploying and maintaining these systems over time is getting exponentially more complex and expensive. Data science teams are incurring enormous [technical debt](#) by deploying systems without the processes and tools to maintain, monitor and update them. Further, poor quality data sources create [unplanned work](#) and cause errors that invalidate results.

A couple of years ago, Gartner predicted that *85 percent of AI projects would not deliver for CIOs*. Forrester affirmed this unacceptable situation by stating that *75% of AI projects underwhelm*. We can't claim that AI projects fail only for the reasons we listed. We can say, from our experience working with data scientists on a daily basis, that these issues are real and pervasive. Fortunately, data science teams can address these challenges by applying lessons learned in the software industry.

### TRADITIONAL MODEL DEVELOPMENT VERSUS AI

AI is most frequently implemented using machine learning (ML) techniques. Building a model is different than traditional software development. In traditional programming, data and code are input to the computer which, in turn, generates an output. This is also true of traditional modeling where a hand-coded application (the model) is input to a computer, along with data, to generate results.

In machine learning, the code can learn. The ML application trains the model using data and target results. An ML model developer feeds training data into the ML application, along with correct or expected answers. Errors are then fed back into the learning algorithm to boost

the model's precision. This procedure continues until the model reaches the target level of accuracy. When complete, this process generates a set of parameters that are described by a set of files (code and configuration). In production, the model evaluates input data and generates results. Figure 62 shows the different processes governing traditional and ML model development.

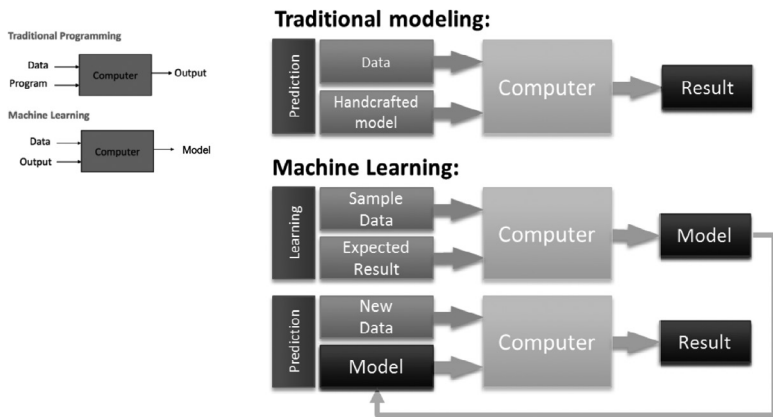


Figure 62: Traditional model programming versus machine learning model development

Below, Figure 63 further elaborates on the complex set of steps that are involved in model building. Naturally, AI projects begin with a business objective. Data is often imperfect so the team has to clean, prepare, mask, normalize and otherwise manipulate data so that it can be used effectively. Feature extraction identifies metrics (measured values) that are informative and facilitate training. After the building and evaluation phases (see Figure 63), the model is deployed, and its performance is monitored. When business conditions or requirements change, the team heads back to the lab for additional training and improvements. This process continues for as long as the model is in use.

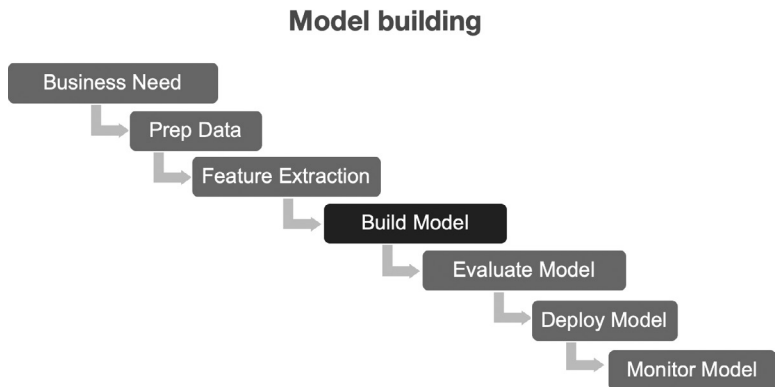
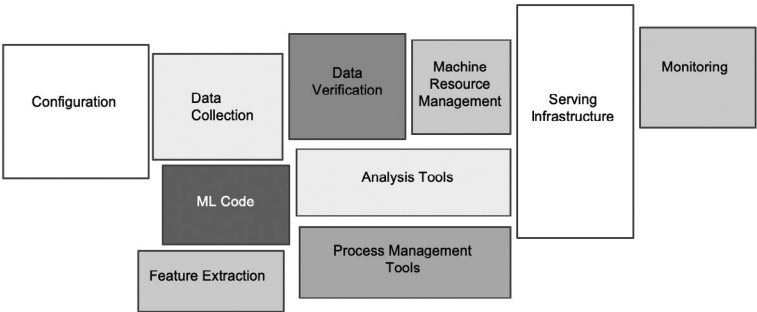


Figure 63: The complex sequence of steps in building an ML model

AI development and deployment is a complex workflow. If executed manually, it is slow, error-prone and inflexible. The actual output of the model development process (a set of files, scripts, parameters, source code, ...) is only a small fraction of what it takes to deploy and maintain a model successfully. Figure 64 below shows the *Machine Learning Code* in a system context. Notice that the ML code is only a small part of the overall system.



Source: *Hidden Technical Debt in Machine Learning Systems*, *Advances in Neural Information Processing Systems 28* (NIPS 2015)

Figure 64: Machine Learning Code is only a small part of the overall system.

Model creation and deployment commonly use the tools shown in Figure 65. Note that this is only a portion of what is required by the system in Figure 64. If the responsibility for these processes and toolchains falls on the data science team, they can end up spending the majority of their time on data cleaning and data engineering. Unfortunately, this is all too common in contemporary enterprises. Addressing this situation requires us to take a holistic view of the value pipeline and analytics creation.

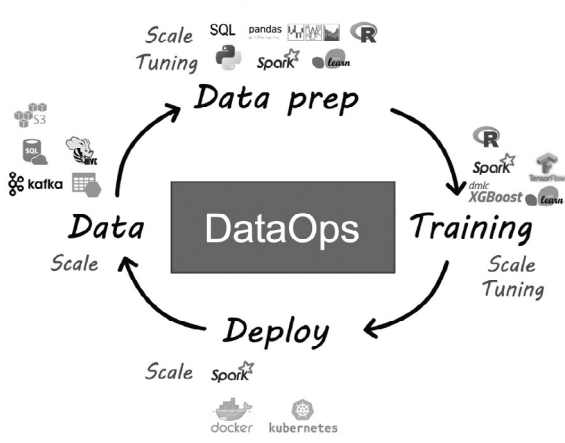
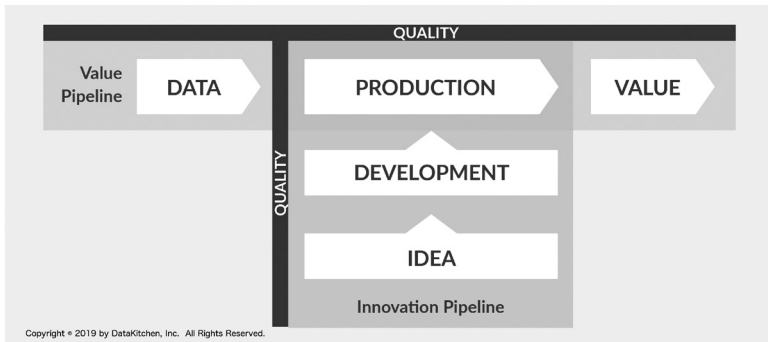


Figure 65: Model development and deployment requires a complex toolchain

## TWO JOURNEYS / TWO PIPELINES

We conceptualize AI (and all data analytics) as two intersecting pipelines. In the first pipeline, data is fed into AI and ML models producing analytics that deliver value to business stakeholders. For example, an ML model reviews credit card purchases and identifies potential fraud. We call this the “[Value Pipeline](#).”

The second pipeline is the process for new model creation — see Figure 62 and Figure 63. In the development pipeline, new AI and ML models are designed, tested and deployed into the Value Pipeline. We call this the “[Innovation Pipeline](#).” Figure 66 depicts the Value and Innovation Pipelines intersecting in production.



**Figure 66: The Value and Innovation Pipelines**

Conceptually, each pipeline is a set of stages implemented using a range of tools. The stages may be executed serially, parallelized or contain feedback loops. In terms of artifacts, the pipeline stages are defined by files: scripts, source code, algorithms, html, configuration files, parameter files, containers and other files. From a process perspective, all of these artifacts are essentially just *source code*. Code controls the entire data-analytics pipeline from end to end: ideation, design, training, deployment, operations, and maintenance.

When discussing code and coding, data scientists who create AI and ML models, often think *“this has nothing to do with me.”* I am a data analyst/scientist, not a coder. I am an ML tool expert. In process terms, what I do is just a sophisticated form of configuration. This is a common misconception and it leads to technical debt. When it is time for that debt to be paid, the speed of new analytics development (cycle time) will slow to a crawl.

## AI / ML / DATA SCIENCE WORK IS JUST CODE

Tools vendors have a business interest in perpetuating the myth that if you stay within the well-defined boundaries of their tool, you are protected from the complexity of software development. This is ill-considered albeit well-meaning.

Don't get us wrong. We love our tools, but don't buy into this falsehood. The \$50+ billion AI market is divided into two segments: "tools that create code" and "tools that run code." The point is — AI is code. The data scientist creates code and must own, embrace and manage the complexity that comes along with it.

## LESSONS LEARNED — DATAOPS

The good news is that when AI is viewed through a different lens, it can leverage the same processes and methodologies that have boosted the productivity of software engineering 100x in the last decades. We call these techniques (collectively) [DataOps](#). It includes three important methodologies: [Agile Software Development](#), [DevOps](#) and [statistical process controls](#) (SPC).

- Studies show that software development projects complete significantly faster and with far fewer defects when **Agile Development**, an iterative project management methodology, replaces the traditional Waterfall sequential methodology. The Agile methodology is particularly effective in environments where requirements are quickly evolving — a situation well known to data science professionals. Some enterprises understand that they need to be more Agile and that's great. (*Here's your chance to learn from the mistakes of many others.*) You won't receive much benefit from Agile if your quality is poor or your deployment and monitoring processes involve laborious manual steps. "Agile development" alone will not make your team more "agile."
- **DevOps**, which inspired the name DataOps, focuses on [continuous delivery](#) by leveraging on-demand IT resources and by automating test and deployment of code. *Imagine clicking a button in order to test and publish new ML analytics into production.* This merging of software development and IT operations reduces time to deployment, decreases time to market, minimizes defects, and shortens the time required to resolve issues. Borrowing methods from DevOps, DataOps brings these same improvements to data science.
- Like lean manufacturing, DataOps utilizes **statistical process control** (SPC) to monitor and control the Value Pipeline. When SPC is applied to data science, it leads to remarkable improvements in efficiency and quality. With SPC in place, the data flowing through the operational system is verified to be working. If an anomaly occurs, the data team will be the first to know, through an automated alert. Dashboards make the state of the pipeline transparent from end to end.
- DataOps eliminates technical debt and improves quality by orchestrating the Value and Innovation Pipelines. It catches problems early in the data life cycle by implementing tests at each pipeline stage. Further, it greatly accelerates the development of new AI, enabling the data science team to respond much more flexibly to changing business conditions.

## DATAOPS FOR YOUR AI AND ML PROJECT

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. DataOps is not any [specific vendor's solutions](#). It leverages automation, tools and other best practices.

You can implement DataOps for your AI and ML project yourself by following these seven steps:

## Step 1 — Add Data and Logic Tests

To be certain that the data analytics pipeline is functioning properly, it must be tested. Testing of inputs, outputs, and business logic must be applied at each stage of the data analytics pipeline. Tests catch potential errors and warnings before they are released so the quality remains high. Manual testing is time-consuming and laborious. A robust, automated test suite is a key element in achieving continuous delivery, essential for companies in fast-paced markets.

## Step 2 — Use a Version Control System

All of the files and processing steps that turn raw data into useful information are *source code*. All of the artifacts that data scientists create during ML development are just source code. These files control the entire data-analytics pipeline from end to end in an automated and reproducible fashion. When these file artifacts are viewed as code, they can be managed liked code.

In so many cases, the files associated with analytics are distributed in various places within an organization without any governing control. A revision control tool, such as Git, helps to store and manage all of the changes to code. It also keeps code organized, in a known repository and provides for disaster recovery. Revision control also helps software teams parallelize their efforts by allowing them to *branch and merge*.

## Step 3 — Branch and Merge

When an analytics professional wants to make updates, he or she checks a copy of all of the relevant code out of the revision control system. He or she then can make changes to a local, private copy of the code. These local changes are called a branch. Revision control systems boost team productivity by allowing many developers to work on branches concurrently. When changes to the branch are complete, tested and known to be working, the code can be checked back into revision control, thus merging back into the trunk or main code base.

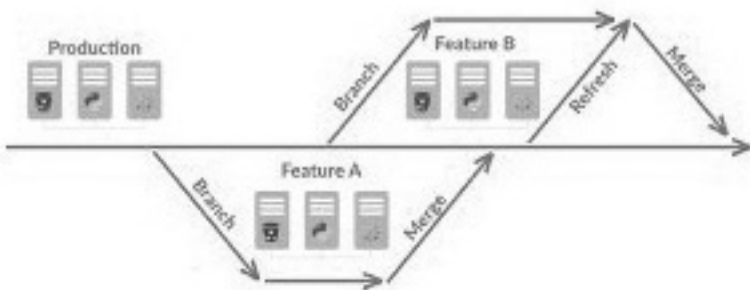


Figure 67: With “Branch and Merge” team members can work on features independently without impacting each other or value pipeline.

Branching and merging allow the data science team to run their own tests, make changes, take risks and experiment. If a set of changes proves to be unfruitful, the branch can be discarded, and the team member can start over.

## Step 4 — Use Multiple Environments

Infrastructure-as-a-service (IaaS) (or alternatively, platform-as-a-service or virtualization) has evolved to the point where new virtual machines, operating systems, stacks, applications and copies of data can be provisioned, with a click or command, under software control. DataOps calls for development and test environments that are separate from operations. The last thing that anyone would want is for a data scientist making changes to crash enterprise-critical analytics. When the team creating new analytics is given their own environments, they can iterate quickly without worrying about impacting operations. IaaS makes it easy to set-up development and test system environments that exactly match a target operations environment. This helps prevent finger-pointing between development, quality assurance and operations/IT.

It's worth reemphasizing that data scientists need a copy of the data. In the past, creating copies of databases was expensive. With storage on-demand from cloud services, a Terabyte data set can be quickly and inexpensively copied to reduce conflicts and dependencies. If the data is still too large to copy, it can be sampled.

## Step 5 — Reuse & Containerize

Data science team members typically have a difficult time leveraging each other's work. Code reuse is a vast topic, but the basic idea is to componentize functionalities in ways that can be shared. Complex functions, with lots of individual parts, can be containerized using a container technology like Docker and Kubernetes. Containers are ideal for highly customized functions that require a skill set that isn't widely shared among the team.

## Step 6 — Parameterize Your Processing

The data analytics pipeline should be designed with run-time flexibility. Which dataset should be used? Is a new data warehouse used for production or testing? Should data be filtered? Should specific workflow steps be included or not? These types of conditions are coded in different phases of the data analytics pipeline using parameters. In software development, a parameter is some information (e.g., a name, a number, an option) that is passed to a program that affects the way that it operates. With the right parameters in place, accommodating the day-to-day needs of the users and data science professionals becomes a routine matter.

## Step 7 — Orchestrate Two Journeys

Many data science professionals dread the prospect of deploying changes that break production systems or allowing poor quality data to reach users. Addressing this requires optimization of both the Value and Innovation Pipelines. In the Value Pipeline (production), data flows into production and creates value for the organization. In the Innovation Pipeline, ideas, in the form of new analytics and AI, undergo development and are added to the Value Pipeline. The two pipelines intersect as new analytics deploy into operations. The DataOps enterprise masters the orchestration of data to production and the deployment of new features both while maintaining impeccable quality. With tests (statistical process control) controlling and monitoring both the data and new development pipelines, the dev team can deploy without worrying about breaking the production systems. With Agile Development and DevOps, the velocity of new analytics is maximized.

Figure 68 below shows how the seven steps of DataOps tie directly into the steps for model development shown in Figure 63. For example, orchestration automates data preparation, feature extraction, model training and model deployment. Version control, branch and merge, environments, reuse & containers and parameterization all apply to these same phases. Tests apply to all of the phases of the model life cycle.

## The Seven Steps and Data Science

|                    | Tests | Version Control | Branch and Merge | Environments | Reuse / Containerize | Parameterize | Orchestrate |
|--------------------|-------|-----------------|------------------|--------------|----------------------|--------------|-------------|
| Business Need      |       |                 |                  |              |                      |              |             |
| Prep Data          | *     | *               | *                | *            | *                    | *            | *           |
| Feature Extraction | *     | *               | *                | *            | *                    | *            | *           |
| Build Model        | *     | *               | *                | *            | *                    | *            | *           |
| Evaluate Model     | *     |                 |                  |              |                      |              |             |
| Deploy Model       | *     | *               | *                | *            | *                    | *            | *           |
| Monitor Model      | *     |                 |                  |              |                      |              |             |

Copyright © 2019 by DataKitchen, Inc. All Rights Reserved.

Figure 68: How the 7 steps of DataOps related to model development

## CONCLUSION

While AI and data science tools improve the productivity of model development, the actual ML code is a small part of the overall system solution. Data science teams that don't apply modern software development principles to the data lifecycle can end up with poor quality and technical debt that causes unplanned work.

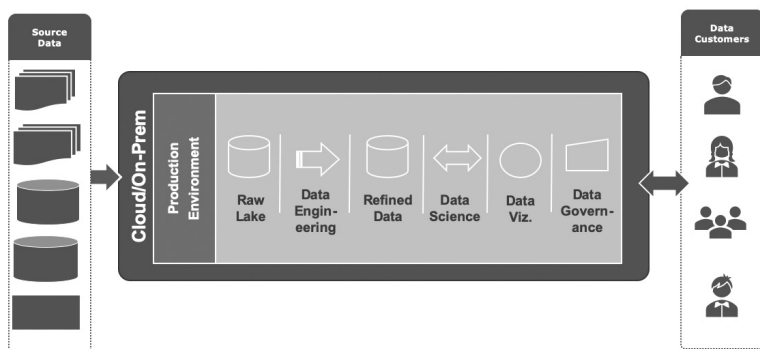
DataOps offers a new approach to creating and operationalizing AI that minimizes technical debt, reduces cycle time and improves code and data quality. It is a methodology that enables data science teams to thrive despite increasing levels of complexity required to deploy and maintain AI in the field. It decouples operations from new analytics creation and rejoins them under the automated framework of continuous delivery. The orchestration of the development, deployment, operations and monitoring toolchains dramatically simplifies the daily workflows of the data science team. Without the burden of technical debt and unplanned work, they can focus on their area of expertise; creating new models that help the enterprise realize its mission.

# The “Right to Repair” Data Architecture with DataOps

We’ve been attending data conferences for over 20 years. It has been common to see presenters display a data architecture diagram like the (simplified) one below (figure 69). A data architecture diagram shows how raw data turns into insights. As the Eckerson Group [writes](#), “a data architecture defines the processes to capture, transform, and deliver usable data to business users.”

In our canonical data architecture diagram, data sources flow in from the left and pass through transformations to generate reports and analytics for users or customers on the right. In the middle, live all of the tools of the trade: raw data, refined data, data lakes/warehouses/marts, data engineering, data science, models, visualization, governance and more. Tools and platforms can exist in the cloud or on premises. Most large enterprise data architectures have evolved to use a mix of both.

## Canonical Data Architecture



Copyright © 2019 by DataKitchen, Inc. All Rights Reserved.

Figure 69: A typical Production-only view of Data Architecture

When data professionals define data architectures, the focus is usually on production requirements: performance, latency, load, etc. Engineers and data professionals do a great job executing on these requirements. The problem is that the specifications don’t include *architecting for rapid change*.

They only think about production, not the process to make changes to production. **A Data-Ops Data Architecture makes the steps to change what is in production a “central idea.”** Thinking first about changes over time to your code, your servers, your tools, and monitoring for errors are first class citizens in the design.

Take this example. Mobile phone designs increasingly locate batteries in fixed locations underneath sensitive electronics. In many cases, batteries can no longer be easily accessed and



replaced by a consumer. The “Right to Repair” movement advocates for policies that enable customers to fix the things that they own instead of throwing them away.

When managers and architects fail to think about architecting the production data pipeline for rapid change and efficient development, it is a little like designing a mobile phone with a fixed battery. You can end up with processes characterized by unplanned work, manual deployment, errors, and bureaucracy. It can take months to deploy a minor [20 line SQL change](#).

Building a data architecture without planning for change is much worse than building a mobile phone with a fixed battery. While mobile phone batteries are swapped every few years, your analytics users are going to want changes every day or sometimes every hour. That may be impossible with your existing data architecture, but you can meet this requirement if you architect for it. If data architectures are designed with these goals in mind, they can be more flexible, responsive, and robust. Legacy data pipelines can be upgraded to achieve these aims by enhancing the architecture with modern tools and processes.

## A DATAOPS DATA ARCHITECTURE

Imagine if your data architects were given these requirements up front. In addition to the standard items, the user story or functional specification could include requirements like these:

1. Update and publish changes to analytics within an hour without disrupting operations
2. Discover data errors before they reach published analytics
3. Create and publish schema changes in a day

If you are a data architect yourself (or perhaps you play one), you may already have creative ideas about how you might address these types of requirements. You would have to maintain separate but identical development, test, and production environments. You would have to orchestrate and automate test, monitoring, and deployment of new analytics to production. When you architect for flexibility, quality, rapid deployment, and real-time monitoring of data (in addition to your production requirements), you are moving towards a DataOps data architecture as shown in figure 70.

The DataOps data architecture expands the traditional operations-oriented data architecture by including support for [Agile iterative development](#), [DevOps](#), and [statistical process control](#). We call these tools and processes collectively a [DataOps](#) Platform. The DataOps elements in

our new data architecture are shown in figure 70.

## DataOps Functional Architecture

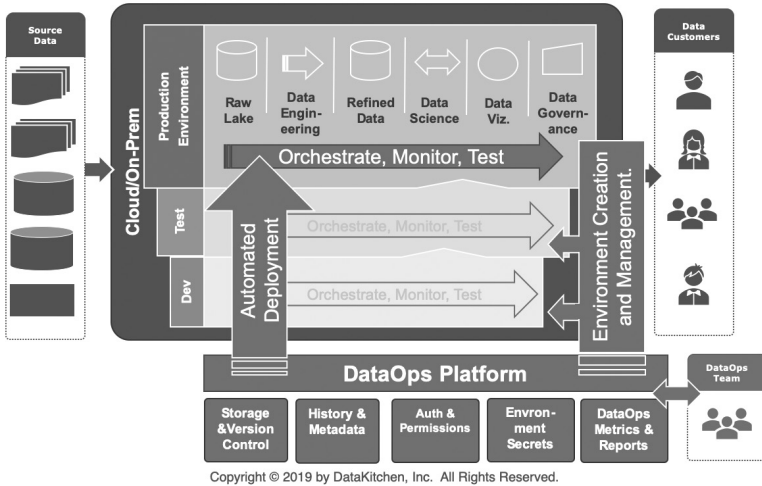


Figure 70: DataOps Functional Data Architecture

### BREAKDOWN OF THE DATAOPS ARCHITECTURE

A DataOps architecture contains support for [environment creation and management](#). This enables separate development, test, and production environments, which in turn support [orchestration](#), [monitoring](#), and [test automation](#). The software automates [impact review](#) and new-analytics deployment so that changes can be vetted and published [continuously](#). Agents in each environment operate on behalf of the DataOps Platform to manage code and configuration, execute tasks, and return test results, logs, and runtime information. This enables the architecture to work across heterogeneous tools and systems. The DataOps Platform also integrates several other functions which support the goal of rapid deployment and high quality with governance:

- **Storage /Revision Control** — [Version control](#) manages changes in artifacts; essential for governance and iterative development. (example: git, docker hub)
- **History and Metadata** — Manage system and activity logs (example, MongoDB)
- **Authorization and Permissions** — Control access to environments (example: Auth0)

- **Environment Secrets** — Role-based access to tools and resources within environments (example: Vault)
- **DataOps Metrics and Reports** — Internal analytics provide a big-picture assessment of the state of the analytics and data team. We call this the [CDO Dashboard](#). (example: Tableau)
- **Automated Deployment** — This involves moving the code/configuration from one environment (e.g., a test environment) to a production environment. (Examples: Jenkins, CircleCI)
- **Environment Creation and Management** — treat your infrastructure as code be able to create places for your team to do work with all the required hardware, software, and test data sets they need. (example: chef, puppet, etc.)
- **Orchestrate, Test, Monitor** — As your pipelines are running, orchestrate all the tools involved, test and monitor, and alert if something goes wrong. (examples, Airflow, Great Expectations, Grafana, etc.)

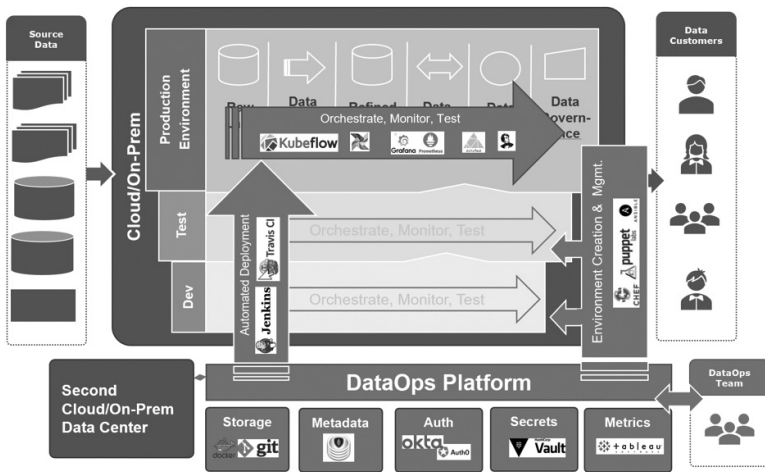


Figure 71: DataOps Data Architecture with Example Tools

# MULTI-LOCATION DATAOPS DATA ARCHITECTURE

Companies are increasingly moving their work from on-premises to the cloud. Enterprises are choosing to have multiple cloud providers, as well. As a result, your data analytics workloads can span multiple physical locations and multiple teams. Your customers [only see the result](#) of that coordination. How can you do DataOps across those locations and teams and not end up with a “Data Oooops”? Think of a “hub and spoke” model for your DataOps Data Architecture. As shown in figure 72, the DataOps Platform is the hub for your distributed sites engaging in development and operations. Testing is also coordinated between the sites.

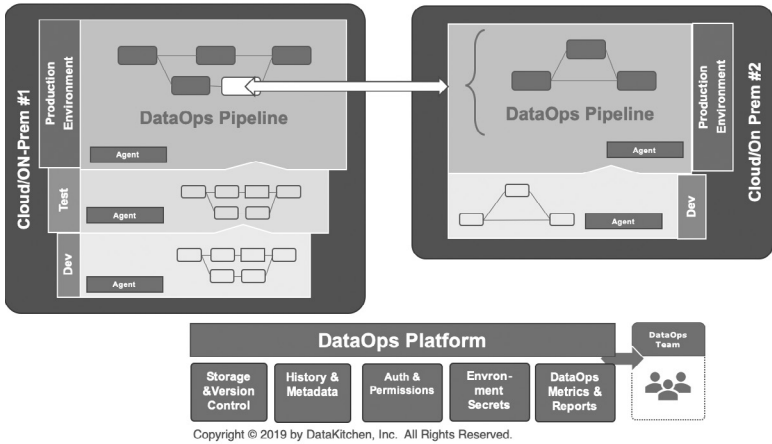


Figure 72: Multi-location DataOps Data Architecture

# BUILDING DATAOPS INTO AN EXISTING DATA ARCHITECTURE

Whether your current data architecture is on-prem or in the cloud or a mix of both; whether you have a standard environment or live in a multi-tool world, you can evolve your system to incorporate [DataOps functionalities](#). You can build a DataOps Platform yourself or leverage solutions from the vibrant and growing DataOps [ecosystem](#). DataOps can help you architect your data operations pipeline to support rapid development and deployment of new analytics, robust quality, and high levels of staff productivity.

You have the “Right to Repair” your data architecture — design for it!

# Enabling Design Thinking in Data Analytics with DataOps

Want to boost data-analytics innovation? Try “**Design Thinking**.”

[Design Thinking](#) is a solution-based design methodology which organizations use to address ill-defined or tricky problems that defy conventional approaches. It uniquely marries design with customer empathy to produce solutions that address latent customer needs. The Design Thinking methodology re-frames problems in human-centric ways, creates many ideas in brainstorming sessions, and then adopts a hands-on approach to prototyping and testing.

The Design Thinking process boils down to three steps:

- **Empathy** — Gain an understanding of the problem you are trying to solve by consulting experts, observing, and empathizing. The discovery process enables designers to think beyond their own assumptions, about the end user’s needs, and the problem space.
- **Ideation** — Generate lots of ideas — as many as possible
- **Experimentation** — Create prototype solutions, test, learn and repeat

Design Thinking has grown beyond its physical design roots to guide innovation in education, business and computer science. As you would expect, data professionals are now applying design thinking to [data science](#). Design thinking can serve as a major boost to corporate innovation. Unfortunately, most data organizations are not set-up for a rapid feedback loop of Ideation and Experimentation, so creativity never shifts into high gear.

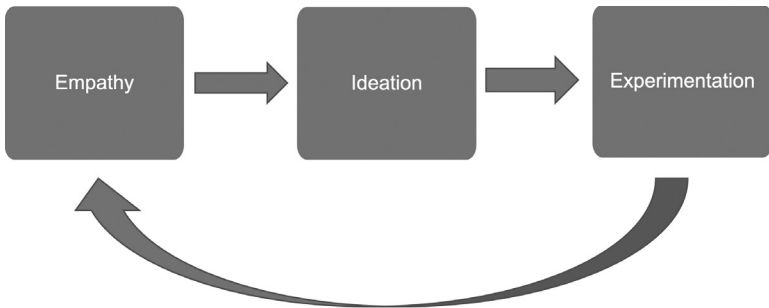


Figure 73: Design Thinking consists of three stages: Empathy, Ideation and Experimentation

Figure 73 shows the stages of Empathy, Ideation and Experimentation in series. As any experienced “Design Thinker” will tell you, the stages do not necessarily happen in sequence. Experimentation can lead to deeper Empathy, which fuels Ideation. The process could have

many feedback loops. One issue that frustrates Design Thinking in data analytics is that Experimentation can take much longer than Empathy and Ideation. It can take weeks or months for the data team to implement a relatively minor change in analytics. Nothing interrupts the creative juices of innovation from flowing like waiting and waiting and waiting some more. When users can't see immediate feedback on their ideas, they may lose interest.

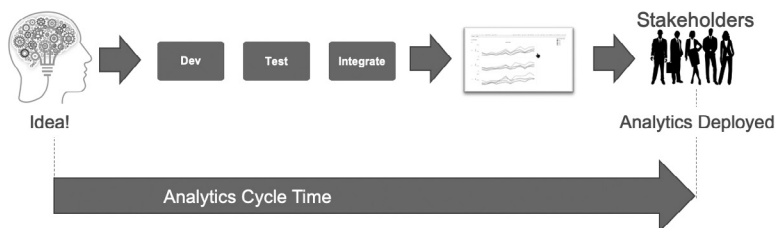


Figure 74: Cycle time is the period of time required to turn a new idea into deployed analytics. In many organizations, cycle time is unacceptably long.

## FACTORS THAT LENGTHEN CYCLE TIME

We'd like to say that data teams work hand-in-hand with their users like a well-oiled machine, fielding new idea proposals, implementing them rapidly and quickly iterating toward higher-quality models and analytics. Unfortunately, our experience is the opposite. Data teams are constantly interrupted by data and analytics errors. Data scientists spend 75% of their time massaging data and executing manual steps.

Productive Design Thinking depends on a quick turn-around between ideation and experimentation. The problem is that the Experimentation phase can be unacceptably slow. Lengthy analytics [cycle time](#) occurs for a variety of reasons:

- Poor [teamwork](#) within the data team
- Lack of collaboration [between groups](#) within the data organization
- Waiting for IT to disposition or configure system resources
- Waiting for access to data
- Moving [slowly and cautiously](#) to avoid poor quality
- Requiring approvals, such as from an [Impact Review Board](#)
- Inflexible [data architectures](#)
- [Process bottlenecks](#)
- [Technical debt](#) from previous deployments
- Poor quality creating unplanned work

As daunting as some of these challenges are, some data organizations have proven that it is possible to achieve rapid cycle time. They do this using a methodology called DataOps.

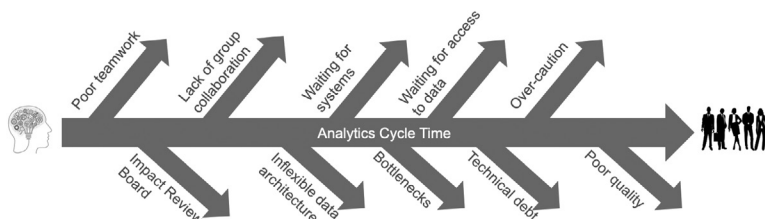


Figure 75: Factors that derail the dev team and lengthen analytics cycle time

## HOW DATAOPS MINIMIZES CYCLE TIME

Data analytics can leverage the same processes and methodologies that have boosted the productivity of software engineering 100x in the last decades. We call these techniques (collectively) [DataOps](#). It includes three important methodologies: [Agile Software Development](#), [DevOps](#) and [statistical process controls](#) (SPC). DataOps requires data teams to rethink how they manage projects, create and deploy new features, publish analytics and control quality:

- **Managing projects** — Features are delivered iteratively using Agile Development. Agile is perfect for the “Ideate and Experiment” cycle characteristic of Design Thinking
- **Creation and deployment of analytics** — Cycle time is minimized when [development and production environments are aligned](#) and when tools support [seamless teamwork](#) among members of the dev team and [between groups](#) within the data organization. DataOps also orchestrates the quality assurance and continuous deployment of code.
- **Operations** — Automated orchestration cleans raw data, executes ETL and publishes analytics as part of the operational workflow.
- **Quality** — Tests validate raw data as well as inputs, outputs and business logic in every stage of operations and new-analytics deployment. Dashboards and real-time alerts provide transparency related to issues.

Quality is an important aspect of cycle time. A team can't reduce cycle time if they are being constantly interrupted by quality problems and high-severity alerts. DataOps applies automated testing to the data operations pipeline as well as the release pipeline for new analytics.

DataOps comprehends that enterprises live in a multi-language, multi-tool, heterogeneous environment with complex workflows. To implement DataOps, extend your existing environment to align with DataOps principles. As we have written extensively, you can implement DataOps by yourself in [seven steps](#), or you can adopt a [DataOps Platform](#) from a third party vendor.

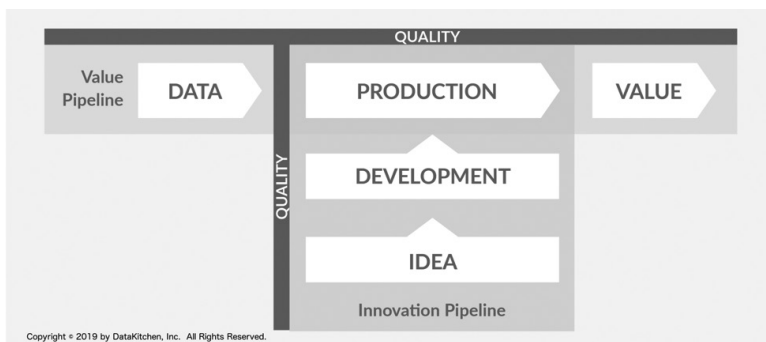


Figure 76: DataOps manages the creation and deployment of analytics ("Innovation Pipeline") and orchestrates data operations ("Value Pipeline").

## DATAOPS ENABLES DESIGN THINKING

With automated orchestration, end-to-end testing and seamless transitions from dev to production, DataOps minimizes analytics cycle time, enabling the close coupling of Ideation and Experimentation required for Design Thinking. Analytics teams publish changes to analytics quickly and confidently. When users propose ideas, new analytics can be created rapidly, providing immediate feedback. When users see immediate results, it triggers their brain to further Ideate. When implemented at high velocity, Design Thinking can make short work out of an organization's most formidable challenges.

# DataOps Puts Agility into Agile Data Warehousing

Data [analytics](#) professionals get used to being in no-win situations. Internal customers make a simple request; for example, add a new file to the database. Users expect requests like these to take days, yet, in many large organizations, they require months to complete. At DataKitchen, we repeatedly hear from companies that they need to improve their cycle time for new analytics. One approach, *Agile Data Warehousing*, applies [Agile principles](#) to [data warehouse](#) projects in an attempt to speed innovation. However, many companies quickly discover that simply implementing [Scrum](#) is not sufficient to attain results.

Imagine that you oversee a fifty-person team managing numerous large integrated databases (DB) for a big insurance or financial services company. You have 300 terabytes (TB) of data which you manage using a proprietary database. Between software, licensing, maintenance, support and associated hardware, you pay \$10M per year in annual fees. Even putting another single CPU into production could cost hundreds of thousands of dollars.

Someday these large databases will move to the cloud at a fraction of the cost. New databases will be turned on and off like light bulbs with the enterprise only paying for the resources they consume. That's a long-term goal. In the short term, the team has to produce results using the existing platform.

You can't afford separate instantiations of the entire data set for development, quality assurance (QA), performance testing and production so non-production machines are given subsets of the data. The necessity of provisioning physically separate hardware instantiations is one barrier to greater Agility.



The machine environments are different and have to be managed and maintained separately. New analytics are tested on each machine in turn — first in dev, then QA and finally production. You may not catch every problem in dev and QA since they aren't using the same data and environment as production.

Running regression tests manually is time-consuming so it can't be done often. This creates risk whenever new code is deployed. Also, when changes are made on one machine they have to be manually installed on the others. The steps in this procedure are detailed in a 30-page text document, which is updated by a committee through a cumbersome series of reviews and meetings. It is a very siloed and fractured process, not to mention inefficient; during upgrades, the DB is offline, so new work is temporarily on hold.

In our hypothetical company, the organization of the workforce is also a factor in slowing the team's velocity. Everyone is assigned a fixed role. Adding a table to a database involves several discrete functions: a [Data Quality](#) person who analyzes the problem, a Schema/Architect who designs the [schema](#), an ETL engineer who writes the ETL, a Test Engineer that writes tests and a Release Engineer who handles deployment. Each of these functions is performed sequentially and requires considerable documentation and committee review before any action is taken. Hand-off meetings mark the transition from one stage to the next.

The team wants to move faster but is prevented from doing so due to heavyweight processes, serialization of tasks, overhead, difficulty in coordination and lack of automation. They need a way to increase collaboration and streamline the many inefficiencies of their current process without having to abandon their existing tools.

## HOW DATAOPS HELPS

[DataOps](#) is a new approach to data analytics that automates the [orchestration](#) of data to production and the deployment of new features, both while maintaining impeccable quality. DataOps does not mandate the use of any particular tool or technology, but support in the following areas can be critical to Agile Data Warehousing in large teams, such as the one described:

**Shared Workspace** – DataOps creates a shared workspace so team members have visibility into each other's work. This enables the team to work more collaboratively and seamlessly outside the formal structure of the hand-off meeting. DataOps also streamlines documentation and reduces the need for formal meetings as a communication forum.

**Orchestration** – DataOps deploys code updates to each machine instantiation and automates the execution of tests along each stage of the data analytics pipeline. This includes data and logic tests that validate both the production and feature deployment pipelines. Tests are parameterized so they can run in the subset database of each particular machine environment equally well. As the test suite improves, it grows to reflect the full breadth of the production environment. Automated tests are run repeatedly so you can be confident that new features have not broken old ones.

These tools and process changes together break down the organizational and technology barriers that prevent the team from implementing Agile methods in data analytics. DataOps unburdens the team from non-value-add tasks and empowers them to self-organize around new creative initiatives. When the team is free to innovate, the continuous improvement culture built into DataOps will begin working to reduce the cycle time of new analytics from months to days (and less). This ultimately puts the Agility back into Agile Data Warehousing by delivering high-quality analytics to users in a timely fashion.

# Speed Up Innovation with DataOps

## LEVERAGING DATA LAKES, DATA WAREHOUSES AND SCHEMAS FOR FASTER ANALYTICS

Analytics professionals often strain to make one change to their analytic pipeline per month. [DataOps](#) increases their productivity by an order of magnitude. DataOps accelerates innovation by automating and orchestrating the data analytics pipeline and speeding ideas to production. It does this by applying Agile Development, DevOps and statistical process controls to data analytics. This enables the [DataOps Engineer](#) to quickly respond to requests for new analytics while guaranteeing a high level of quality. In order to understand this, it is helpful to know a little about the role of data lakes, schemas and data warehouses in DataOps.



## DATAOPS REQUIRES EASY ACCESS TO DATA

When data is moved from disparate silos into a common repository, it is much easier for a data analytics team to work with it. The common store is called a [data lake](#). To optimize DataOps, it is often best to move data into a data lake using on-demand simple storage.

People often speak about data lakes as a repository for raw data. It can also be helpful to move processed data into the data lake. There are several important advantages to using data lakes. First and foremost, the data analytics team controls access to it. Nothing can frustrate progress more than having to wait for access to an operational system (ERP, CRM, MRP, ...). Additionally, a data lake brings data together in one place. This makes it much easier to process. Imagine buying items at garage sales all over town and placing them in your backyard. When you need the items, it is much easier to retrieve them from the backyard rather than visiting each of the garage sale sites. A data lake serves as a common store for all of the organization's critical data. Easy, unrestricted access to data eliminates restrictions on productivity that slow down the development of new analytics.

Note that if you put public company financial data in a data lake, everyone who has access to the data lake is an “insider.” If you have confidential data, HIPAA data (Health Insurance Portability and Accountability Act of 1996) or Personally identifiable information (PII) — these must be managed in line with government regulations, which vary by country.

The structure of a data lake is designed to support efficient data access. This relates to how data is organized and how software accesses it. A database schema establishes the relationship between the entities of data.

## UNDERSTANDING SCHEMAS

A database schema is a collection of tables. It dictates how the database is structured and organized and how the various data relate to each other. Below is a schema that might be used in a pharmaceutical-sales analytics use case. There are tables for products, payers, period, prescribers and patients with an integer ID number for each row in each table. Each sale recorded has been entered in the fact table with the corresponding IDs that identify the product, payer, period, and prescriber respectively. Conceptually, the IDs are pointers into the other tables.

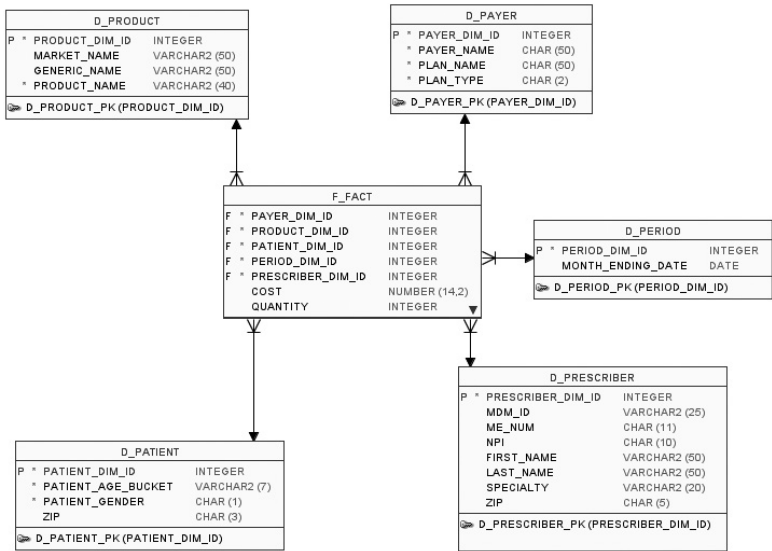


Figure 77: The Schema of a Pharmaceutical-Sales Analytics System

The schema establishes the basic relationships between the data tables. A schema for an operational system is optimized for inserts and updates. The schema for an analytics system, like the [star schema](#) shown here, is optimized for reads, aggregations, and is easily understood by people.

Suppose that you want to do analysis of patients based on their MSA (metropolitan service area). An MSA is a metropolitan region usually clustered near a large city. For example, Cambridge, Massachusetts is in the Greater Boston MSA. The prescriber table has a zip-code field. You could create a zip-code-to-MSA lookup table or just add MSA as an attribute to the patient table. Both of these are schema changes. In one case you add a table and in the other case you add a column.

## TRANSFORMS CREATE DATA WAREHOUSES

The data lake provides easier access, but lacks the optimizations needed for visualizations or modeling. For example, data often enters the data lake in the format of the source system and not using an optimized schema that facilitates analysis. Data warehouses better address analytic-specific requirements. For example, the data warehouse could have a schema that supports specific visualization, modeling or other features.

You might hear the term *data mart* in relation to data analytics. Data marts are a streamlined form of data warehouses. The two are conceptually very similar.

Data transforms (scripts, source code, algorithms, ...) create data warehouses from data lakes. In [DataOps](#) this process is optimized by keeping transform code in source control and by automating the deployment of data warehouses. An automated deployment process is significantly faster, more robust and more productive than a manual deployment process.

## THE DATAOPS PIPELINE

The automation of the pipeline that transforms the schemas of data lakes, creating data warehouses and data marts, is a key reason that DataOps is able to improve the speed and quality of the data analytic pipeline. Without using a data lake, data is highly dispersed, and difficult to access. Schemas of operational systems are difficult to navigate and most likely not optimized for analytics.

DataOps moves the enterprise beyond slow, inflexible, disorganized and error-prone manual processes. The DataOps pipeline leverages data lakes and transforms them into well-crafted data warehouses using [continuous deployment](#) techniques. This speeds the creation and deployment of new analytics by an order of magnitude. Additionally, the DataOps pipeline is constantly monitored using statistical process control so the analytics team can be confident of the quality of data flowing through the pipeline. Work Without Fear or Heroism. With these tools and process improvements, DataOps compresses the cycle time of innovation while ensuring the robustness of the analytic pipeline. Faster and higher quality analytics ultimately lead to better insights that enable an enterprise to thrive in a dynamic environment.

# How to Inspire Code Reuse in Data Analytics

In [DataOps](#), the data analytics team moves at lightning speed using highly optimized tools and processes. One of the most important productivity tools is the ability to reuse and [containerize](#) code.

When we talk about reusing code, we mean reusing data analytics components. All of the files that comprise the data analytics pipeline — scripts, source code, algorithms, html, configuration files, parameter files — we think of these as code. Like other software development, code reuse can significantly boost coding velocity.



Code reuse saves time and resources by leveraging existing tools, libraries or other code in the extension or development of new code. If a software component has taken several months to develop, it effectively saves the organization several months of development time when another project reuses that component. This practice can be used to decrease projects budgets. In other cases, code reuse makes it possible to complete projects that would have been impossible if the team were forced to start from scratch.

Containers make code reuse much simpler. A container packages everything needed to run a piece of software — code, runtimes, tools, libraries, configuration files — into a stand-alone executable. Containers are somewhat like virtual machines but use fewer resources because they do not include full operating systems. A given hardware server can run many more containers than virtual machines.

A container eliminates the problem in which code runs on one machine, but not on another, because of slight differences in the set-up and configuration of the two servers or software environments. A container enables code to run the same way on every machine by automating the task of setting up and configuring a machine environment. This is one DataOps techniques that facilitates moving code from development to production — the run-time environment is the same for both. One popular open-source container technology is Docker.

Each step in the data-analytics pipeline is the output of the prior stage and the input to the next stage. It is cumbersome to work with an entire data-analytics pipeline as one monolith, so it is common to break it down into smaller components. On a practical level, smaller components are much easier to reuse by other team members.

Some steps in the data-analytics pipeline are messy and complicated. For example, one operation might call a custom tool, run a python script, use FTP and other specialized logic. This operation might be both hard to set up, because it requires a specific set of tools, and difficult to create, because it requires a specific skill set. This scenario is another common use case for creating a container. Once the code is placed in a container, it is much easier to use by other programmers who aren't familiar with the custom tools inside the container but know how to use the container's external interfaces. All of the complexity is embedded inside the container. It is also easier to deploy that code to different environments. Containers make code reuse much more turnkey and allow developers much greater flexibility in sharing their work with each other.

# What Data Scientists Really Need

Kurt Cagle's perceptive analysis of data science titled "[Why You Don't Need Data Scientists](#)" explains the many reasons that data science falls short of the high expectations usually placed on it:

- Fancy dashboards are pretty but are only as valuable as the data behind them. Data quality often....stinks.
- Data sets are quirky and difficult to work with.
- Users/stakeholders know their business domain but little about what data can do for them.
- A multimillion-dollar initiative to rebuild the data pipeline from the ground up is generally *off the table*.
- The people who own the databases won't give [data scientists](#) access.
- Everyone agrees that integrating data from disparate databases is really, really hard, but in reality, it's much harder than people think.

These are all excellent points and often the conversation ends here — in exasperation. We can tell you that we have been there and have the PTSD to prove it. Fortunately, a few years ago, we found a way out of what may seem at times like a no-win situation. We believe that the secret to successful data science is a little about *tools* and a lot about *people and processes*.



## DON'T BOIL THE OCEAN

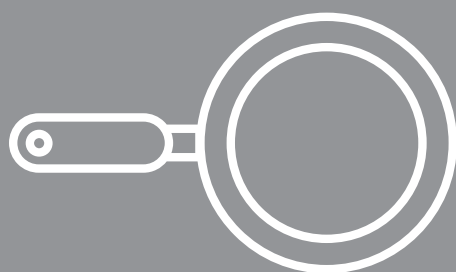
Use Agile methods to create new analytics. Leverage infrastructure that enables teams to work together in an Agile way. Start small and simple. Create something quickly that adds value. Get feedback from your stakeholders. Repeat iteratively.

## TESTS ARE BEST

Implement automated process controls that monitor data at every stage of your data analytics pipeline. Think of your data analytics as a lean manufacturing pipeline where the quality of data cannot drift outside statistical and logical bounds. Let your tools work 24x7 so data scientists can stay focused on creating analytics that add value.

## AUTOMATE AND ORCHESTRATE

Data Scientists spend 75% of their time doing [data engineering](#). It's about time that data professionals took a page from DevOps. Automate workflow and the deployment of new analytics. Orchestrate the end-to-end data pipeline so we stop sucking the life out of data scientists. A single data engineer should be able to support ten data analysts and scientists, who in turn should be supporting 100 business professionals. An automated pipeline can get you there.



# DataOps No-Knead Bread

by Gil Benghiat

## INGREDIENTS AND TOOLS

- 3 cups bread flour
- 1/4 teaspoons instant yeast
- ½ teaspoon salt
- 1 ½ cups very warm, almost hot water
- Oil for work surface (canola)

## INSTRUCTIONS

1. Combine flour, yeast and salt in a large bowl and stir with your DataKitchen spoon. Add water and stir until blended; dough will be shaggy. You may need an extra ¼ cup of water to get all the flour to blend in. Cover bowl with plastic wrap. Let dough rest at least 4 hours (12-18 hours is good too) at warm room temperature, about 70 degrees.
2. Lightly oil a work surface and place dough on it; fold it over on itself once or twice. Cover loosely with plastic wrap and let rest 30 minutes more. This is a good time to turn the oven on to 425°F.
3. Put a 6-to-8-quart heavy covered pot (cast iron, enamel, Pyrex or ceramic) in the oven as it heats. When dough is ready, carefully remove pot from oven. Slide your hand under dough and put it into pot, seam side up. Shake pan once or twice if dough is unevenly distributed; it will straighten out as it bakes.
4. Cover with lid and bake 30 minutes, then remove lid and bake another 15 to 30 minutes, until loaf is beautifully browned. Cool on a rack.

## NOTES

In a convection oven, cook 23 minutes with the lid on, and then 5 minutes with the lid off.

You don't need to pre-heat the pot. You can put the dough on a cookie sheet. The only difference is the crust will not be as crunchy or as beautifully browned. You can experiment with a round shape or Italian or French loaf shapes. The longer shapes will take less time to cook.

You can also cook at a lower temperature (e.g. 350°F). In all cases, take the bread out when the internal temperature reaches 190°F - 200°F. Use a meat thermometer to check.

*Derived from New York Times Speedy No-Knead Bread*

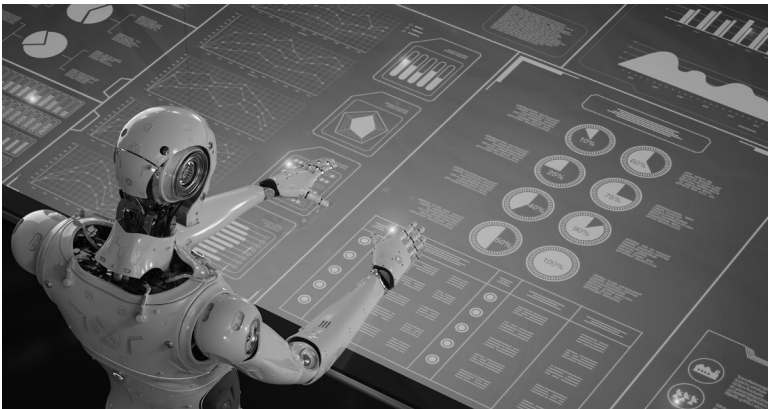
# DataOps for Data Quality

## Disband Your Impact Review Board: Automate Analytics Testing

Some companies take six months to write 20 lines of SQL and move it into production.

The last thing that an analytics professional wants to do is introduce a change that breaks the system. Nobody wants to be the object of scorn, the butt of jokes, or a cautionary tale. If that 20-line SQL change is misapplied, it can be a “career-limiting move” for an analytics professional.

Analytics systems grow so large and complex that no single person in the company understands them from end to end. A large company often institutes slow, bureaucratic procedures for introducing new analytics in order to reduce fear and uncertainty. They create a waterfall process with specific milestones. There is a lot of documentation, checks and balances, and meetings — lots of meetings.



## IMPACT ANALYSIS

One of the bottlenecks in an analytics release process is called “impact analysis.” Impact analysis gathers experts on all of the various subsystems (data feeds, databases, transforms, data lakes/warehouses, tools, reports, ...) so they can review the proverbial 20 lines of SQL and try to anticipate if/how it will adversely impact data operations.

Imagine you are building technical systems that integrate data and do models and visualizations. How does a change in one area affect other areas? In a traditional established company, that information is locked in various people’s heads. The company may think it has no choice but to gather these experts together in one room to discuss and analyze proposed changes. This is called an “impact analysis meeting.” The process includes the company’s most senior technical contributors; the backbone of data operations. Naturally, these individuals are extremely busy and subject to high-priority interruptions. Sometimes it takes weeks to gather them in one room. It can take additional weeks or months for them to approve a change.

The impact analysis team is a critical bottleneck that slows down updates to analytics. A [DataOps](#) approach to improving analytics cycle time adopts process optimization techniques from the manufacturing field. In a factory environment, a small number of bottlenecks often limit throughput. This is called the [Theory of Constraints](#). Optimize the throughput of bottlenecks and your end-to-end cycle time improves (check out “*The Goal*” by Eliyahu M. Goldratt).

## GET OUT OF YOUR HEAD

The Impact Analysis Meeting is a bottleneck because it relies upon your top technical experts — one of the most oversubscribed resources in the company. What if you could extract all the knowledge and experience trapped in the brains of your company’s experts and code it into a series of tests that would perform the impact analysis for you? This would give you a quick way to test out changes to analytics without requiring bureaucratic procedures and meetings. If the tests pass, you could deploy with confidence. No more waiting on the impact review team. With a comprehensive test suite, you reduce reliance on the impact analysis bottleneck and move a lot faster.

## AUTOMATING IMPACT ANALYSIS

Manual testing moves the bottleneck from impact review to the testing team. Manual testing is performed step-by-step, by a person. This tends to be expensive as it requires someone to create an environment and run tests one at a time. It can also be prone to human error.

DataOps automates testing. Environments are spun up under machine control and test scripts, written in advance, are executed in batch. Automated testing is much more cost-effective and reliable than manual testing, but the effectiveness of automated testing depends on the quality and breadth of the tests. In a DataOps enterprise, members of the analytics team spend 20% of their time writing tests. Whenever a problem is encountered, a new test is added. New tests accompany every analytics update. The breadth and depth of the test suite continuously grow.

One advantage of automated testing is that it's easier to run so it's executed repeatedly and regularly. Manual testing is often too expensive and slow to run on a regular basis. To ensure high quality, you have to be able to consistently and regularly test your data and code.

These concepts are new to many data teams, but they are well established in the software industry. As figure 78 shows, the cycle time of software development releases has been (and continues to be) reduced by orders of magnitude through automation and process improvements. The automation of impact analysis can have a similar positive effect on your organization's analytics cycle time.

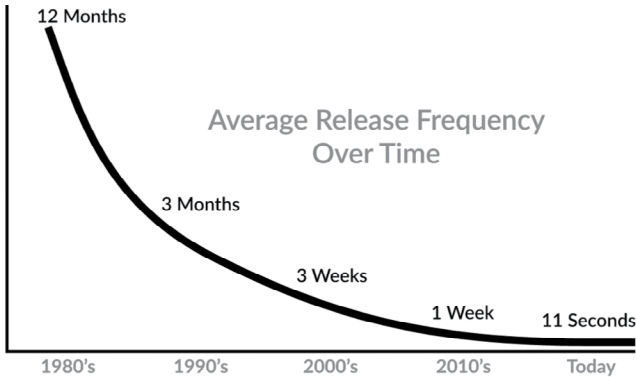


Figure 78: Software developers have reduced the cycle time for new releases by orders of magnitude using automation and process improvements

## ANALYTICS IS CODE

At this point some of you are thinking *this has nothing to do with me. I am a data analyst/scientist, not a coder. I am a tool expert. What I do is just a sophisticated form of configuration.* This is a common point of view in data analytics. However, it leads to a mindset that slows down analytics cycle time.

Tools vendors have a business interest in perpetuating the myth that if you stay within the well-defined boundaries of their tool, you are protected from the complexity of software development. This is ill-considered.

Don't get us wrong. We *love our tools*, but don't buy into this falsehood.

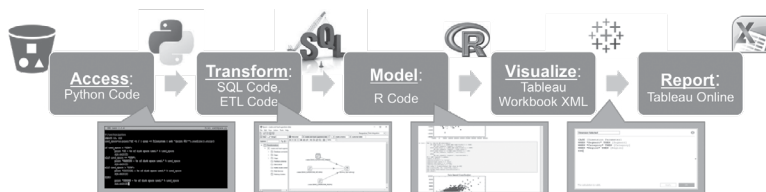


Figure 79: From data access to visualization to reports, there is code running at every stage of the data operations pipeline

The \$100B analytics market is divided into two segments: tools that create code and tools that run code. The point is — data analytics is code. The data professional creates code and must own, embrace and manage the complexity that comes along with it.

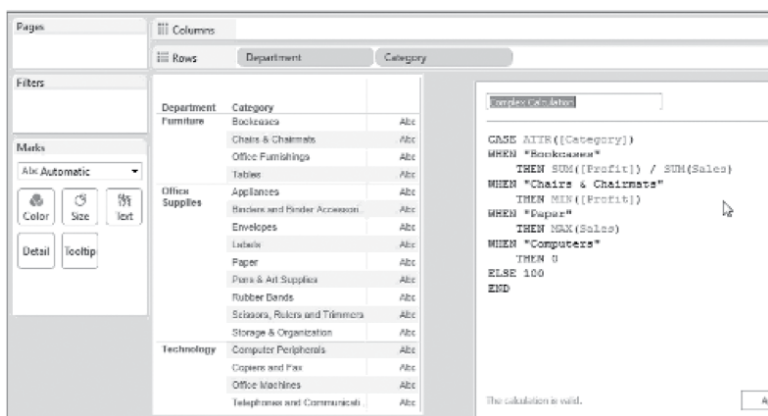


Figure 80: Tableau files are stored as XML, and can contain conditional branches, loops and embedded code.

Figure 79 shows a data operations pipeline with code at every stage of the pipeline. Python, SQL, R — these are all code. The tools of the trade (Informatica, Tableau, Excel, ...) these too are code. If you open an Informatica or Tableau file, it's XML. It contains conditional branches (if-then-else constructs), loops and you can embed Python or R in it.

Remember our 20-line SQL change that took six months to implement? The problem is that analytics systems become so complex that they can easily break if someone makes one misbegotten change. The average data-analytics pipeline encompasses many tools (code generators) and runs lots of code. Between all of the code and people involved, data operations becomes a *combinatorially* complex hairball of systems that could come crashing down with one little mistake.

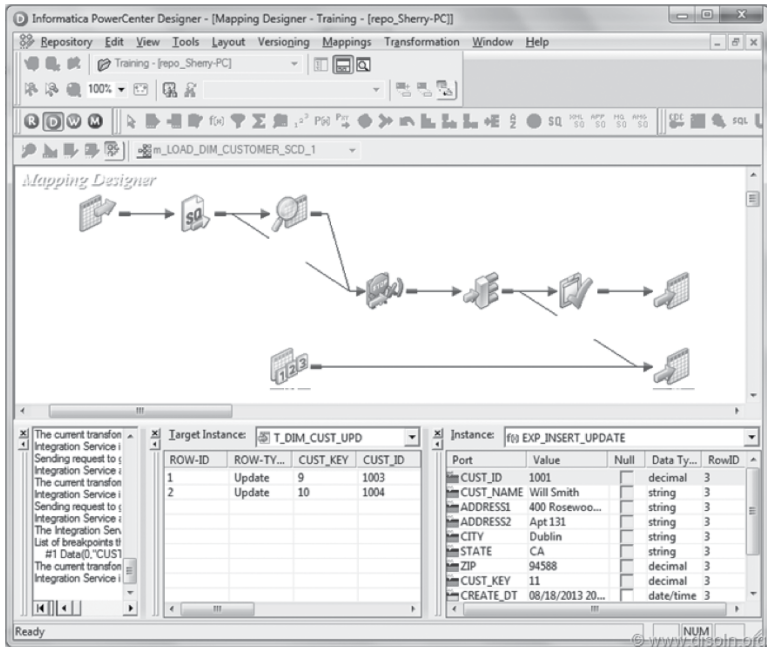


Figure 81: Informatica presents a UI that creates ETL in an XML format that is then converted to Java and executed on the machine.

For example, imagine that you have analytics that sorts customers into five bins based on some conditional criterion. Deep inside your tool's XML file is an if-then-else construct that is responsible for sorting the customers correctly. You have numerous reports based off of a template that contains this logic. They provide information to your business stakeholders: top customers, middle customers, gainers, decliners, whales, profitable customers,...

There's a team of IT engineers, database developers, data engineers, analysts and [data scientists](#) that manage the end to end system that supports these analytics. One of these individuals makes a change. They convert the sales volume field from an integer into a decimal. Perhaps they convert a field that was US dollars into a different currency. Maybe they rename a column. Everything in the analytics pipeline is so interdependent; the change breaks all of the reports that contain the if-then-else logic upon which the original five categories are built. All of a sudden, your five customer categories become one category, or the wrong customers are sorted into the wrong bins. None of the dependent analytics are correct, reports are showing incorrect data, and the VP of Sales is calling you hourly.

At an abstract level, every analytic insight produced, every deliverable, is an interconnected chain of code modules delivering value. The data analytics pipeline is best represented by a [directed acyclic graph](#) (DAG). For example, see figure 82.

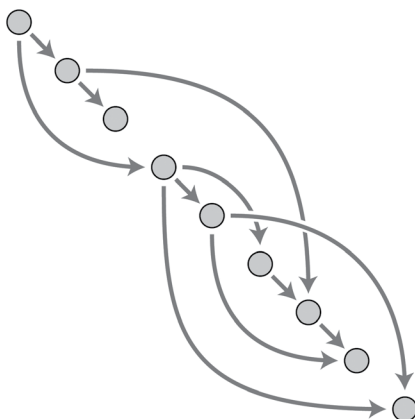


Figure 82: The Directed Acyclic Graph (DAG) models the steps in the data analytics pipeline

Whether you use an analytics tool like Informatica or Tableau, an [Integrated Development Environment](#) (IDE) like Microsoft Visual Studio (figure 83) or even a text editor like Notepad, you are creating code. The code that you create interacts with all of the other code that populates the DAG that represents your data pipeline.

To automate impact analysis, think of the end-to-end data pipeline holistically. Your test suite should verify software entities on a stand-alone basis as well as how they interact.

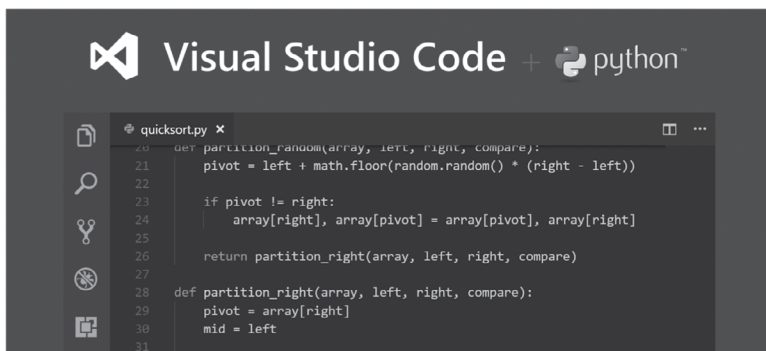


Figure 83: Developers write SQL, Python and other code using an integrated development environment or sometimes a simple editor like Notepad.

## TYPES OF TESTS

The software industry has decades of experience ensuring that code behaves as expected. Each type of test has a specific goal. If you spend any time discussing testing with your peers, these terms are sure to come up:

- **Unit Tests** – testing aimed at each software component as a stand-alone entity
- **Integration Tests** – focus on the interaction between components to ensure that they are interoperating correctly
- **Functional Tests** – verification against functional specification or user stories.
- **Regression Tests** – rerun every time a change is made to prove that an application is still functioning
- **Performance Tests** – verify a system's responsiveness, stability and availability under a given workload
- **Smoke Tests** – quick, preliminary validation that the major system functions are operational

## TESTS TARGET DATA OR CODE OR BOTH

It's also helpful to frame the purpose and context of a test. Tests can target data or code and run as part of the data operations pipeline. Location balance, historical balance and [statistical process controls](#) (time balance) tests are directed at the data flowing through an operations pipeline. The code that runs the data processing steps in the pipeline is fixed. The code is tightly controlled and only changed via a release process. Data that moves through operations, on the other hand, is variable. New data flows through the pipeline continuously. As figure 84 shows, the data operations pipeline delivers value to users. [DataOps](#) terms this the [Value Pipeline](#).

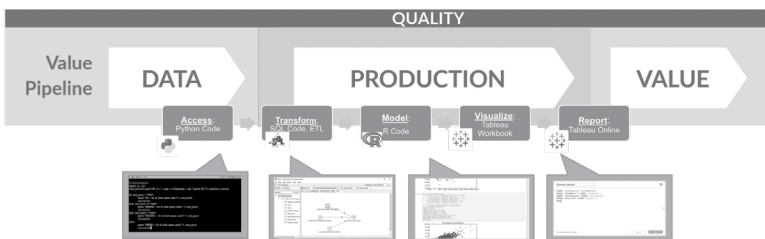


Figure 84: Data Operations: The Value Pipeline

The development of new analytics follows a different path, which is shown in figure 85 as the [Innovation Pipeline](#). The Innovation Pipeline delivers new insights to the data operations pipeline, regulated by the release process. To safely develop new code, the analyst needs an isolated [development environment](#). When creating new analytics, the developer creates an environment analogous to the overall system. If the database is terabytes in size, the data

professional might copy it for test purposes. If the data is petabytes in size, it may make sense to sample it; for example, take 10% of the overall data. If there are concerns about

|               | Data Fixed          | Data Variable  |
|---------------|---------------------|----------------|
| Code Fixed    |                     | Value Pipeline |
| Code Variable | Innovation Pipeline |                |

Table 5: In the Value Pipeline code is fixed and data is variable. In the Innovation Pipeline, data is fixed, and code is variable.

privacy or other regulations, then sensitive information is removed. Once the environment is set up, the data typically remains stable.

In the [Innovation Pipeline](#) code is variable, but data is fixed. Tests target the code, not the data. The unit, integration, functional, performance and regression tests that were mentioned above are aimed at vetting new code. All tests are run before promoting ([merging](#)) new code to production. Code changes should be managed using a [version control](#) system, for example GIT. A good test suite serves as an automated form of impact analysis that can be run on any and every code change before deployment.

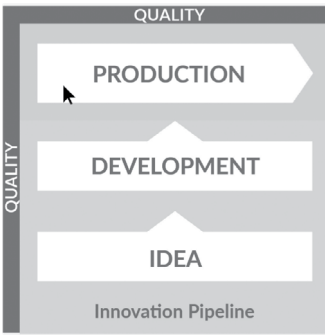
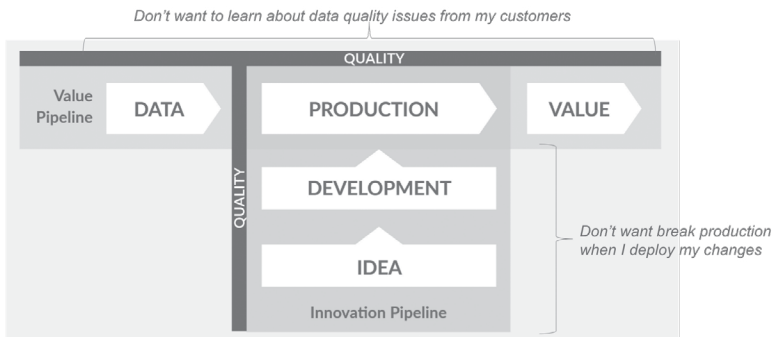


Figure 85: New analytics are developed in the Innovation Pipeline

Some tests are aimed at both data and code. For example, a test that makes sure that a database has the right number of rows helps your data and code work together. Ultimately both data tests and code tests need to come together in an integrated pipeline as shown in figure 86. DataOps enables code and data tests to work together so all around quality remains high.



**Figure 86: Ultimately the Value and Innovation Pipelines work together to maintain data and code quality**

A unified, automated test suite that tests/monitors both production data and analytic code is the linchpin that makes DataOps work. Robust and thorough testing removes or minimizes the need to perform manual impact analysis, which avoids a bottleneck that slows innovation. Removing constraints helps speed innovation and improve quality by minimizing analytics cycle time. With a highly optimized test process you'll be able to expedite new analytics into production with a high level of confidence.

*20 new lines of SQL? You'll have it right away.*

# Build Trust Through Test Automation and Monitoring

*“Trust takes years to build, seconds to break, and forever to repair.”*

We recently talked to a [data team](#) in a financial services company that lost the trust of their users. They lacked the resources to implement quality controls so bad data sometimes leaked into user analytics. After several high-profile episodes, department heads hired their own people to create reports. For a data-analytics team, this is the nightmare scenario, and it could have been avoided.

Organizations trust their data when they believe it is accurate. A data team can struggle to produce high-quality analytics when resources are limited, business logic keeps changing and data sources have *less-than-perfect* quality themselves. Accurate data analytics are the product of quality controls and sound processes.

The data team can't spend 100% of its time checking data, but if data analysts or scientists spend 10-20% of their time on quality, they can produce an automated testing and monitoring system that does the work for them. Automated testing can work 24x7 to ensure that bad data never reaches users, and when a mishap does occur, it helps to be able to assure users that new tests can be written to make certain that an error never happens again. Automated testing and monitoring greatly multiplies the effort that a data team invests in quality.



## DATA FLOW AS A PIPELINE

Think of data analytics as a manufacturing pipeline. There are inputs (data sources), processes (transformations) and outputs (analytics). A typical manufacturing process includes tests at every step in the pipeline that attempt to identify problems as early as possible. As every manufacturer knows, it is much more efficient and less expensive to catch a problem in incoming inspection as opposed to finished goods.

Figure 87 depicts the data-analytics pipeline. In this diagram, databases are accessed and then data is transformed in preparation for being input into models. Models output visualizations and reports that provide critical information to users.

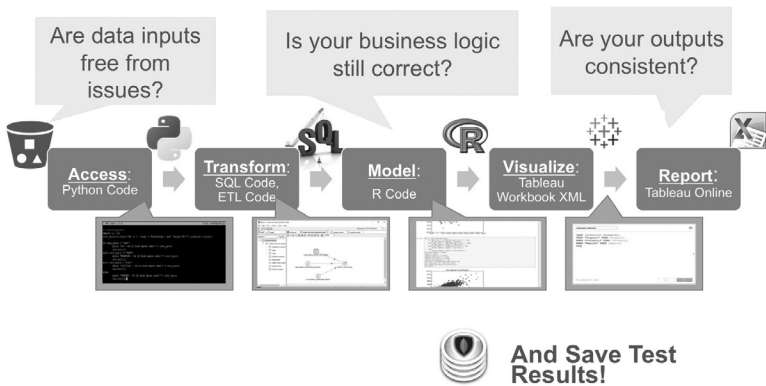


Figure 87: Testing each stage of the data-analytic pipeline

Along the way, tests ask important questions. Are data inputs free from issues? Is business logic correct? Are outputs consistent? As in lean manufacturing, tests are performed at every step in the pipeline. For example, data input tests are analogous to manufacturing incoming quality control. Figure 88 shows examples of data input, output and business logic tests.

Data input tests strive to prevent any bad data from being fed into subsequent pipeline stages. Allowing bad data to progress through the pipeline wastes processing resources and increases the risk of never catching an issue. It also focuses attention on the quality of data sources, which must be actively managed — manufacturers call this *supply chain management*.

Data output tests verify that a pipeline stage executed correctly. Business logic tests validate data against tried and true assumptions about the business. For example, perhaps all European customers are assigned to a member of the Europe sales team.

Test results saved over time provide a way to check and monitor quality versus historical levels.

|                       |  |
|-----------------------|--|
| <b>Inputs</b>         | <b>Verifying the inputs to an analytics processing stage</b><br>Count Verification - Check that row counts are in the right range, ...<br>Conformity - US Zip5 codes are five digits, US phone numbers are 10 digits, ...<br>History - The number of prospects always increases, ...<br>Balance - Week over week, sales should not vary by more than 10%, ...<br>Temporal Consistency - Transaction dates are in the past, end dates are later than start dates, ...<br>Application Consistency - Body temperature is within a range around 98.6F/37C, ...<br>Field Validation - All required fields are present, correctly entered, ... |
| <b>Business Logic</b> | <b>Checking that the data matches business assumptions</b><br>Customer Validation - Each customer should exist in a dimension table<br>Data Validation - 90 percent of data should match entries in a dimension table  |
| <b>Output</b>         | <b>Checking the result of an operation, for example, a cross-product join</b><br>Completeness - Number of customer prospects should increase with time<br>Range Verification - Number of physicians in the US is less than 1.5 million   |

Figure 88: Tests validate data inputs and outputs and verify that data is consistent with business logic.

## FAILURE MODES

A disciplined data production process classifies failures according to severity level. Some errors are fatal and require the data analytics pipeline to be stopped. In a manufacturing setting, the most severe errors “stop the line.”

Some test failures are warnings. They require further investigation by a member of the data analytics team. Was there a change in a data source? Or a redefinition that affects how data is reported? A warning gives the data-analytics team time to review the changes, talk to domain experts, and find the root cause of the anomaly.

Many test outputs will be informational. They help the data engineer, who oversees the pipeline, to monitor routine pipeline activity or investigate failures.

| Severity      | Required Action         |
|---------------|-------------------------|
| Error         | Stop the pipeline       |
| Warning       | Investigate the failure |
| Informational | Be aware of information |

Table 6: Actions required for different failure modes

## TYPES OF TESTS

The data team may sometimes feel that its work product is *under a microscope*. If the analytics look “off,” users can often tell immediately. They are experts in their own domain and will often see problems in analytics with only a quick glance.

Finding issues before your internal customers do is critically important for the data team. There are three basic types of tests that will help you find issues before anyone else: location balance, historical balance and statistical process control.

## LOCATION BALANCE TESTS

Location Balance tests ensure that data properties match business logic at each stage of processing. For example, an application may expect 1 million rows of data to arrive via [FTP](#). The Location Balance test could verify that the correct quantity of data arrived initially, and that the same quantity is present in the database, in other stages of the pipeline and finally, in reports.

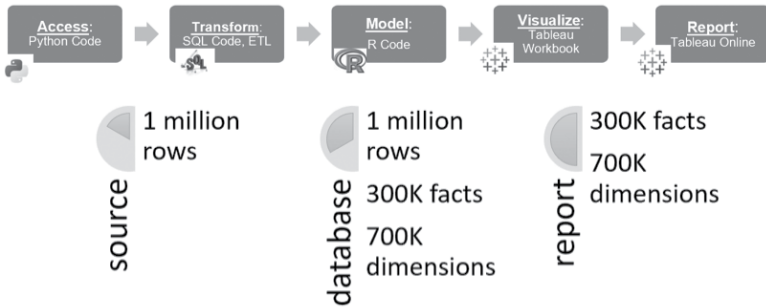


Figure 89: Location Balance Tests verify 1M rows in raw source data, and the corresponding 1M rows / 300K facts / 700K dimension members in the database schema, and 300K facts / 700K dimension members in a Tableau report

## HISTORICAL BALANCE

Historical Balance tests compare current data to previous or expected values. These tests rely upon historical values as a reference to determine whether data values are reasonable (or within the range of reasonable). For example, a test can check the top fifty customers or suppliers. Did their values unexpectedly or unreasonably go up or down relative to historical values?

It's not enough for analytics to be correct. Accurate analytics that “look wrong” to users raise credibility questions. Figure 90 shows how a change in allocations of [SKUs](#), moving from pre-production to production, affects the sales volumes for product groups G1 and G2. You can bet that the VP of sales will notice this change immediately and will report back that the analytics look *wrong*. This is a common issue for analytics — the report is correct, but it reflects poorly on the data team because it *looks wrong* to users. *What has changed?* When confronted, the data-analytics team has no ready explanation. *Guess who is in the hot seat.*

Historical Balance tests could have alerted the data team ahead of time that product group sales volumes had shifted unexpectedly. This would give the data-analytics team a chance to investigate and communicate the change to users in advance. Instead of hurting credibility, this episode could help build it by showing users that the reporting is under control and that the data team is on top of changes that affect analytics. *“Dear sales department, you may notice a change in the sales volumes for G1 and G2. This is driven by a reassignment of SKUs within the product groups.”*

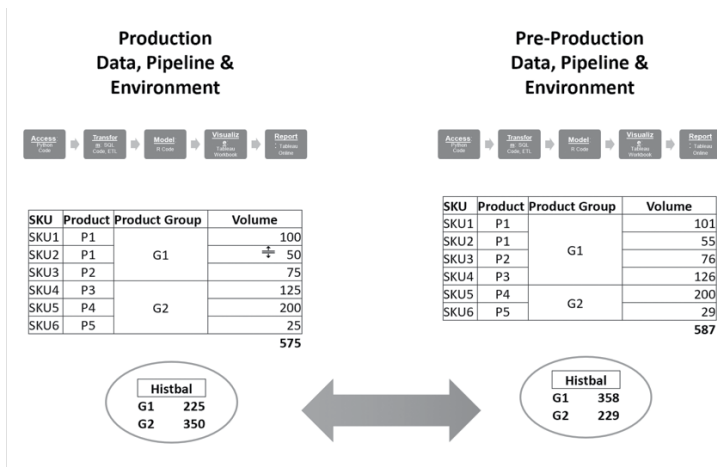


Figure 90: It's not enough for analytics to be correct. Accurate analytics that “look wrong” to users raise credibility questions.

## STATISTICAL PROCESS CONTROL

Lean manufacturing operations measure and monitor every aspect of their process in order to detect issues as early as possible. These are called Time Balance tests or more commonly, [statistical process control](#) (SPC). SPC tests repeatedly measure an aspect of the data pipeline screening for error or warning patterns. SPC offers a critical tool for the data team to catch failures before users see them in reports.

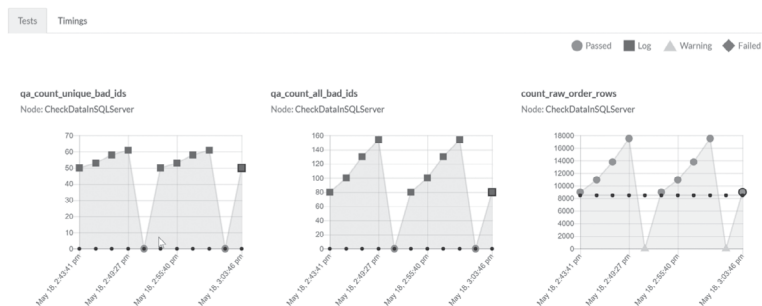


Figure 91: Statistical Process Control tests apply numerical criteria to data-analytics pipeline measurements

## NOTIFICATIONS

A complex process could have thousands of tests running continuously. When an error or warning occurs, a person on the data team should be alerted in real-time through email, text or a notification service like slack. This frees the data team from the distraction of having to periodically poll test results. If and when an event takes place, they'll be notified and can take action.

```
Test Results

Tests: Failed
    No Tests Failed

Tests: Warning
    Step (create-m-location)
        1. compare_raw_rosters (19 equal-to 0)

Tests: Log
    No Tests

Tests: Passed
    Step (put-raw-alignment)
        1. test-T_RHEUM_STRUCTURE-local-row-count (231 equal-to 231)
        2. test-pso-territory-id-in-structure (0 equal-to 0)
        3. test-duplicate-t-zip-terr (0 equal-to 0)
        4. test-T_ZIP_TERR-local-row-count (41294 equal-to 41294)
        5. test-hybrid-territory-id-in-structure (0 equal-to 0)
        6. test-size-structure-history (235 greater-than 230)
```

Figure 92: Example data test result notification email

Automated tests and alerts enforce quality and greatly lessen the day-to-day burden of monitoring the pipeline. The organization's trust in data is built and maintained by producing consistent, high-quality analytics that help users understand their operational environment. That trust is critical to the success of an analytics initiative. After all, trust in the data is really trust in the data team.

# How Data Analytics Professionals Can Sleep Better

## LEAN MANUFACTURING SECRETS THAT YOU CAN APPLY TO DATA ANALYTICS

What could data analytics professionals possibly learn from car manufacturers? It turns out, a lot. Automotive giant Toyota pioneered a set of methods, later folded into a discipline called [lean manufacturing](#), in which employees focus relentlessly on improving quality and reducing non-value-add activities. This culture enabled Toyota to grow into the one of the world's leading car companies. The Agile and DevOps methods that have led to stellar improvements in coding velocity are really just an example of lean manufacturing principles applied to software development.



Conceptually, manufacturing is a pipeline process. Raw materials enter the manufacturing floor through the stock room, flow to different work stations as work-in-progress and exit as finished goods. In data-analytics, data progresses through a series of steps and exits in the form of reports, models and visualizations. Each step takes an input from the previous step, executes a complex procedure or set of instructions and creates output for the subsequent step. At an abstract level, the data-analytics pipeline is analogous to a manufacturing process. Like manufacturing, data analytics executes a set of operations and attempts to produce a consistent output at a high level of quality. In addition to lean-manufacturing-inspired methods like Agile and DevOps, there is one more useful tool that can be taken from manufacturing and applied to data-analytics process improvement.

[W. Edwards Deming](#) championed [statistical process control](#) (SPC) as a method to improve manufacturing quality. SPC uses real-time product or process measurements to monitor and control quality during manufacturing processes. If the process measurements are maintained within specific limits, then the manufacturing process is deemed to be functioning properly. When SPC is applied to the data-analytics pipeline, it leads to remarkable improvements in efficiency and quality. For example, Google executes over one hundred million automated

test scripts per day to validate any new code released by software developers. In the Google consumer surveys group, code is deployed to customers eight minutes after a software engineer finishes writing and testing it.

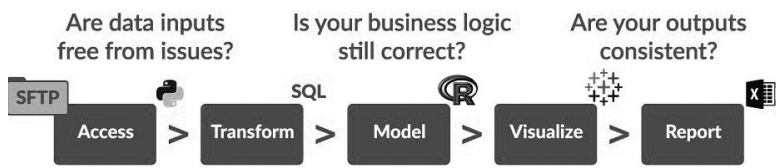


Figure 93: Tests verify the results for each intermediate step in the analytics pipeline.

In data analytics, tests should verify that the results of each intermediate step in the production of analytics matches expectations. Even very simple tests can be useful. For example, a simple row-count test could catch an error in a join that inadvertently produces a Cartesian product. Tests can also detect unexpected trends in data, which might be flagged as warnings. Imagine that the number of customer transactions exceeds its historical average by 50%. Perhaps that is an anomaly that upon investigation would lead to insight about business seasonality.

Tests in data analytics can be applied to data or models either at the input or output of a phase in the analytics pipeline. Tests can also verify business logic.

|                       |  |
|-----------------------|--|
| <b>Inputs</b>         | <b>Verifying the inputs to an analytics processing stage</b><br>Count Verification - Check that row counts are in the right range, ...<br>Conformity - US Zip5 codes are five digits, US phone numbers are 10 digits, ...<br>History - The number of prospects always increases, ...<br>Balance - Week over week, sales should not vary by more than 10%, ...<br>Temporal Consistency - Transaction dates are in the past, end dates are later than start dates, ...<br>Application Consistency - Body temperature is within a range around 98.6F/37C, ...<br>Field Validation - All required fields are present, correctly entered, ... |
| <b>Business Logic</b> | <b>Checking that the data matches business assumptions</b><br>Customer Validation - Each customer should exist in a dimension table<br>Data Validation - 90 percent of data should match entries in a dimension table  |
| <b>Output</b>         | <b>Checking the result of an operation, for example, a cross-product join</b><br>Completeness - Number of customer prospects should increase with time<br>Range Verification - Number of physicians in the US is less than 1.5 million   |

Figure 94: Tests are applied to inputs, outputs or business logic.

The data analytics pipeline is a complex process with steps often too numerous to be monitored manually. SPC allows the data analytics team to monitor the pipeline end-to-end from a big-picture perspective, ensuring that everything is operating as expected. As an automated test suite grows and matures, the quality of the analytics is assured without adding cost. This makes it possible for the data analytics team to move quickly – enhancing analytics to address new challenges and queries – without sacrificing quality.





# DataOps and Your Career

## DataOps Engineer Will Be the Sexiest Job in Analytics

Years ago, prior to the advent of Agile Development, a friend of mine worked as a release engineer. His job was to ensure a seamless build and release process for the software development team. He designed and developed builds, scripts, installation procedures and managed the [version control](#) and issue tracking systems. He played a mean mandolin at company parties too.

The role of release engineer was (and still is) critical to completing a successful software release and deployment, but as these things go, my friend was valued less than the software developers who worked beside him. The thinking went something like this — developers could make or break schedules and that directly contributed to the bottom line. Release engineers, on the other hand, were never noticed, unless something went wrong. As you might guess, in those days the job of release engineer was compensated less generously than development engineer. Often, the best people vied for positions in development where compensation was better.



## RISING FORTUNES

Today, the fortunes of release engineers have risen sharply. In companies that are implementing [DevOps](#) there is no more important person than the release engineer. The job title has been renamed DevOps engineer and it is one of the most highly compensated positions in the field of software engineering. According to salary surveys, experienced DevOps engineers make six figure salaries. DevOps specialists are so hard to find that firms are hiring people without college degrees, if they have the right experience.

Whereas a release engineer used to work off in a corner tying up loose ends, the DevOps engineer is a high-visibility role coordinating the development, test, IT and operations functions. If a DevOps engineer is successful, the wall between development and operations melts away and the dev team becomes more agile, efficient and responsive to the market. This has a huge impact on the organization's culture and ability to innovate. With so much at stake, it makes sense to get the best person possible to fulfill the DevOps engineer role and compensate them accordingly. When DevOps came along, the release engineer went from fulfilling a secondary supporting role to occupying the most sought-after position in the department. Many release engineers have successfully rebranded themselves as DevOps engineers and significantly upgraded their careers.

## DATAOPS FOR DATA ANALYTICS

A similar change, called [DataOps](#), is transforming the roles on the data analytics team. DataOps is a better way to develop and deliver analytics. It applies Agile development, DevOps and lean manufacturing principles to data analytics producing a transformation in data-driven decision making.

Data engineers, data analysts, data scientists — these are all important roles, but they will be valued even more under DataOps. Too often, data analytics professionals are trapped into relying upon non-scalable methods: [heroism, hope or caution](#). DataOps offers a way out of this no-win situation.

The capabilities unlocked by DataOps impacts everyone that uses data analytics — all the way to the top levels of the organization. DataOps breaks down the barriers between data analytics and operations. It makes data more easily accessible to users by redesigning the data analytics pipeline to be more flexible and responsive. It will completely change what people think of as possible in data analytics.

In many organizations, the DataOps engineer will be a separate role. In others, it will be a shared function. In any case, the opportunity to have a high-visibility impact on the organization will make DataOps engineering one of the most desirable and highly compensated functions. Like the release engineer whose career was transformed by DevOps, DataOps will boost the fortunes of data analytics professionals. DataOps will offer select members of the analytics team a chance to reposition their roles in a way that significantly advances their career. If you are looking for an opportunity for growth as a DBA, ETL Engineer, BI Analyst, or another role look into DataOps as the next step.

And watch out [Data Scientist](#), the real [sexiest job of the 21st century](#) is DataOps Engineer.

# Building a DataOps Team

Picture what you could accomplish if your organization had accurate and detailed information about products, processes, customers and the market. If your company does not have a data analytics function, you need to start one. Better yet, if data analytics is not serving as a competitive advantage in your organization, you need to *step up your game* and establish a [DataOps](#) team.

Data analytics analyzes internal and external data to create value and actionable insights. Analytics is a positive force that is transforming organizations around the globe. It helps cure diseases, grow businesses, serve customers better and improve operational efficiency.

In analytics there is mediocre and there is *better*. A typical data analytics team works slowly, all the while living in fear of a high-visibility [data quality](#) issue. A high-performance data analytics team rapidly produces new analytics and flexibly responds to marketplace demands while maintaining impeccable quality. We call this a DataOps team. A DataOps team can [Work Without Fear or Heroism](#) because they have automated controls in place to enforce a high level of quality even as they shorten the cycle time of new analytics by an order of magnitude. Want to upgrade your data analytics team to a DataOps team? It comes down to roles, tools and processes.



## MEET THE DATAOPS TEAM

There are four key roles in any DataOps team. Note that larger organizations will tend to have many people in each role. Smaller companies might have one person performing multiple roles. *See the table down below for some key tools associated with each of the roles described as well as alternate job titles.* Most of these roles are familiar to data analytics professionals, but DataOps adds an essential ingredient that makes the team much more productive.

### DATA ENGINEER

The [data engineer](#) is a software or computer engineer that lays the groundwork for other members of the team to perform analytics. The data engineer moves data from operational systems (ERP, CRM, MRP, ...) into a [data lake](#) and writes the transforms that populate [schemas](#) in data warehouses and data marts. The data engineer also implements [data tests](#) for quality.

## DATA ANALYST

The data analyst takes the data warehouses created by the data engineer and provides analytics to stakeholders. They help summarize and synthesize massive amounts of data. The data analyst creates visual representations of data to communicate information in a way that leads to insights either on an ongoing basis or by responding to ad-hoc questions. Some say that a data analyst summarizes data that reflects past performance ([descriptive analytics](#)) while future predictions are the domain of the data scientist.

## DATA SCIENTIST

Data scientists perform research and tackle open-ended questions. A [data scientist](#) has domain expertise, which helps him or her create new algorithms and models that address questions or solve problems.

For example, consider the inventory management system of a large retailer. The company has a limited inventory of snow shovels, which have to be allocated among a large number of stores. The data scientist could create an algorithm that uses weather models to predict buying patterns. When snow is forecasted for a particular region it could trigger the inventory management system to move more snow shovels to the stores in that area.

| Roles                   | Other Job Titles  | Responsibilities  | Skills   | Tools  |
|-------------------------|---|---|--|--|
| <b>Data Engineer</b>    | Database Architect<br>Data Modeler,<br>Database Administrator,<br>Data QA Engineer,<br>ETL Engineer                                     | Data lakes,<br>Data warehouses,<br>Data marts,<br>Schema design                                 | Databases,<br>Programming,<br>Cloud infrastructure,<br>Simple storage  | SQL,<br>Informatica,<br>DataStage,<br>SSIS,<br>Talend              |
| <b>Data Analyst</b>     | Data Visualization Designer,<br>Business Data Analyst,<br>BI Tableau Developer,<br>Reporting Analyst,<br>Business Intelligence Engineer | Visualizations:<br>Charts,<br>Graphs,<br>Dashboards,<br>Tables,<br>Reports                      | Programming,<br>Statistics,<br>Machine learning,<br>Data cleaning,<br>Data visualization                         | Excel,<br>Looker,<br>Tableau,<br>Qlik View,<br>Altryx,<br>Spotfire |
| <b>Data Scientist</b>   | Machine Learning Researcher,<br>Machine Learning Engineer,<br>Quantitative Analyst,<br>AI Programmer<br>Actuary                         | Algorithms,<br>Models   | Domain subject matter expert,<br>Advanced mathematics,<br>Machine learning,<br>Data mining tools,<br>Programming | R,<br>Python,<br>SAS,<br>SPSS                                      |
| <b>DataOps Engineer</b> |   | Orchestrating the analytic pipeline,<br>Promoting features to production,<br>Automating quality | Agile Development,<br>DevOps,<br>Statistical Process Control   | DataKitchen,<br>data test frameworks,<br>python, shell scripts     |

Table 7: The DataOps Team

## DATAOPS IS THE PROCESS AND THE TOOLS

Many data analytics teams fail because they focus on people and tools and ignore process. This is similar to fielding a sports team with players and equipment, but no game plan describing how everyone will work together. The game plan in data analytics is included in something that we call DataOps.

[DataOps](#) is a combination of tools and process improvements that enable rapid-response data analytics, at a high level of quality. Producing analytics that are responsive, flexible, continuously deployed and quality controlled requires data analytics to draw upon techniques learned in other fields.

- **Agile Development** – an iterative project management methodology that completes software projects faster and with far fewer defects.
- **DevOps** – a software development process that leverages on-demand IT resources and automated test and deployment of code to eliminate the barriers between development (Dev) and operations (Ops). DevOps reduces time to deployment, decreases time to market, minimizes defects, and shortens the time required to resolve issues. DevOps techniques help analytics teams break down the barriers between data and ops (DataOps).
- **Lean Manufacturing** – DataOps utilizes statistical process control (SPC) to monitor and control the data analytics pipeline. When SPC is applied to data analytics, it leads to remarkable improvements in efficiency and quality. With quality continuously monitored and controlled, data analytics professionals can Work Without Fear or Heroism.

The process and tools enhancements described above can be implemented by anyone on the analytics team or a new role may be created. We call this role the DataOps Engineer.

## DATAOPS ENGINEER

The [DataOps Engineer](#) applies [Agile Development](#), [DevOps](#) and statistical process controls to data analytics. He or she orchestrates and automates the data analytics pipeline to make it more flexible while maintaining a high level of quality. The DataOps Engineer uses tools to break down the barriers between operations and data analytics, unlocking a high level of productivity from the entire team.

As DataOps breaks down the barriers between data and operations, it makes data more easily accessible to users by redesigning the data analytics pipeline to be more responsive, efficient and robust. This new function will completely change what people think of as possible in data analytics. The opportunity to have a high-visibility impact on the organization will make DataOps engineering one of the most desirable and highly compensated functions on the data-analytics team.



# DataOps Triple Chocolate Peanut Butter Cookies

by Aarthy Kannan Adityan

## INGREDIENTS AND TOOLS

- 1 cup butter
- 3/4 cup brown sugar
- 3/4 cup white sugar
- 1 1/2 teaspoon vanilla extract
- 3/4 cup chocolate peanut butter
- 2 eggs
- 2 1/3 cup flour
- 1 teaspoon baking soda
- 3/4 cup cocoa powder
- 1 cup semi-sweet chocolate chips
- 1 cup peanut butter chips

## INSTRUCTIONS

1. Preheat oven to 350°F
2. Soften butter to room temperature
3. Line a baking sheet with parchment paper
4. In a large bowl, cream together softened butter, brown sugar and white sugar
5. Add vanilla extract, chocolate peanut butter and eggs and mix well
6. Stir in flour, baking soda and cocoa powder and combine until blended
7. Fold chocolate chips and peanut butter chips into batter
8. Scoop batter onto prepared baking sheet using a cookie or ice-cream scoop, leaving enough space in-between for cookies to expand
9. Bake for 14-16 minutes
10. Transfer cookies to a wire rack to cool

*Serving size: about 20 cookies*

# DataOps Examples and Case Studies

## Grow Sales Using a DataOps-Powered Customer Data Platform

Data analytics can help drive corporate growth by providing customer analytics and ultimately actionable insights to the sales and marketing teams. Unfortunately, the fast-paced, dynamic nature of sales makes it difficult for the customer-facing teams to tolerate the slow and deliberate manner in which analytics is typically produced. In an earlier chapter, we identified [eight major challenges of data analytics](#):

- **The Goalposts Keep Moving** – Sales and marketing requirements change constantly and the requests for new analytics never cease.
- **Data Lives in Silos** – Data is collected in separate operational systems and typically, none of these systems talk to each other.
- **Data Formats are not Optimized** – Data in operational systems is usually not structured in a way that lends itself to the efficient creation of analytics.
- **Data Errors** – Data will eventually contain errors, which can be difficult to resolve quickly.
- **Bad Data Ruins Good Reports** – When data errors work their way through the data pipeline into published analytics, internal stakeholders can become dissatisfied. These errors also harm the hard-won trust in the [analytics team](#).
- **Data Pipeline Maintenance Never Ends** – Every new or updated data source, schema enhancement, analytics improvement or other change triggers an update to the data pipeline. These updates may be consuming 80% of your team's time.
- **Manual Process Fatigue** – Manual procedures for data integration, cleansing, transformation, quality assurance and deployment of new analytics are error-prone, time-consuming and tedious.
- **The Trap of “[Hope and Heroism](#)”** – To cope with the above challenges, data professionals work long hours, make changes (without proper testing) and “hope” for the best or just retreat into a posture of over-caution in which projects just execute more slowly.



## OVERCOMING THE EIGHT CHALLENGES

If you have managed an analytics team for any period of time, you have likely encountered these and similar challenges. However, you don't have to accept the status quo. It is possible to implement processes and methodologies that address these challenges and enable your data-analytics team to improve their productivity by an order of magnitude while achieving a higher level of [data quality](#). In this new approach to customer and market analytics, the data-analytics team executes at previously unimaginable speed, efficiency and quality:

**Rapid-Response Analytics** – The sales and marketing team will continue to demand a never-ending stream of new and changing requirements, but the data-analytics team will delight your sales and marketing colleagues with rapid responses to their requests. New analytics will inspire new questions that will, in turn, drive new requirements for analytics. The feedback loop between analytics and sales/marketing will iterate so quickly that it will infuse excitement and creativity throughout the organization. This will lead to breakthroughs that vault the company to a leadership position in its markets.

**Data Under Your Control** – Data from all of the various internal and external sources will be integrated into a consolidated database that is under the control of the data-analytics team. Your team will have complete access to it at all times, and they will manage it independently of IT, using their preferred tools. With data under its control, the data-analytics team can modify the format and architecture of data to meet its own operational requirements.

**Impeccable Data Quality** – As data flows through the data-analytics pipeline, it will pass through tests and filters that ensure that it meets quality guidelines. Data will be monitored for anomalies 24x7, preventing bad data from ever reaching sales and marketing analytics. You'll have a dashboard providing visibility into your data pipeline with metrics that delineate problematic data sources or other issues. When an issue occurs, the system alerts the appropriate member of your team who can then fix the problem before it ever receives visibility. As the manager of the data-analytics team, you'll spend far less time in uncomfortable meetings discussing issues and anomalies related to analytics.

**Automated Efficiency** – Data feeds and new analytics will be deployed using automation, freeing the data-analytics team from tedious manual processes. The analytics team will be able to focus on its highest priorities – creating new analytics for sales and marketing that create value for the company.

The processes, methodologies and tools required to realize these efficiencies combine two powerful ideas: The Customer Data Platform (CDP) and a revolutionary new approach to analytics called [DataOps](#). Below we'll explain how you can implement your own Data-Ops-powered CDP that improves both your analytics cycle time and data-pipeline quality by 10X or more.

### CUSTOMER DATA PLATFORM

A Customer Data Platform (CDP) provides sales and marketing with a unified view of all customer-related data whether internal or external, in a single integrated database. Once setup, a CDP enables the analytics team to create and manage customer data themselves, without reliance upon resources from IT or other departments. This helps sales and marketing better leverage the company's valuable data while responding to market demands quickly and proactively. The figure below shows how a CDP consolidates data from numerous databases. Each operational database becomes a data source that continuously feeds a copy of its data into a centralized CDP database.

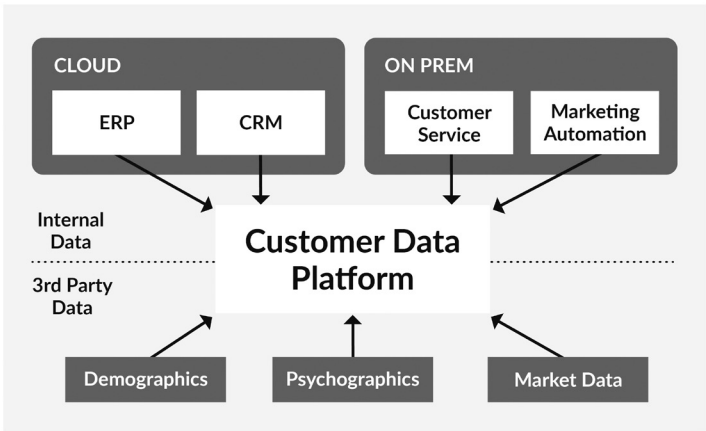


Figure 95: The Customer Data Platform consolidates data from operational systems to provide a unified customer view for sales and marketing.

### DATAOPS

A CDP is a step in the right direction, but it won't provide much improvement in team productivity if the team relies on cumbersome processes and procedures to create analytics. DataOps is a set of methodologies and tools that will help you optimize the processes by

which you create analytics, manage the data-analytics pipeline and automatically deploy new analytics and data. [DataOps](#) rests on three foundational principles:

**Agile Development** – DataOps utilizes a methodology called Agile Development to minimize the cycle time for new data analytics. Studies show that software development projects using Agile complete significantly faster and with far fewer defects.

**DevOps** – In DataOps, new analytics production is automated and monitored. Automated tests verify new analytics before publishing them to sales and marketing users. This allows the analysts to focus less on the mechanics of deploying analytics and more on the creation of new insights that address sales and marketing requests. In the software development domain, the automated deployment of code is called DevOps. Prominent software industry leaders use DevOps to publish software updates many times per second while assuring quality. DataOps incorporates DevOps methods and principles to publish new analytics and data in an automated fashion.

**Statistical Process Control** – DataOps employs a methodology called Statistical Process Control (SPC) to assure [data quality](#) using end-to-end data pipeline automation and quality controls. SPC is a lean manufacturing method that institutes continuous testing on data flowing from sources to users, ensuring that data stays within statistical limits and remains consistent with business logic. SPC monitors data and verifies it 24x7. If an anomaly occurs, SPC notifies the data-analytics team via an automated alert. This reduces the operational burden on team members while improving data quality and reliability. Also, the quantity of data and the number of data sources can more easily scale independently of the size of the [data engineering](#) team.

When implemented in concert, Agile, DevOps and SPC take the productivity of data-analytics professionals to a whole new level. DataOps will help you get the most out of your data, human resources and integrated CDP database.

# Achieving Growth Targets by Implementing a DataOps-Powered Customer Data Platform

As the leader of a data-analytics organization, your mission is to utilize data from operational systems and other sources to create insights that help the organization achieve its growth targets. Figure 96 provides a conceptual view of this flow. The dark boxes show the domain that is under the control of the analytics leader. It includes people, tools, technologies and processes that together comprise the DataOps-powered CDP.

## The Eight Challenges of Data Analytics

1. The Goalposts Keep Moving
2. Data Lives in Silos
3. Data Formats are not Optimized
4. Data Errors
5. Bad Data Ruins Good Reports
6. Data Pipeline Maintenance Never Ends
7. Manual Process Fatigue
8. The Trap of Hope and Heroism



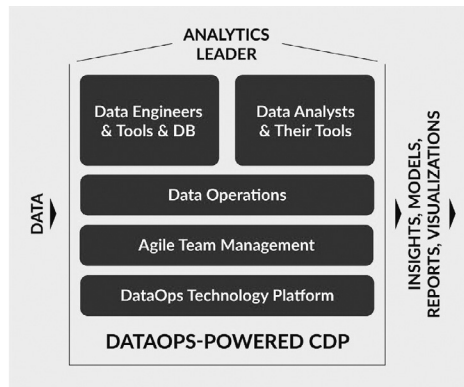


Figure 96: The resources under data-analytics control that leverage data to meet business objectives

## DATA ANALYSTS AND THEIR TOOLS

The Data Analyst works to satisfy the needs of the sales and marketing department by continually delivering insights. The data analyst creates visual representations of data to communicate information in a way that leads to insights either on an ongoing basis or by responding to ad-hoc questions. With a DataOps-powered CDP, data analysts work with autonomy and speed, drawing analytics from CDP data. Analysts use tools like Tableau and Alteryx to create these insights and independently promote their investigative work into production deliverables as needed.

Every resource, technology and tool in the data-analytics organization exists to support the data analyst's ability to serve Sales and Marketing. This also applies to [Data Scientists](#) who also deliver insights directly to Sales and Marketing colleagues.

## DATA ENGINEERS AND THEIR TOOLS AND DATABASES

The [Data Engineer](#) works with the IT department and all data source providers to institute automated processes that move data from various data sources into a trusted, integrated CDP database under the complete control of the [data-analytics team](#). The CDP database may include a data lake, which provides revision history, easy access, control and error recovery.

The engineer writes transforms that operate on the [data lake](#), creating data warehouses and data marts used by data analysts and scientists. The data engineer also implements tests that monitor data at every point along the data-analytics pipeline assuring a high level of quality.

The data engineer lays the groundwork for other members of the team to perform analytics without having to be operations experts. With a dedicated data engineering function, DataOps provides a high level of service and responsiveness to the data-analytics team.

## DATA OPERATIONS

The DataOps-powered CDP provides [automated](#) support for the creation, monitoring and management of the end-to-end data pipeline. This includes stewardship of every aspect of the journey from data sources to reporting. Statistical Process Control (SPC) data-quality tests monitor each stage of the automated pipeline, alerting data engineering when data fails to meet statistical controls or match business logic.

With tests monitoring each stage of the automated data pipeline, DataOps can produce a dashboard showing the status of the pipeline. The DataOps dashboard provides a high-level overview of the end-to-end data pipeline. Is any data failing quality tests? What are the error rates? Which are the troublesome data sources? With this information at his or her fingertips, the Data Engineer can proactively improve the data pipeline to increase robustness. In the event of a high-severity data anomaly, an alert is sent to the Data Engineer who can take steps to protect production analytics and work to resolve the error. If the anomaly relates to a data supplier, data engineering can work with the vendor to drive the issue to resolution. Workarounds and data patches can be implemented as needed with information in release notes for users. In many cases, errors are resolved without the users (or the organization's management) ever being aware of any problem.

## AGILE TEAM MANAGEMENT

The Agile methodology governs the creation of new analytics, producing a steady stream of valuable innovations and improvements to analytic insights in short increments of time. Agile is particularly effective in environments where requirements are quickly evolving — a situation all too familiar to data-analytics professionals. Agile development is not only a method; it is also a philosophy and a mindset. Developers collaborate with sales/marketing customers, respond to change, measure progress through “delivered analytics,” release frequently, seek feedback on releases, and adjust behavior to become more effective.

## DATAOPS PLATFORM

The various methodologies, processes, people (and their tools) and the CDP analytics database are tied together cohesively using a technical environment called a DataOps Platform. The DataOps Platform includes support for:

- Agile project management
- Deployment of new analytics
- Execution of the data pipeline (orchestration)
- Integration of all tools and platforms
- Management of development and production environments
- Source-code [version control](#)
- Testing and monitoring of [data quality](#)
- Data Operations reporting and dashboards

The high degree of automation offered by DataOps eliminates a great deal of work that has traditionally been done manually. This frees up the team to create new analytics requested by stakeholder partners.

A [DataOps Platform](#) is not a one-size-fits-all tool. It is the central application that coordinates the various tools that drive your orchestration, testing, deployment, model deployment, development-environment management, change management, and data integration. You can create your own DataOps Platform from scratch, although partnering with a supplier can reduce time to market. Working with a partner also gives you the option of treating the entire CDP and data pipeline as a managed service, which can be initially outsourced and then partly or entirely taken over by internal resources at a later time.

## DATAKITCHEN DATAOPS-CDP SOLUTION

An enterprise can outsource [data engineering](#), databases, data operations, Agile team management and the DataOps Technology Platform to gain efficiencies. DataKitchen offers a [DataOps Technology Platform](#) as well as managed services for each of the gray boxes in figure 97. In essence, DataKitchen offers all aspects of the DataOps-powered CDP except for data analysis and data science, which rely upon vertical market expertise and close collaboration with sales and marketing.

The enterprise can also outsource the functions shown initially but insource them at a later date. Once set-up, the DataOps Platform can be easily and seamlessly transitioned to an internal team.

Customer Data Platforms promise to drive sales and improve the customer experience by unifying customer data from numerous disjointed operational systems. As a leader of the analytics team, you can take control of sales and marketing data by implementing efficient analytics-creation and deployment processes using a DataOps-powered CDP. A DataOps platform makes analytics responsive and robust. This enables your data analysts and scientists to rise above the bits and bytes of data operations and focus on new analytics that help the organization achieve its goals.

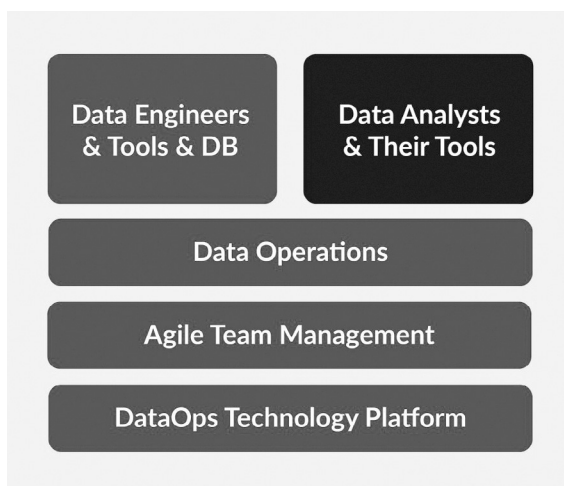


Figure 97: DataKitchen DataOps-Powered CDP and Managed Services

# How a Mixed Martial Arts Fighter Would Approach Data Analytics

By James Royster, Director, Commercial Analytics, Celgene

Mixed Martial Arts (MMA) combines striking, wrestling and other fighting techniques into a unified sport. Every martial art and fighting technique has its strengths and strategic advantages. Boxing is known for punching but also provides footwork, guard position and head movement. Wrestling relies upon takedowns. Karate features striking techniques such as kicking. MMA is a hybrid of all of these (and many more) drawing upon each mode of combat as needed for a given competitive situation. If an MMA athlete competed against a boxer or karate expert, the mixed martial artist would clearly have an unfair advantage. MMA's real strength is its versatility and its ability to absorb new methods.



[DataOps](#) is the mixed martial arts of data analytics. It is a hybrid of Agile Development, DevOps and the [statistical process controls](#) drawn from lean manufacturing. Like MMA, the strength of DataOps is its readiness to evolve and incorporate new techniques that improve the quality, reliability, and flexibility of the data analytics pipeline. DataOps gives data analytics professionals an unfair advantage over those who are doing things the old way — using [hope, heroism or just going slowly](#) in order to cope with the rapidly changing requirements of the competitive marketplace.

Agile development has revolutionized the speed of software development over the past twenty years. Before Agile, development teams spent long periods of time developing specifications that would be obsolete long before deployment. Agile breaks down software development into small increments, which are defined and implemented quickly. This allows a development team to become much more responsive to customer requirements and ultimately accelerates time to market.

Data analytics shares much in common with software development. Conceptually, the data analytics pipeline takes raw data, passes it through a series of steps and turns it into actionable information. Files, such as scripts, code, algorithms, configuration files, and many others, drive each processing stage. These files, taken as a whole, are essentially just code. As a coding endeavor, data analytics has the opportunity to improve implementation speeds by an order of magnitude using techniques like Agile development. DevOps offers an additional opportunity for improvement.

The difficulty of procuring and provisioning physical IT resources has often hampered data analytics. In the software development domain, leading-edge companies are turning to DevOps, which utilizes cloud resources instead of on-site servers and storage. This allows developers to procure and provision IT resources nearly instantly and with much greater control over the run-time environment. This improves flexibility and yields another order of magnitude improvement in the speed of deploying features to the user base.

DataOps also incorporates lean manufacturing techniques into data analytics through the use of statistical process controls. In manufacturing, tests are used to monitor and improve the quality of factory-floor processes. In DataOps, tests are used to verify the inputs, business logic, and outputs at each stage of the data analytics pipeline. The data analytics professional adds a test each time a change is made. The suite of tests grows over time until it eventually becomes quite substantial. The tests validate the quality and integrity of a new release when a feature set is released to the user base. Tests allow the data analytics professional to quickly verify a release, substantially reducing the amount of time spent on deploying updates.

[Statistical process control](#) also monitors data, alerting the data team to an unexpected variance. This may require updates to the business logic built into the tests, or it might lead data scientists down new paths of inquiry or experimentation. The test alerts can be a starting point for creative discovery.

The combination of Agile development, DevOps, and statistical process controls gives DataOps the strategic tools to reduce time to insight, improve the quality of analytics, promote [reuse](#) and refactoring and lower the marginal cost of asking the next business question. Like mixed martial arts, DataOps draws its effectiveness from an eclectic mix of tools and techniques drawn from other fields and domains. Individually, each of these techniques is valuable, but together they form an effective new approach, which can take your data analytics to the next level.

# Reinvent Marketing Automation with the DataKitchen DataOps Platform

A global pharmaceutical giant sought to drive top-line growth by modernizing its marketing operations. The project included a migration to Salesforce Marketing Cloud, integrations with numerous internal and third-party data sources, and a continuous flow of data. The plan initially required eighteen months for implementation. Using the [DataKitchen DataOps Platform](#), which automates deployment, controls quality and supports [Agile development](#) of analytics, the company was able to start delivering value in six weeks and completed the migration in about one third the time.

## THE CHALLENGE OF MARKETING AUTOMATION AT SCALE

The company faced numerous difficulties when implementing marketing automation for multiple global business units:

- **Distributed Data** – The company's marketing analytics and customer data were distributed in many specialized systems that do not easily talk to each other. This made it challenging to link customer data from one system with another. Customers engage through emails, partner websites, advertisements, campaigns and across product lines. Each of these touchpoints produces a continuous stream of fine-grained opt-in/opt-out requests which all must be consolidated and synchronized. Cross-referencing customer data with third-party databases also provides valuable segmentation information.
- **Supporting Agile** – The company lacked the technical infrastructure to implement Agile development of data flows and analytics. Using a slow and inflexible development process prevented them from keeping up with the fast-paced requirements of the sales and marketing teams.
- **Iterating on Data Quality** – Analysts had trouble specifying [data quality](#) rules until they saw the data. In a non-agile environment, this caused requirements to keep changing, causing delays.
- **Continuous Change Requests** – Once a system is operational, users are inspired to request additional data sources, segmentations and other enhancements. With long development cycles, it was difficult for the team to keep up with the users' continuous demands.

## AUTOMATED EFFICIENCY AND QUALITY WITH DATAKITCHEN

The company utilized the DataKitchen Platform to oversee and monitor the end-to-end data pipeline. With the DataKitchen solution, the company is now able to:

- **Automate and monitor data pipelines** – Data flows from sources to user analytics in a continuous pipeline.
- **Implement continuous deployment** – New analytics are tested and deployed to users with speed and confidence using automation.

- **Control quality** – Data is continuously monitored for anomalies with alerts and dashboards that provide real-time information about data quality and operations.
- **Manage sandboxes** – [Development environments](#) are created as needed to prevent enhancements from disrupting operations.

## Marketing Ops - solution

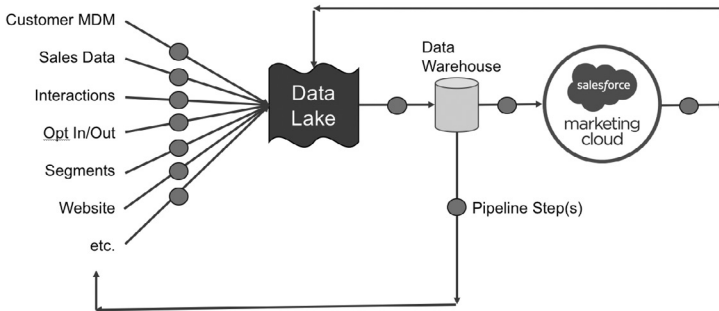


Figure 98: With DataKitchen, marketing automation data flows continuously from numerous sources through the analytics pipeline with efficiency and quality.

## BUSINESS IMPACT

With the [DataKitchen Platform](#), the company was able to break the long 18-month project into sprints and began to deliver value in six weeks. The agility of the DataKitchen [DataOps](#) approach enabled the analytics team to rapidly respond to changing user requirements with a continuous series of enhancements. Users no longer waited months to add new data sources or make other changes. The team can now deploy new data sources, update schemas and produce new analytics quickly and efficiently without fear of disrupting the existing data pipelines.

DataKitchen's lean manufacturing control helped the team be more proactive addressing [data quality](#) issues. With monitoring and alerts, the team is now able to provide immediate feedback to data suppliers about issues and can prevent bad data from reaching user analytics.

All this has led to improved insight into customers and markets and higher impact marketing campaigns that drive revenue growth.

DataKitchen's DataOps Platform helped this pharmaceutical company achieve its strategic goals by improving analytics quality, responsiveness, and efficiency. DataKitchen software provides support for improved processes, automation of tools, and [agile development](#) of new analytics. With DataKitchen, the analytics team was able to deliver value to users in 1/10th the time, accelerating and magnifying their impact on top line growth.

# Meeting the Product Launch Challenge with DataOps

Celgene is a \$12B biopharmaceutical company committed to delivering innovative treatments for patients worldwide. Celgene relies upon [DataKitchen](#) to enable rapid-response, high-quality data analytics that help the company maximize product lifetime revenue.

*“So much of what we do involves business questions that are fire drills. Executives want answers as quickly as possible. The infrastructure that we’ve set-up with DataKitchen allows us to mix and match data in new ways so that we can quickly get the answer to a question.”*

*—Manager, Data Analyst, Celgene*

It costs between \$2-3B to bring a new pharmaceutical to market. When a new drug is introduced, it is already halfway through its patent life. This makes the first 6-12 months of a pharmaceutical launch critical to a product’s lifetime revenue. The vendor needs up-to-date information to allocate samples, plan marketing events, and monitor progress vs. goals. With so much at stake, pharmaceutical companies like Celgene make strategic investments to maximize product adoption and adherence during the initial phase of a drug product’s life cycle.

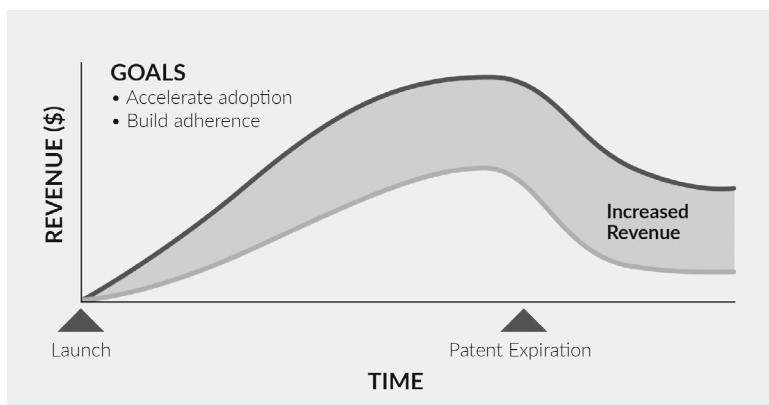


Figure 99: The first year is critical to a pharmaceutical product’s lifetime revenue.

## THE ROLE OF DATA IN LAUNCH SUCCESS

Celgene has found that using analytics to understand customers and markets can significantly improve product launch success. The analytics produced range from weekly, standardized reports to ad-hoc analyses. In the first months of a product’s lifecycle, the sales and marketing teams can’t wait weeks or months for new analytics. Every new data source, question, and innovative idea demands an immediate and accurate response.

THE OBSTACLES TO RESPONSIVE, HIGH-QUALITY ANALYTICS

The Celgene [data engineering](#) and analytics teams faced many obstacles that prevented analytics responsiveness and quality. Data was organized in silos—using a variety of technologies and isolated platforms. Without the right processes and tools in place, the data engineering and analytics teams can spend a majority of their time on data engineering and pipeline maintenance. This distracts them from their main mission — producing analytic insights that help the business attain its objectives.

THE DATAKITCHEN PLATFORM

Celgene chose DataKitchen to enable data engineering that streamlines development and data operations processes, helping the data analysts and scientists to remain focused on creating value — at the speed of business. With the DataKitchen Platform, Celgene has been able to:

- **Automate orchestration** – DataKitchen automates the deployment of new analytics and performs data pipeline orchestration, freeing up the team from manual processes and enabling them to focus on extracting value from data.
- **Monitor data quality** – [Statistical process control](#) and dashboards help monitor and control the quality of the end-to-end data pipeline, and real-time alerts provide high-level visibility into incidents and provide critical information.
- **Automate deployment** – Once new features pass all their tests, just a push of a button is required to deploy into production, with confidence.

| DATA SOURCES & INTEGRATIONS            |
|--|
| IQVIA (IMS)                            |
| Symphony                               |
| Veeva (salesforce.com)                 |
| Specialty Pharmacies                   |
| Email & Web Interactions               |
| Sales Alignments Targets and Prospects |
| Product Hierarchies                    |
| Customer Segments                      |

Table 8

ANALYTICS AT SUPER SPEED

With help from [DataKitchen](#), Celgene was able to improve the productivity of the data engineering function by an order of magnitude. With automation of their data pipeline, they were able to update 30X more visualizations per week. They were able to compress the

| METRIC                                       | BEFORE           | WITH DATAKITCHEN |
|--|------------------|------------------|
| Data analysts supported by one data engineer | 0.5              | 12               |
| Schema changes per week by one data engineer | 1                | 12               |
| Sales people supported by one data analyst   | 50               | 250              |
| Cycle time to publish new visualizations     | weeks/<br>months | next<br>day      |
| Visualizations updated/wk                    | 50               | 1500             |

Table 9

cycle time required to produce new analytics from weeks (or months) to one day. This enabled the data analytics team to successfully address the volume of questions from sales and marketing, helping them maximize product adoption during the critical first phase of their product launch. Instead of just fighting fires, the data team felt like they had acquired *superpowers*.

## MOVING FORWARD WITH DATAKITCHEN

Use of DataKitchen fostered a tight collaboration between data analytics and the business unit, unlocking creativity made possible by [DataOps](#) agility and quality. Impressed by the performance of the data-analytics team using DataKitchen, Celgene decided to expand their use of DataKitchen throughout the company.

*“DataKitchen has enabled us to become nimble and agile when it comes to data. We are now a self-service data organization – from the marketing department to the sales reps.”*

*—Director, Market Insights, Celgene*



# DataOps Classic Baked Macaroni and Cheese

*by Joanne Ferrari*

## INGREDIENTS AND TOOLS

- 2 Cups Milk
- 2 Tablespoons Butter
- 2 Tablespoons All-Purpose Flour
- ½ Teaspoon Salt
- ¼ Teaspoon Freshly Ground Black Pepper
- 1 (10 oz.) Block Extra Sharp Cheddar Cheese, Shredded
- ¼ Teaspoon Ground Red Pepper (Optional)
- ½ (16 oz.) Package Elbow Macaroni, Cooked

## INSTRUCTIONS

1. Preheat oven to 400°. Microwave milk at HIGH for 1 ½ minutes. Melt butter in a large skillet or Dutch oven over medium-low heat; whisk in flour until smooth. Cook, whisking constantly for 1 minute.
2. Gradually whisk in warm milk and cook, whisking constantly 5 minutes or until thickened.
3. Whisk in salt, black pepper, 1 cup shredded cheese, and if desired, red pepper until smooth; stir in pasta. Spoon pasta mixture into a lightly greased 2-qt. baking dish; top with remaining cheese. Bake at 400° for 20 minutes or until golden and bubbly.

## NOTES

For this recipe, it is recommended that you grate the block(s) of cheese. I combine Sharp Cheddar and Swiss cheeses — my favorite. Pre-shredded varieties won't give you the same sharp bite or melt into creamy goodness over your macaroni as smoothly as block cheese that you grate yourself. You can go reduced-fat (but then it's even more important to prep your own). Grating won't take long, and the rest of this recipe is super simple. Use a pasta that has plenty of nooks to capture the cheese—like elbows, shells, or cavatappi. Try it just once, and I guarantee that Classic Baked Macaroni and Cheese will become your go-to comfort food.

# DataOps Survey

## Tomorrow's Forecast: Cloudy with a Chance of Data Errors

### KEY FINDINGS OF THE 2019 DATAOPS SURVEY

Whatever you were planning to accomplish this week — forget about it!

Chances are good that you'll be interrupted by data errors. That is the clear message communicated by the respondents to a joint DataOps survey conducted by DataKitchen and Eckerson Research.

The survey contains feedback from 300 data-analytics professionals who work for medium to large-sized companies across multiple industries in the US and Europe. The full report will be available in June 2019, but here are the key results. The survey sheds light upon three issues that embody the analytics industry's challenges. From a recent NewVantage Partners Report, we know that despite the hype and investment, the number of companies that identify as data-driven is declining. Gartner estimated that 60% of big data projects fail. We see results in our survey that may help explain this.

### THE IMPACT OF DATA ERRORS

One egregious issue is that analytics too often contain errors which erode the credibility of the data team. 30% of respondents to the DataOps survey reported more than 11 errors per month. This is a staggering figure. That means that the data team is probably moving from fighting one fire to next with little time for any value-add activity. The managers of these enterprises learn not to trust the data. Is it any wonder that companies are becoming less data-driven?

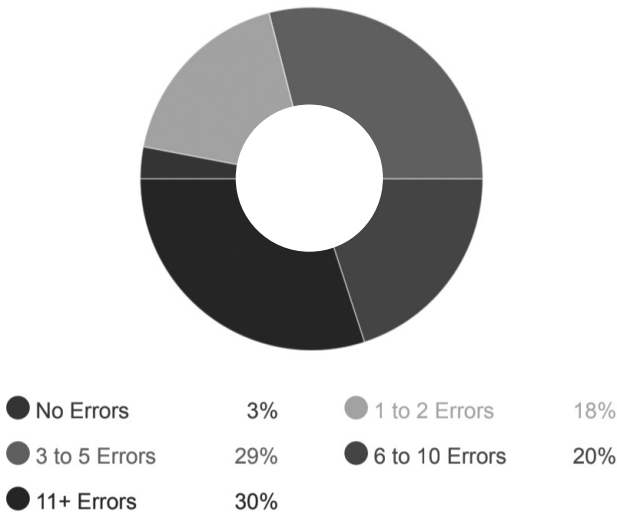
The DataOps enterprises that DataKitchen works with have less than 1 data error per year. Only 3% of the companies surveyed approached that level of quality. Another 18% reported 1-2 errors per month. Would W. Edwards Deming have considered that an acceptable failure rate? Would Toyota? In a manufacturing setting, each one of these errors could be the equiv-

alent of a product recall. For the sake of discussion, let's presume that 1-2 errors per month in an enterprise analytics context are tolerable (it's really not). That still leaves nearly 80% of companies surveyed reporting 5, 10 or even more errors per month. That has a big impact on how an organization views its data and may even explain why the average tenure of a Chief Data Officer is only around 2 years.

### THE IMPACT OF DATA ERRORS

Data errors negatively impact the productivity of the analytics teams in several ways. They flood Kanban boards with new tasks. They cause unproductive context switches. Wary of making further errors, the data team may become overly cautious, working more slowly. In short, data errors are a major bottleneck that affect the entire workflow of new analytics development. We call this analytics cycle time, and it is one of the critical bottlenecks that slow the ability of analytics to create value for an enterprise.

**On average, how many errors (e.g., incorrect data, broken reports, late delivery, customer complaints) do you have each month?**



A short cycle time enables an analytics team to respond quickly to requests for new analytics. When analytics are produced quickly, the data team can keep pace with the endless stream of requests from the business unit. A short cycle time fosters close collaboration with business users and in our experience this unlocks an organization's creativity. However, the cycle time in most data organizations is plagued with inefficient manual processes, bureaucracy, lack of task coordination and dependencies on bottlenecks. For example, survey respondents provided interesting feedback about the time it takes to create an analytics development environment.

## MOST COMPANIES HAVE WAY TOO MANY ERRORS PER MONTH

**79%**

### DELAYS IN ENVIRONMENT CREATION

Isolated development environments are an important way that analytics development can proceed without impacting data operations. If it takes weeks or months for IT to provision hardware and software or to make data available, the queue time severely impacts the productivity of analytics professionals.

78% of respondents indicated that it takes days, weeks or months to create a development environment. 38% of users surveyed report that it took weeks or months. That wait time prevents the data analytics team from even beginning to work on the critical analytics that the organization has requested. This means that their time-to-value is much slower than it should be.

**On average, how long does it take your team to create a new development environment with the appropriate test data, servers, and tools?**



● Minutes

12%

● Hours

10%

● Days

40%

● Weeks

28%

● Months

10%

DataKitchen Interpretation

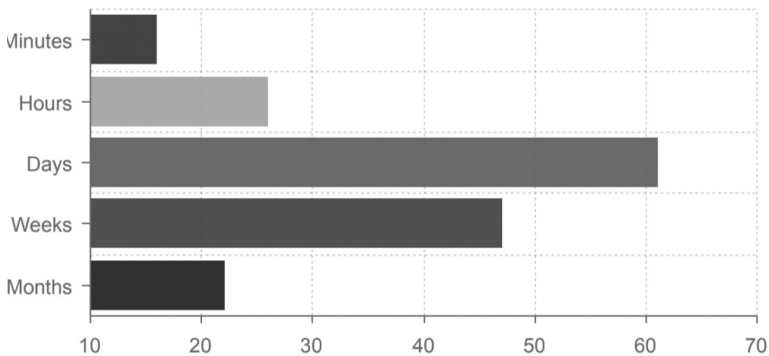
**MOST COMPANIES ARE VERY SLOW  
CREATING NEW DEVELOPMENT  
ENVIRONMENTS**

**78%**

## LENGTHY DEPLOYMENT TIMES

We asked our survey respondents directly about the end-to-end cycle time of creating new analytics. In light of the above, it is not surprising that we see that far too many organizations undergo lengthy periods of time to create and deploy analytics. 76% of organizations surveyed take days, weeks or months to move from analytics development to production. If one data engineer needs to support a dozen data analysts and each data analyst needs to support hundreds of salespeople, the cycle time for new analytics must be reduced to hours or better yet, minutes. Enterprises can reach these performance targets with a DataOps approach to analytics development and deployment

**On average, how long does it take to move a new or modified data  
analytic pipeline from development to production?**



DataKitchen Interpretation

**MOST COMPANIES ARE TOO SLOW TO  
DEPLOY CHANGES INTO PRODUCTION**

**76%**

## CONCLUSION

The three survey responses above help explain why enterprises are not able to respond to user requests for new and updated analytics in a reasonable time frame. For organizations that suffer from these constraints, the case could be made that nothing else matters but addressing these bottlenecks. The ability to rapidly produce and deploy analytics is at the heart of a data team's ability to add value. If viewed from the perspective of the Theory of Constraints, these bottlenecks limit the overall throughput of value creation. Any improvement in analytics development cycle time improves the overall throughput of the system. An improvement other than in the bottleneck is an illusory accomplishment when an analytics team suffers from long development cycle times.

**DataOps offers a way to reduce errors, shorten the time it takes to set up a development environment, and minimize analytics development cycle time. Nothing could state more clearly why analytics organizations need a DataOps initiative now.**

## **Additional Recipes**

# DataOps Vegan Corn Chowder

by Eran Strod

## INGREDIENTS AND TOOLS

### Cashew Cream

- 1 cup cashews soaked in water for at least 2 hours
- 2 cups veg stock
- 4 teaspoons cornstarch (can sub tapioca starch if desired)
- Drain the cashews. In a blender, combine all the ingredients and work for 2 to 5 minutes or until smooth, scraping down the sides with a rubber spatula several times. Set aside.

### Soup

- 1 Tablespoon olive oil
- 1 large onion coarsely chopped
- 2 celery ribs, chopped
- 3 cups veg broth
- 1 large carrot chopped
- 1 red pepper diced (could sub 1 bag Frozen mixed-vegetables, thawed in a pinch)
- 1 potato, diced
- 3 ears of fresh corn (cut the kernels off and scrape the corn cobs for corn milk to add to the soup)
- Can of corn

## INSTRUCTIONS

1. Heat the oil in 4-quart pot
2. When hot add onion and celery with a pinch of salt, cook until start to soften.
3. Add carrots and potatoes
4. Add corn and red pepper and stir-fry for 10 minutes
5. Add 3 cups veg stock and the corn milk
6. Bring to a boil, lower the head and cover — simmer 10 min or until veg tender but not overcooked.
7. Stir in Cashew Cream and stir gently for 7 minutes until nicely thickened.
8. Blend up to half the soup to make more liquid and add it back in
9. Add salt & pepper to taste, depending on the type of veg stock you used.

*My own adaptation of a vegan New England Clam Chowder recipe from the Boston Globe from Isa-does-it by Isa Chandra Moskowitz*



# DataOps Resources

|                             |   |
|-----------------------------|---|
| The Agile Manifesto         | <a href="http://agilemanifesto.org/">http://agilemanifesto.org/</a>   |
| DatOps Blog                 | <a href="http://bit.ly/2Ef2Hto">http://bit.ly/2Ef2Hto</a>   |
| The DataOps Manifesto       | <a href="http://dataopsmanifesto.org">http://dataopsmanifesto.org</a>   |
| DataOps News                | <a href="http://bit.ly/2ORDIUr">http://bit.ly/2ORDIUr</a>   |
| DataOps SlideShare          | <a href="http://bit.ly/2PygnSb">http://bit.ly/2PygnSb</a>   |
| DataOps Videos              | <a href="http://bit.ly/2UFcKO8">http://bit.ly/2UFcKO8</a>   |
| Scrum Guides                | <a href="http://www.scrumguides.org">http://www.scrumguides.org</a>   |
| Statistical Process Control | <a href="https://en.wikipedia.org/wiki/Statistical_process_control">https://<br/>en.wikipedia.<br/>org/wiki/Statistical_process_control</a> |
| W. Edwards Deming           | <a href="https://en.wikipedia.org/wiki/W._Edwards_Deming">https://<br/>en.wikipedia.<br/>org/wiki/W._Edwards_Deming</a>                     |
| Wikipedia DataOps           | <a href="http://bit.ly/2DnlqR1">http://bit.ly/2DnlqR1</a>   |
| Wikipedia DevOps            | <a href="https://en.wikipedia.org/wiki/DevOps">https://en.wikipedia.org/wiki/DevOps</a>   |

# About the Authors

**Christopher Bergh** is a Founder and Head Chef at DataKitchen where, among other activities, he is leading DataKitchen's DataOps initiative. Chris has more than 25 years of research, engineering, analytics, and executive management experience.

Previously, Chris was Regional Vice President in the Revenue Management Intelligence group in Model N. Before Model N, Chris was COO of LeapFrogRx, a descriptive and predictive analytics software and service provider. Chris led the acquisition of LeapFrogRx by Model N in January 2012. Prior to LeapFrogRx Chris was CTO and VP of Product Management of MarketSoft (now part of IBM) an innovative Enterprise Marketing Management software. Prior to that, Chris developed Microsoft Passport, the predecessor to Windows Live ID, a distributed authentication system used by 100s of Millions of users today. He was awarded a US Patent for his work on that project. Before joining Microsoft, he led the technical architecture and implementation of Firefly Passport, an early leader in Internet Personalization and Privacy. Microsoft subsequently acquired Firefly. Chris led the development of the first travel-related e-commerce web site at NetMarket. Chris began his career at the Massachusetts Institute of Technology's (MIT) Lincoln Laboratory and NASA Ames Research Center. There he created software and algorithms that provided aircraft arrival optimization assistance to Air Traffic Controllers at several major airports in the United States.

Chris served as a Peace Corps Volunteer Math Teacher in Botswana, Africa. Chris has an M.S. from Columbia University and a B.S. from the University of Wisconsin-Madison. He is an avid cyclist, hiker, reader, and father of two college age children.

**Gil Benghiat** is a Founder and VP of Products at DataKitchen where he is focusing on DataKitchen users and the Agile data practices.

Gil has held various technical and leadership roles at Solid Oak Consulting, HealthEdge, Phreesia, LeapFrogRx (purchased by Model N), Relicore (purchased by Symantec), Phase Forward (IPO and then purchased by Oracle), Netcentric, Sybase (purchased by SAP), and AT&T Bell Laboratories (now Nokia Bell Labs).

Gil's career has been data oriented starting with collecting and displaying network data at AT&T Bell Labs, managing data at Sybase, collecting and cleaning clinical trial data at PhaseForward, integrating pharmaceutical sales data at LeapFrogRx, protecting patient and financial data at Phreesia, processing claims data at HealthEdge, and liberating data at Solid Oak Consulting.

Gil holds an M.S. in Computer Science from Stanford University and a Sc.B. in Applied Mathematics/Biology from Brown University. He completed hiking all 48 of New Hampshire's, 4,000 peaks and is now working on the New England 67, and is the father of one high school and two college age boys.

**Eran Strod** works in marketing at DataKitchen where he writes white papers, case studies and the DataOps blog. Eran was previously Director of Marketing for Atrenne Integrated Solutions (now Celestica) and has held product marketing and systems engineering roles at Curtiss-Wright, Black Duck Software (now Synopsys), Mercury Systems, Motorola Computer Group (now Artesyn), and Freescale Semiconductor (now NXP), where he was a contributing author to the book "Network Processor Design, Issues and Practices."

Eran began his career as a software developer at CSPi working in the field of embedded computing.

Eran holds a B.A. in Computer Science and Psychology from the University of California at Santa Cruz and an M.B.A. from Northeastern University. He is father to two children and enjoys hiking, travel and watching the New England Patriots.

