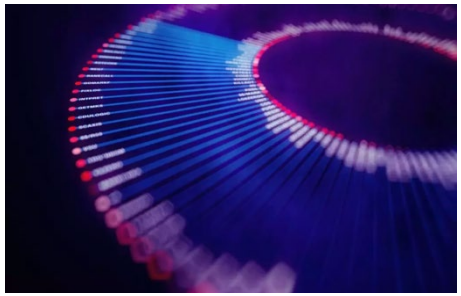
Demystifying Data Observability

An Intellyx Analyst Guide for Unravel Data

By Jason Bloomberg and Jason English

Table of Contents

Foreword

Even the world's most advanced enterprises need to better optimize data-intensive applications for operational resiliency, performance and cost across cloud-based and hybrid IT data stacks.

Complex analytics, machine learning, and data transformation applications operate distinctly differently from typical software applications, and thus demand their own unique type of data observability.

Whether the activity is a specific data analyst's daily queries, or a repeating event emanating from a Kafka stream, every action produces a thread of data operations that can create a splash in several different data lakes, databases and systems of record, making troubleshooting hitches harder.

This Intellyx Analyst Guide will educate IT leaders on the specific requirements and bottlenecks of data-dependent applications that can be addressed by Data Observability.

By Jason English

Partner & Principal Analyst,
Intellyx

Why Do We Need DataOps Observability?

Part 1 of the Demystifying Data Observability Series for Unravel Data

Don’t we already have DevOps?

DevOps was started more than a decade ago as a movement, not a product or solution category.

DevOps offered us a way of collaborating between development and operations teams, using automation and optimization practices to continually accelerate the release of code, measure everything, lower costs, and improve the quality of application delivery to meet customer needs.

DevOps practices also include better management practices such as self-service environments, test and release automation, monitoring, and cost optimization.

On the journey toward DevOps, teams who apply this methodology deliver software more quickly, securely, reliably, and with less burnout.

For dynamic applications to deliver a successful user experience at scale, we still need DevOps to keep delivery flowing. But as organizations increasingly view data as a primary source of business value, data teams are tasked with building and delivering reliable data products and data applications. Just as DevOps principles emerged to enable efficient and reliable delivery of applications by software development teams, DataOps best practices are helping data teams solve a new set of data challenges.

What is DataOps?

If "data is the new oil," as pundits like to say, then data is also the most valuable resource in today's modern data-driven application world.

The combination of commodity hardware, ubiquitous high-bandwidth networking, cloud data warehouses, and infrastructure abstraction methods like containers and Kubernetes creates an exponential rise in our ability to use data itself to dynamically compose functionality such as running analytics and informing machine learning-based inference within applications.

Enterprises recognized data as a valuable asset, welcoming the newly minted CDO (chief data officer) role to the E-suite, with responsibility for data and data quality across the organization. While leading-edge companies like Google, Uber and Apple increased their return on data investment by mastering DataOps, many leaders struggled to staff up with enough data scientists, data analysts, and data engineers to properly capitalize on this trend.

Progressive DataOps companies began to drain data swamps by pouring massive amounts of data (and investment) into a new modern ecosystem of cloud data warehouses and data lakes from open source Hadoop and Kafka clusters to vendor-managed services like Databricks, Snowflake, Amazon EMR, BigQuery, and others.

The elastic capacity and scalability of cloud resources allowed new kinds of structured, semi-structured, and unstructured data to be stored, processed and analyzed, including streaming data for real-time applications.

As these cloud resources quickly grew and scaled, they became a complex tangle of data sources, pipelines, dashboards, and machine learning models, with a variety of interdependencies, owners, stakeholders, and products with SLAs. Getting additional cloud resources and launching new data pipelines was easy, but operating them well required a lot of effort, and making sense of the business value of any specific component to prioritize data engineering efforts became a huge challenge.

Software teams went through the DevOps revolution more than a decade ago, and even before that, there were well-understood software engineering disciplines for design/build/deploy/change, as well as monitoring and observability. Before DataOps, data teams didn't typically think about test and release cycles, or misconfiguration of the underlying infrastructure itself.

Where DevOps optimized the lifecycle of software from coding to release, DataOps is about the flow of data, breaking data out of work silos to collaborate on the movement of data from its inception, to its arrival, processing, and use within modern data architectures to feed production BI and machine learning applications.

DataOps jobs, especially in a cloudy, distributed data estate, aren't the same as DevOps jobs. For instance, if a cloud application becomes unavailable, DevOps teams might need to reboot the server, adjust an API, or restart the K8s cluster.

If a DataOps-led application starts failing, it may show incorrect results instead of simply crashing, and cause leaders informed by faulty analytics and AI inferences to make disastrous business decisions. Figuring out the source of data errors and configuration problems can be maddeningly difficult, and DataOps teams may even need to restore the whole data estate – including values moving through ephemeral containers and pipelines – back to a valid, stable state for that point in time.

Why does DataOps need its own observability?

Once software observability started finding its renaissance within DevOps practices and early microservices architectures five years ago, we also started seeing some data management vendors pivoting to offer 'data observability' solutions.

The original concept of data observability was concerned with database testing, properly modeling, addressing and scaling databases, and optimizing the read/write performance and security of both relational and cloud back ends.

In much the same way that the velocity and automated release cadence of DevOps meant dev and ops teams needed to shift component and integration testing left, data teams need to tackle data application performance and data quality earlier in the DataOps lifecycle.

In essence, DataOps teams are using agile and other methodologies to develop and deliver analytics and machine learning at scale. Therefore they need DataOps observability to clarify the complex inner plumbing of apps, pipelines and clusters handling that moving data. Savvy DataOps teams must monitor ever-increasing numbers of unique data objects moving through data pipelines.

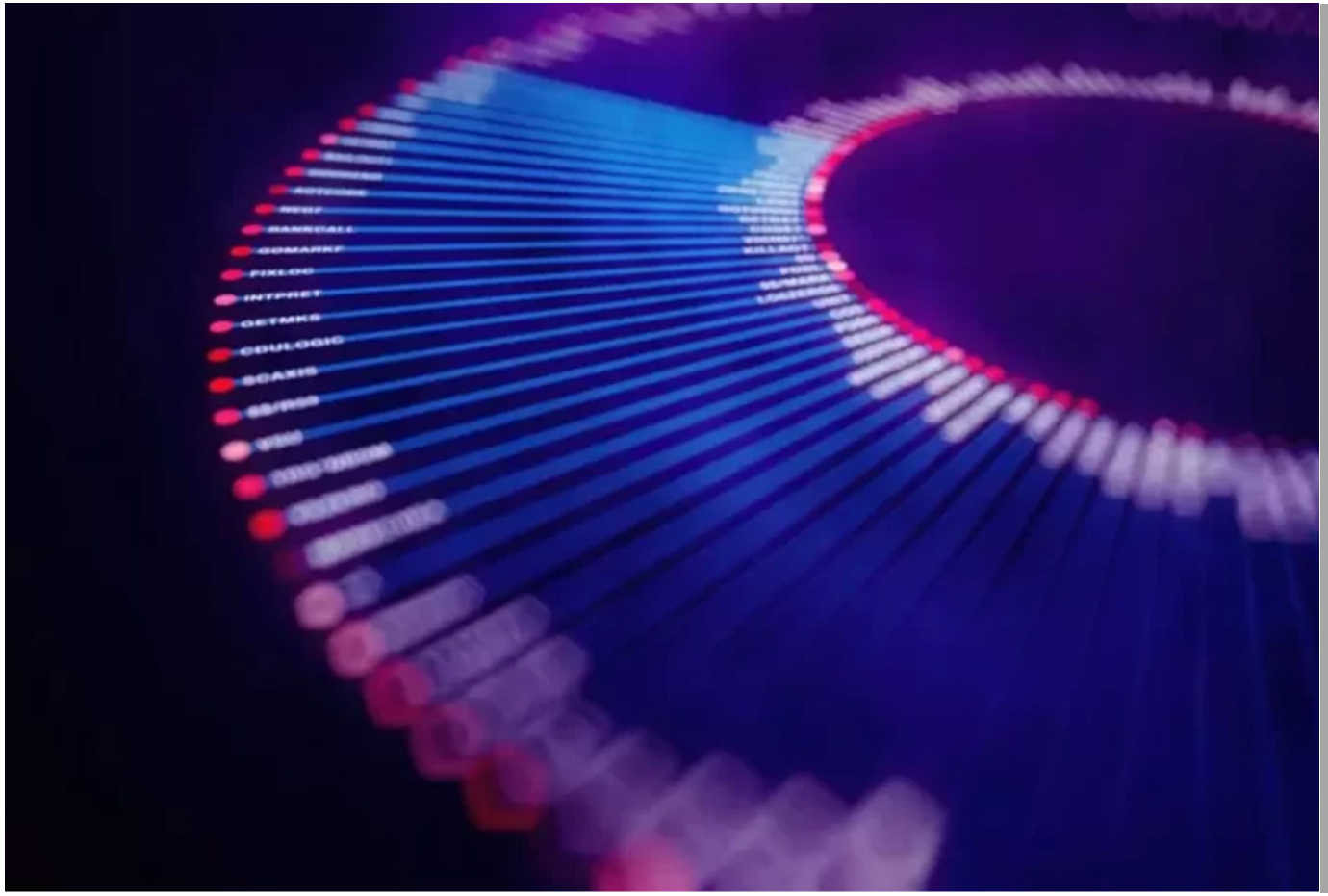
The KPIs for measuring success in DataOps observability include metrics and metadata that standard observability tools would never see: differences or anomalies in data layout, table partitioning, data source lineages, degrees of parallelism, data job and subroutine runtimes and resource utilization, interdependencies and relationships between data sets and cloud infrastructures – and the business tradeoffs between speed, performance and cost (or FinOps) of implementing recommended changes.

The Intellyx Take

A recent survey noted that 97 percent of data engineers 'feel burned out' at their current jobs, and 70 percent say they are likely to quit within a year! That's a wakeup call for why DataOps observability matters now more than ever.

We must maintain the morale of understaffed and overworked data teams, when these experts take a long time to train and are almost impossible to replace in today's tight technical recruiting market.

Any enterprise that intends to deliver modern DataOps should first consider equipping data teams with DataOps observability capabilities. Observability should go beyond the traditional metrics and telemetry of application code and infrastructure, empowering DataOps teams to govern data and the resources used to refine and convert raw data into business value as it flows through their cloud application estates.



By Jason Bloomberg

Managing Partner & Analyst, Intellyx

The Evolution from DevOps to DataOps

Part 2 of the Demystifying Data Observability Series for Unravel Data

DevOps Precursors

The traditional, pre-cloud approach to building custom software in large organizations separated the application development ('dev') teams from the IT operations ('ops') personnel responsible for running software in the corporate production environment.

In between these two teams, organizations would implement a plethora of processes and gates to ensure the quality of the code and that it would work properly in production before handing it to the ops folks to deploy and manage.

Such 'throw it over the wall' processes were slow and laborious, leading to deployment cycles many months long. The importance of having software that worked properly, so the reasoning went, was sufficient reason for such onerous delays.

Then came the Web. And the cloud. And enterprise digital transformation initiatives. All of these macro-trends forced enterprises to rethink their plodding software lifecycles.

Not only were they too slow to deliver increasingly important software capabilities, but business requirements would evolve far too quickly for the deployed software to keep up.

Such pressures led to the rise of agile software methodologies, cloud native computing, and DevOps.

Finding the Essence of DevOps

The original vision of DevOps was to pull together the dev and ops teams to foster greater collaboration, in hopes that software deployment cadences would accelerate while maintaining or improving the quality of the resulting software.

Over time, this 'kumbaya' vision of seamless collaboration itself evolved. Today, we can distill the essence of modern DevOps into these five elements:

- A cultural and organizational shift away from the 'throw it over the wall' mentality to greater collaboration across the software lifecycle
- A well-integrated, comprehensive automation suite that supports CI/CD activities, along with specialists who manage and configure such technologies [*i.e., DevOps engineers]*
- A proactive, shift-left mentality that seeks to represent production behavior declaratively early in the lifecycle for better quality control and rapid deployment
- Full-lifecycle observability that shifts problem resolution to the left via better prediction of problematic behavior and preemptive mitigation of issues in production
- Lean practices to deliver value and improve efficiency throughout the software development lifecycle

Furthermore, DevOps doesn't live in a vacuum. Rather, it is consistent with and supports other modern software best practices, including infrastructure-as-code, GitOps, and the 'cattle not pets' approach to supporting the production environment via metadata representations that drive deployment. Most companies put off modernizing monoliths longer than they should due to fear and uncertainty – but there is always a moment when the business can't afford to not change. Time for our intrepid explorers to swap out the production system with an upgraded one, hopefully without setting
off any traps.

The Evolution of DataOps

Before information technology (IT), organizations had management of information systems (MIS). And before MIS, at the dawn of corporate computing, enterprises implemented data processing (DP).

The mainframes at the heart of enterprise technology as far back as the 1960s were all about processing data – crunching numbers in batch jobs that yielded arcane business results, typically dot-matrix printed on green and white striped paper.

Today, IT covers a vast landscape of technology infrastructure, applications, and hybrid on-premises and cloud environments – but data processing remains at the heart of what IT is all about.

Early in the evolution of DP, it became clear that the technologies necessary for processing transactions were different from the technology the organization required to provide business intelligence to line-of-business (LoB) professionals.

Enterprises required parallel investments in online transaction processing (OLTP) and online analytical processing (OLAP), respectively. OLAP proved the tougher nut to crack, because enterprises generated voluminous quantities of transactional data, while LoB executives required complex insights that would vary over time – thus taxing the ability of the data infrastructure to respond to the business need for information.

Operating these early OLAP systems was relatively straightforward, centering on administering the data warehouses. In contrast, today's data estate – the sum total of all the data infrastructure in a modern enterprise – is far more varied than in the early data warehousing days.

Motivations for DataOps

Operating this data estate has also become increasingly complex, as the practice of DataOps rises in today's organizations.

Complexity, however, is only one motivation for DataOps. There are more reasons why today's data estate requires it:

- Increased mission-criticality of data, as digital transformations rework organizations into digital enterprises
- Increased importance of real-time data, a capability that data warehouses never delivered
- Greater diversity of data-centric use cases beyond basic business intelligence
- Increased need for dynamic applications of data, as different LoBs need an ever-growing variety of data-centric solutions
- Growing need for operational cost predictability, optimization, and governance

Driving these motivations is the rise of AI, as it drives the shift from code-based to data-based software behavior. In other words, AI is more than just another data-centric use case. It repositions data as the central driver of software functionality for the enterprise.

The Intellyx Take

For all these reasons, DataOps can no longer follow the simplistic data warehouse administration pattern of the past. Today's data estate is dynamic, diverse, and increasingly important, requiring organizations to take a full-lifecycle approach to collecting, transforming, storing, querying, managing, and consuming data.

As a result, DataOps requires the application of core DevOps practices along the data lifecycle. DataOps requires the cross-lifecycle collaboration, full-lifecycle automation and observability, and the shift-left mentality that DevOps brings to the table – only now applied to the enterprise data estate.

Thinking of DataOps as 'DevOps for data' may be too simplistic an explanation of the role DataOps should play. Instead, it might be more accurate to say that as data increasingly becomes the driver of software behavior, DataOps becomes the new DevOps.

By Jason English

Partner & Principal Analyst,
Intellyx

DataFinOps: More On The Menu Than Data Cost Convergence

Part 3 of the Demystifying Data Observability Series for Unravel Data

IT and data executives find themselves in a quandary about deciding how to wrangle an exponentially increasing volume of data to support their business requirements – without breaking an increasingly finite IT budget.

Like an overeager diner at a buffet who's already loaded their plate with the cheap carbs of potatoes and noodles before they reach the protein-packed entrees, they need to survey all of the data options on the menu before formulating their plans for this trip.

In our previous chapters of this series, we discussed why DataOps needs its own kind of observability, and then how DataOps is a natural evolution of DevOps practices. Now there's a whole new set of options in the data observability menu to help DataOps teams track the intersection of value and cost.

From ROI to FinOps

Executives can never seem to get their fill of ROI insights from IT projects, so they can measure bottom-line results or increase top-line revenue associated with each budget line item. After all, predictions about ROI can shape the perception of a company for its investors and customers.

Unfortunately, ROI metrics are often discussed at the start of a major technology product or services contract – and then forgotten as soon as the next initiative gets underway.

Some Considerations Frequently Seen On The FinOps Menu Include:

- Based on customer demand or volatility in our consumption patterns, should we buy capacity on-demand or reserve more cloud capacity?
- Which FinOps tools should we buy, and what functionality should we build ourselves, to deliver this important new capability?
- Which cloud cost models are preferred for capital expenditures (capex) projects and operational expenditures (opex)?
- What is the potential risk and cost of known and unknown usage spikes, and how much should we reasonably invest in analysts and tools for preventative purposes?

As a discipline, FinOps has come a long way, building communities of interest among expert practitioners, product, business, and finance teams as well as solution providers through its own FinOps Foundation and instructional courses on the topic.

FinOps + DataOps = DataFinOps?

Real-time analytics and AI-based operational intelligence are enabling revolutionary business capabilities, enterprise-wide awareness, and innovative machine learning-driven services. All of this is possible thanks to a smorgasbord of cloud data storage and processing, cloud data lakes, cloud data warehouse, and cloud lakehouse options.

Unfortunately, the rich streams of data required for such sophisticated functionality bring along the unwanted side effect of elastically expanding budgetary waistbands, due to ungoverned cloud storage and compute consumption costs. Nearly a third of

all data science projects go more than 40% over budget on cloud data, according to a recent survey–a huge delta between cost expectations and reality.

How can better observability into data costs help the organization wring more value from data assets without cutting into results, or risking cost surprises?

As it turns out, data has its own unique costs, benefits, and value considerations. Combining the disciplines of FinOps and DataOps – which I'll dub DataFinOps just for convenience here – can yield a unique new set of efficiencies and benefits for the enterprise's data estate.

Some of the unique considerations of DataFinOps:

1) Which groups within our company are the top spenders on cloud data analytics, and is anything anomalous about their spending patterns versus the expected budgets?
2) What is the value of improving data performance or decreasing the latency of our data estate by region or geography, in order to improve local accuracy, reduce customer and employee attrition and improve retention?
3) If we are moving to a multi-cloud, hybrid approach, what is an appropriate and realistic mix of reserved instances and spot resources for processing data of different service level agreements (SLAs)?
4) Where are we paying excessive ingress / egress fees within our data estate? Would it be more cost effective to process data near the data or move our data elsewhere?
5) How much labor do our teams spend building and maintaining data pipelines, and what is that time worth?
6) Are cloud instances being intelligently right-sized and auto-scaled to meet demand?

Systems-oriented observability platforms such as DataDog and Dynatrace can measure system or service level telemetry, which is useful for a DevOps team looking at application-level cloud capacity and cost/performance ratios. Unfortunately these tools do not dig into enough detail to answer data analytics-specific FinOps questions.

Taming a market of data options

Leading American grocery chain Kroger launched its 84.51° customer experience and data analytics startup to provide predictive data insights and precision marketing for its parent company and other retailers, across multiple cloud data warehouses such as Snowflake and Databricks, using data storage in multiple clouds such as Azure and GCP.

Using the Unravel platform for data observability, they were able to get a grip on data costs and value across multiple data platforms and clouds without having to train up more experts on the gritty details of data job optimization within each system.

> *"The end result is giving us tremendous visibility into what is happening within our platforms. Unravel gave recommendations to us that told us what was good and bad. It simply cut to the chase and told us what we really needed to know about the users and sessions that were problematic. It not only identified them, but then made recommendations that we could test and implement."*
>
> — **Jeff Lambert,** Vice President Engineering, 84.51°

It's still early days for this transformation, but a data cost reduction of up to 50% would go a long way toward extracting value from deep customer analytics, as transaction data volumes continue to increase by 2x or 3x a year as more sources come online.

The Intellyx Take

It would be nice if CFOs could just tell CIOs and CDOs to simply stop consuming and storing so much data, and have that reduce their data spend. But just like in real life, crash diets will never produce long-term results, if the 'all-you-can-eat' data consumption pattern isn't changed.

The hybrid IT underpinnings of advanced data-driven applications evolves almost every day. To achieve sustainable improvements in cost/benefit returns on data, analysts and data scientists would have to become experts on the inner workings of each public cloud and data warehousing vendor.

DataFinOps practices should encourage data team accountability for value improvements, but more importantly, it should give them the data observability, AI-driven recommendations, and governance controls necessary to both contain costs, and stay ahead of the organization's growing business demand for data across hybrid IT data resources and clouds.

By Jason Bloomberg

Managing Partner & Analyst, Intellyx

DataOps Resiliency: Tracking Down Toxic Workloads

Part 4 of the Demystifying Data Observability Series for Unravel Data

In the first three articles in this four-post series, my colleague Jason English and I explored DataOps observability, the connection between DevOps and DataOps, and data-centric FinOps best practices.

In this concluding article in the series, I'll explore DataOps resiliency – not simply how to prevent data-related problems, but also how to recover from them quickly, ideally without impacting the business and its customers.

Observability is essential for any kind of IT resiliency – you can't fix what you can't see – and DataOps is no exception. Failures can occur anywhere in the stack, from the applications on down to the hardware. Understanding the root causes of such failures is the first step to fixing, or ideally preventing, them.

The same sorts of resiliency problems that impact the IT environment at large can certainly impact the data estate. Even so, traditional observability and incident management tools don't address specific problems unique to the world of data processing.

In particular, DataOps resiliency must address the problem of *toxic workloads*.

Understanding Toxic Workloads

Toxic data workloads are as old as relational database management systems (RDBMSs), if not older. Anyone who works with SQL on large databases knows there are some queries that will cause the RDBMS to slow dramatically or completely grind to a halt.

The simplest example: SELECT * FROM TRANSACTIONS where the TRANSACTIONS table has millions of rows. Oops! Your resultset also has millions of rows!

JOINs, of course, are more problematic, because they are difficult to construct, and it's even more difficult to predict their behavior in databases with complex structures.

Such toxic workloads caused problems in the days of single on-premises databases. As organizations implemented data warehouses, the risks compounded, requiring increasing expertise from a scarce cadre of query-building experts.

Today we have data lakes as well as data warehouses, often running in the cloud where the meter is running all the time. Organizations also leverage streaming data, as well as complex data pipelines that mix different types of data in real time.

With all this innovation and complexity, the toxic workload problem hasn't gone away. In fact, it has gotten worse, as the nuances of such workloads have expanded.

Breaking Down the Toxic Workload

Poorly constructed queries are only one of the causes of a modern toxic workload. Other root causes include:

- *Poor quality data* – one table with NULL values, for example, can throw a wrench into seemingly simple queries. Expand that problem to other problematic data types and values across various cloud-based data services and streaming data sources, and small data quality problems can easily explode into big ones.
- *Coding issues* – Data engineers must create data pipelines following traditional coding practices – and whenever there's coding, there are software bugs. In the data warehouse days, tracking down toxic workloads usually revealed problematic queries. Today, coding issues are just as likely to be the root cause.
- *Infrastructure issues* – Tracking down the root causes of toxic workloads means looking everywhere – including middleware, container infrastructure, networks, hypervisors, operating systems, and even the hardware. Just because a workload runs too slow doesn't mean it's a data issue. You have to eliminate as many possible root causes as you can – and quickly.
- *Human issues* – Human error may be the root cause of any of the issues above – but there is more to this story. In many cases, root causes of toxic workloads boil down to a shortage of appropriate skills among the data team or a lack of effective collaboration within the team. Human error will always crop up on occasion, but a skills or collaboration issue will potentially cause many toxic workloads over time.

The bottom line: DataOps resiliency includes traditional resiliency challenges but extends to data-centric issues that require data observability to address.

Data Resiliency at Mastercard

Mastercard recently addressed its toxic workload problem on Hadoop, as well as Impala, Spark, and Hive.

The payment processor has petabytes of data across hundreds of nodes, as well as thousands of users who access the data in an ad hoc fashion – that is, they build their own queries.

Mastercard's primary issue was poorly constructed queries, a combination of users' inexperience as well as the complexity of the required queries.

In addition, the company faced various infrastructure issues, from overburdened data pipelines to maxed-out storage and disabled daemons.

All these problems led to application failures, system slowdowns and crashes, and resource bottlenecks of various types.

To address these issues, Mastercard brought in Unravel Data. Unravel quickly identified hundreds of unused data tables. Freeing up the associated resources improved query performance dramatically.

Mastercard also uses Unravel to help users tune their own query workloads as well as automate the monitoring of toxic workloads in progress, preventing the most dangerous ones from running in the first place.

Overall, Unravel helped Mastercard improve its mean time to recover (MTTR) – the best indicator of DataOps Resiliency.

The Intellyx Take

The biggest mistake an organization can make around DataOps observability and resiliency is to assume these topics are special cases of the broader discussion of IT observability and resiliency.

In truth, the areas overlap – after all, infrastructure issues are often the root causes of data-related problems – but without the particular focus on DataOps, many problems would fall through the cracks.

The need for this focus is why tools like Unravel's are so important. Unravel adds AI optimization and automated governance to its core data observability capabilities, helping organizations optimize the cost, performance, and quality of their data estates.

DataOps resiliency is one of the important benefits of Unravel's approach – not in isolation, but within the overall context for resiliency that is so essential to modern IT.

About Jason Bloomberg

Jason Bloomberg is founder and managing partner of enterprise IT industry analysis firm Intellyx. He is a leading IT industry analyst, author, keynote speaker, and globally recognized expert on multiple disruptive trends in enterprise technology and digital transformation.

He is #13 on the Top 50 Global Thought Leaders on Cloud Computing 2023 and #10 on the Top 50 Global Thought Leaders on Mobility 2023, both by Thinkers 360. He is a leading social amplifier in Onalytica's Who's Who in Cloud? for 2022 and a Top 50 Agile Leaders of 2022 by Team leadersHum.

Mr. Bloomberg is the author or coauthor of five books, including Low-Code for Dummies, published in October 2019.

About Jason 'JE' English

Jason "JE" English is Partner & Principal Analyst at Intellyx. Drawing on expertise in designing, marketing and selling enterprise software and services, he is focused on covering how agile collaboration between customers, partners and employees accelerates innovation.

A writer and community builder with more than 25 years of experience in software dev/test, cloud computing, security, blockchain and supply chain companies, JE led marketing efforts for the development testing and virtualization software company ITKO from its bootstrap startup days, through a successful acquisition by CA in 2011. He co-authored the book *Service Virtualization: Reality is Overrated* to capture the then-novel practice of test environment simulation for Agile development. Follow him on Twitter at @bluefug.

About Intellyx

Intellyx is the first and only industry analysis, advisory, and training firm focused on customer-driven, technology-empowered digital transformation for the enterprise. Covering every angle of enterprise IT from mainframes to cloud, process automation to artificial intelligence, our broad focus across technologies allows business executives and IT professionals to connect the dots on disruptive trends. Read and learn more at https://intellyx.com or follow them on Twitter at @intellyx.

About Unravel Data

Unravel Data radically transforms the way businesses understand and optimize the performance and cost of their modern data applications – and the complex data pipelines that power those applications. Providing a unified view across the entire data stack, Unravel's market-leading Data Observability platform leverages AI, machine learning, and advanced analytics to provide modern data teams with the actionable recommendations they need to turn data into insights. Some of the world's most recognized brands like Adobe, 84.51 (a Kroger company), and Deutsche Bank rely on Unravel Data to unlock data-driven insights and deliver new innovations to market. To learn more, visit www.unraveldata.com.