Eckerson Group

The Ultimate Guide to DataOps

Product Evaluation and Selection Criteria

By Wayne W. Eckerson

October 2019

Research sponsored by Tamr

Tamr
DATA UNITED

About the Author

Wayne W. Eckerson has been a thought leader in the data and analytics field since the early 1990s. He is a sought-after consultant, noted speaker, and expert educator who thinks critically, writes clearly, and presents persuasively about complex topics. Eckerson has conducted many groundbreaking research studies, chaired numerous conferences, written two widely read books on performance dashboards and analytics, and consulted on BI, analytics, and data management topics for numerous organizations. Eckerson is the founder and principal consultant of Eckerson Group.

About Eckerson Group

Eckerson Group helps organizations get more value from data and analytics. Our experts each have more than 25+ years of experience in the field. Data and analytics is all we do, and we're good at it! Our goal is to provide organizations with a cocoon of support on their data journeys. We do this through online content (thought leadership), expert onsite assistance (full-service consulting), and 30+ courses on data and analytics topics (educational workshops).

Full-Service Consulting
Educational Workshops
Thought Leadership

Get More Value From Your Data

Get more value from your data. Put an expert on your side. Learn what Eckerson Group can do for you!

Table of Contents

Executive Summary

DataOps applies rigor to the development and execution of data pipelines. Borrowing principles from DevOps, agile, lean, and total quality management, DataOps helps data development evolve from an artisanal craft subject to delays and errors into an industrial process that accelerates delivery and improves data quality. Ultimately, DataOps helps data teams satisfy their internal customers: it fosters greater self-service while fulfilling the promise of "faster, better, cheaper."

This report explains why data teams should embrace DataOps and then drills into the technology and tools required to build robust, automated data pipelines. It presents a framework for understanding DataOps processes and associated technologies and then describes five categories of DataOps tools. It finishes with a list of criteria for evaluating all-in-one DataOps products.

Key Takeaways

- Most of the programs that data pipeline developers build fall into four categories: big data, data science, self-service analytics, and data warehousing.
- Tests accelerate development and minimize operational delays. Finding issues before internal customers do is critically important for a development team.
- DataOps tools foster collaboration that is critical to scale development, increase capacity and output, reduce errors, and accelerate time to market.
- Data teams need to create "design patterns" that define the components they will use when creating data pipelines.
- There are five categories of DataOps products on the market today: 1) all-in-one tools, 2) orchestration tools, 3) component tools, 4) case-specific tools, and 5) open source tools

Recommendations

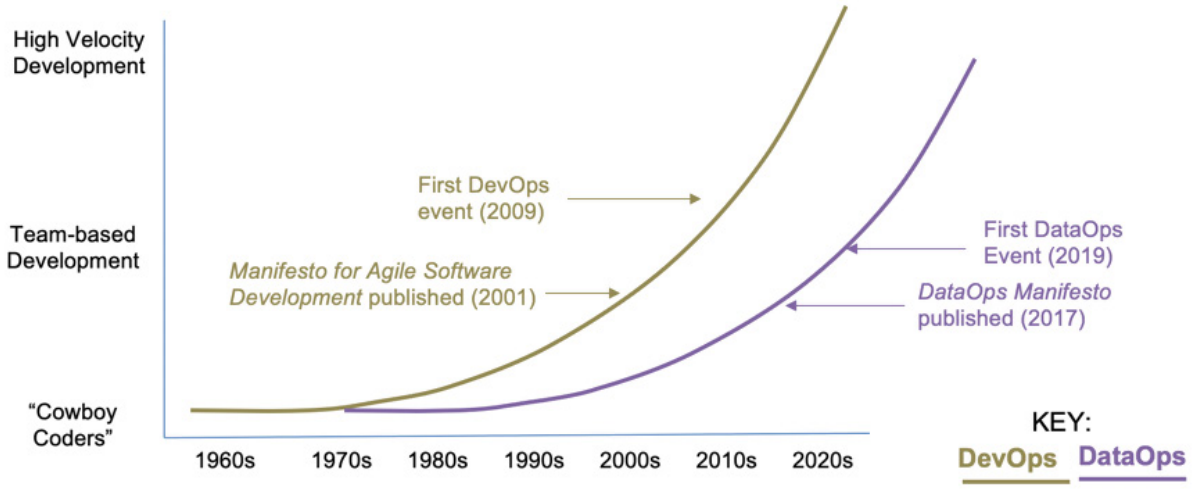
- Testing is the key to DataOps; make sure you deploy a test automation framework and are diligent about creating and running tests at every point in the development and production lifecycle.
- Devote 50% of your code and staff to testing, quality, and development velocity.
- Use Eckerson Group's DataOps framework and evaluation criteria to guide your selection of DataOps tools.
- Remember that DataOps is more than tools; it's a combination of people, processes, and technologies required to build robust, automated data pipelines.

Why DataOps?

DataOps, short for "data operations," brings rigor to the development and management of data pipelines. It promises to turn the creation of analytic solutions from an artisanal undertaking by a handful of developers and analysts to an industrial operation that builds and maintains hundreds or thousands of data pipelines. DataOps not only increases the scale of data analytics development, but it accelerates delivery while improving quality and staff productivity. In short, it creates an environment where "faster, better, cheaper" is the norm, not the exception.

Following DevOps. DataOps trails its DevOps brethren by a decade or so. (See figure 1.) DevOps applies rigor to software engineering, enabling large teams of software developers to deliver continuous releases of high-quality code. New digital startups that apply DevOps have been known to deploy multiple new releases a day. DevOps teams achieve this acceleration by devoting 50% of their code to tests and 50% of their staff to data quality, security, and deployment.

Figure 1. The Trajectory of DevOps and DataOps



Data analytics has yet to devote the same amount of attention to testing, quality, and delivery cycles as software engineering. At most, data analytics teams devote 20% of their code to testing and have one or two quality assurance engineers on staff, who largely perform unit, systems, and integration tests during development but rarely once code has been deployed into production. But this is changing. (See Best Practices in DataOps: How to Create Robust, Automated Data Pipelines, Eckerson Group report, June 2019.)

At most, data analytics teams devote 20% of their code to testing and have one or two quality assurance engineers on staff.

Testing = Speed. Harvinder Atwal, head of data strategy and advanced analytics at MoneySuperMarket, a British price comparison website, says, “A car needs brakes to go fast.” In the world of DataOps, tests are the brakes that developers create when building code. Those tests are applied not just in unit and integration tests during the development phase, but also during production to ensure that data drift hasn’t altered the accuracy of analytic output, and that changes to software configurations and data schema don’t break production jobs.

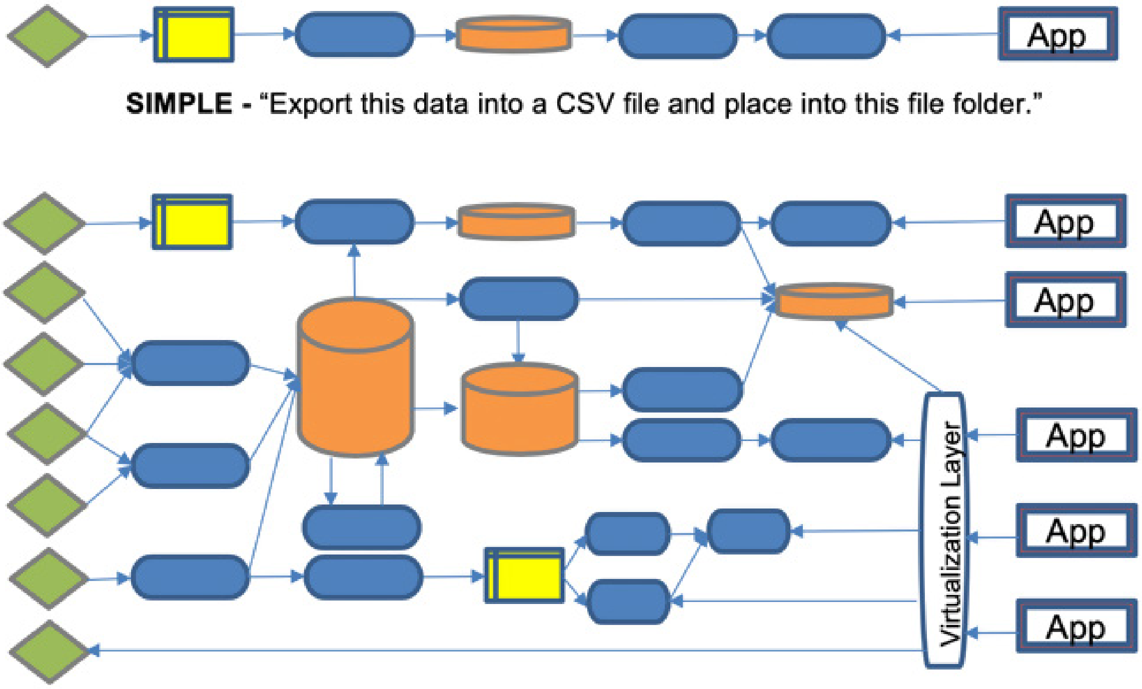
Transforming Cowboy Coders. To the dismay of cowboy coders, DataOps applies overhead to the development process. Developers have to check in code, run it through tests, fix problems, and redeploy. They have to work with new tools and collaborate more with other developers. Consequently, cowboy coders often resist DataOps. Most have worked independently without much structure, process, or controls. They say DataOps will “slow us down” and that the new regimen is better suited for software development than data development. But after they adjust to the new regimen, they never look back.

“Our data architects now love DataOps because it provides a framework to deploy code without worrying about breaking things in production,” says Shakeeb Akhter, former director of enterprise data warehousing at Northwestern Medicine in Chicago. “They now say they can’t believe we lived without this process. And it frees them up to tackle other things, such as predictive analytics, non-relational data, and the cloud.”

Data Pipelines

A data pipeline represents the encoded flow of data from source to consumption. Some data pipelines are simple: “Export this data into a CSV file and place into this folder.” But others are complex: “Move tables from 10 sources into a target database, merge common fields, array into a dimensional schema, aggregate by year, flag null values, convert into an extract for a BI tool, and generate personalized dashboards based on the data.” (See figure 2.)

Figure 2. Types of Data Pipelines



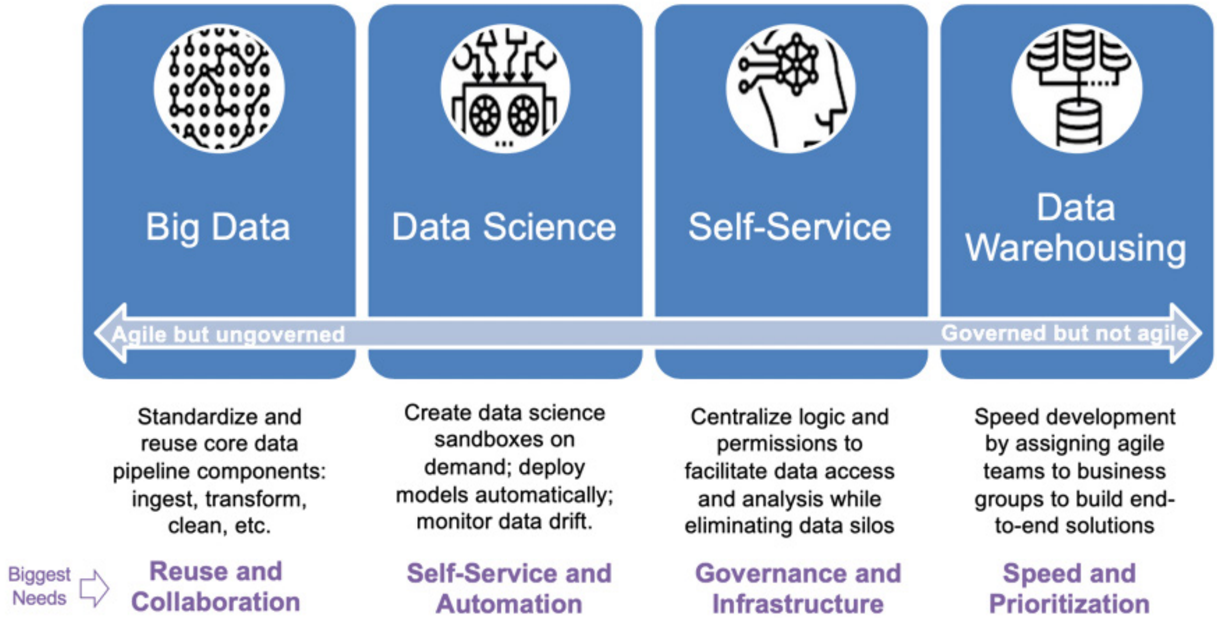
SIMPLE - "Export this data into a CSV file and place into this file folder."

COMPLEX - "Move tables from 10 sources into a target database, merge common fields, array into a dimensional schema, aggregate by year, flag null values, convert into an extract for a BI tool, and generate personalized dashboards based on the data."

The term data pipeline emerged from the big data community. Their data lakes created a feeding frenzy among developers, data scientists, and data analysts who gained unfettered access for the first time to a nearly limitless supply of raw data sourced from a multiplicity of internal and external systems. These individuals created a wide diversity of workflows and programs—or data pipelines—to process that data (e.g., clean, filter, aggregate, move, load) for a variety of purposes and use cases.

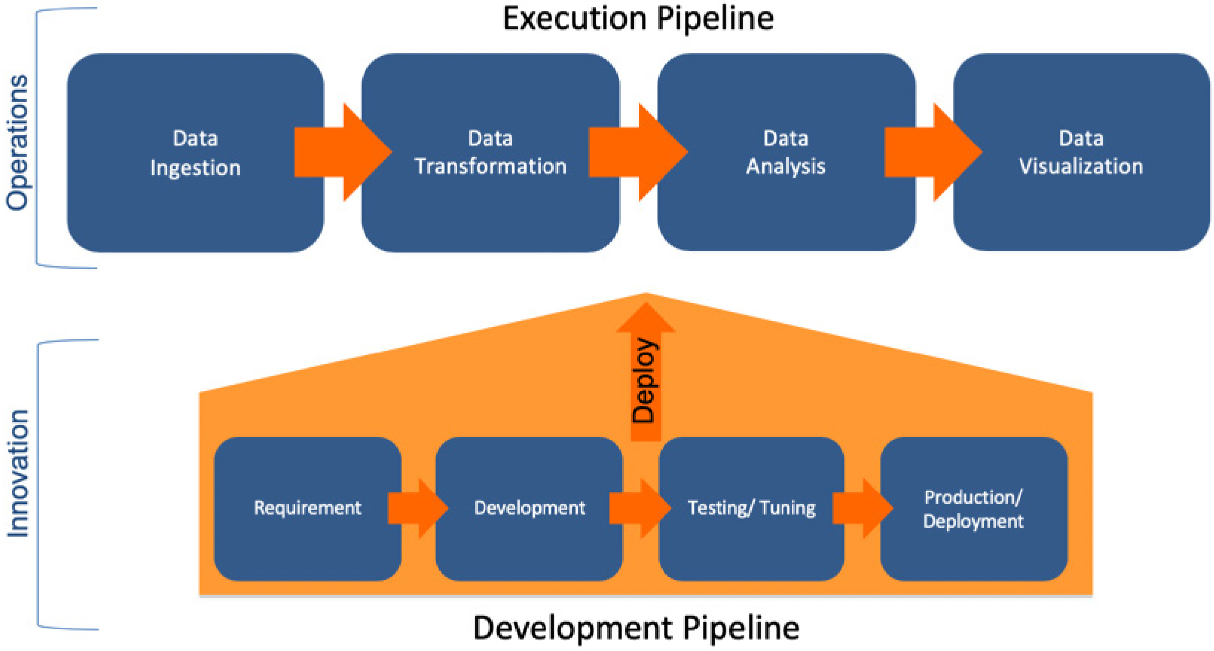
Use Cases. Most of the programs that data developers build fall into one of four categories: big data, data science, self-service analytics, and data warehousing. Big data involves exploring semi-structured data (e.g., social media, clickstream, or sensor data), while self-service focuses on structured data. Data science often uses the output of data exploration to create machine learning models, while data warehousing supports reporting and lightweight analysis. (See figure 3.)

Figure 3. Data Pipeline Use Cases


Big Data
Data Science
Self-Service
Data Warehousing
Agile but ungoverned
Governed but not agile
Standardize and reuse core data pipeline components: ingest, transform, clean, etc.
Create data science sandboxes on demand; deploy models automatically; monitor data drift.
Centralize logic and permissions to facilitate data access and analysis while eliminating data silos
Speed development by assigning agile teams to business groups to build end-to-end solutions
Biggest Needs
Reuse and Collaboration
Self-Service and Automation
Governance and Infrastructure
Speed and Prioritization

Types of Data Pipelines. There are two types of data pipelines: a development pipeline creates code to process data for a new pipeline; an execution pipeline runs the pipeline in production. In most IT shops, these are managed by different teams, creating tremendous inefficiencies: developers can build or change code without being responsible for its downstream impacts. This inevitably increases error rates, delays delivery and frustrates business users. DataOps addresses the disconnect between development and operation teams, ensuring that new or altered code executes the first time without bugs. (See figure 4.)

Figure 4. Development and Operations Pipelines



Testing

Tests are central to both types of pipelines. In development pipelines, the data is fixed but code is variable, while in production the data is variable and code is fixed. So, development tests the code, and production tests the data. Teams must build tests for both types of pipelines and run them continuously in both development and production to ensure that changes don't adversely affect outcomes.

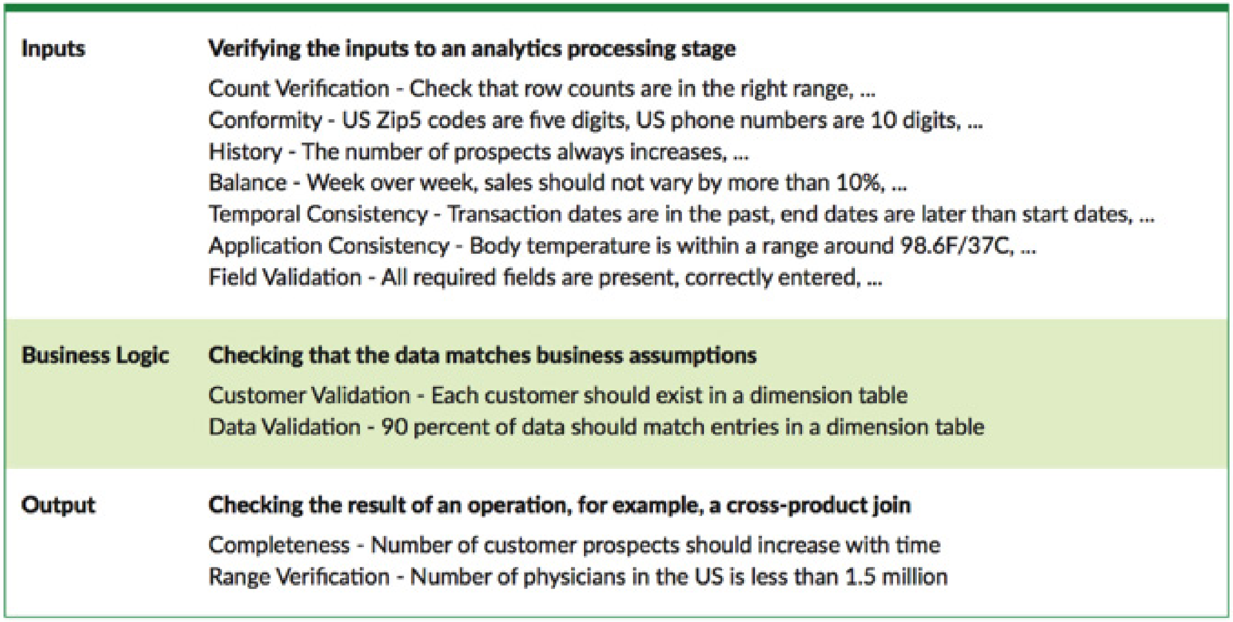
Tests accelerate development and minimize operational delays. Unfortunately, too many development shops make changes, perform minimal tests, and then cross their fingers that the changes don't break something. As one DataOps guru says, "Hope is not a strategy; it is your enemy."

Intel has more than 1,000 tests in its test automation framework, and it keeps adding tests all the time.

Intel has more than 1,000 tests in its test automation framework, and it keeps adding tests all the time. It continually measures individual and group progress over time. "Test automation is a huge part of what we do," says Greg Martinez, former enterprise analytics engineer manager at Intel. "Without it, we can't maintain high quality at the scale and speed with which we operate. We are only as good as our test practices, and we strive to improve here."

Types of Tests. Finding issues before internal customers do is critically important for a development team. There are three types of tests. Data input tests prevent bad data from entering a new node in a pipeline stage; business logic tests validate that data matches business assumptions; and *data output tests* verify that a pipeline stage executed properly. Good DataOps tools save test results so administrators can monitor quality against historical levels. (See figure 5.)

Figure 5. Types of Tests



Inputs	**Verifying the inputs to an analytics processing stage** 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, ...
Business Logic	**Checking that the data matches business assumptions** Customer Validation - Each customer should exist in a dimension table Data Validation - 90 percent of data should match entries in a dimension table
Output	**Checking the result of an operation, for example, a cross-product join** Completeness - Number of customer prospects should increase with time Range Verification - Number of physicians in the US is less than 1.5 million

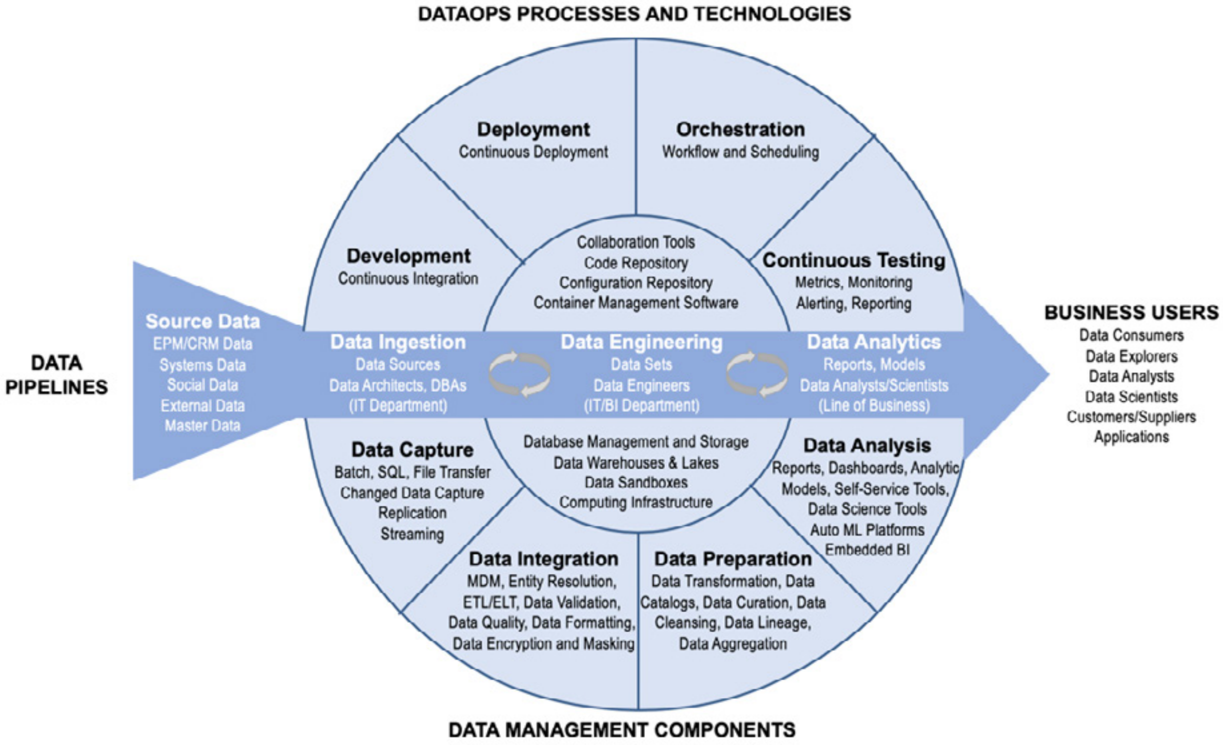
From "Build Trust through Test Automation and Monitoring," DataKitchen, September 6, 2018.

It's important that some of these tests pass the business "sniff test." Business users can glance at a report or data set and know immediately if the data is right or that visual layouts or aids they rely upon have changed. For example, they know their top customers, products, and regions by heart and notice any deviation. Sometimes this attentiveness backfires, such as when a development team fixes a long-standing calculation error and delivers correct data that the business users insist is wrong. Fixing such errors requires a large dollop of diplomacy!

DataOps Technology

DataOps tools foster collaboration that is critical to scale development teams and increase their capacity. DataOps is often associated most strongly with a portfolio of DevOps tools that help large development teams work together more efficiently and effectively. Figure 6 depicts a DataOps framework with tools and technologies.

Figure 6. DataOps Technical Framework



Data Pipeline Components. The blue arrow represents a typical data pipeline with its core components: data ingestion, data engineering, and data analytics. The pipeline processes source data for consumption by a range of business users and applications. A data pipeline is often iterative because analyzing data causes developers to rethink their processing logic or acquire additional data.

DataOps Processes

Atop the data pipeline in figure 6 are core DataOps processes and associated technologies.

Continuous Integration/Continuous Delivery. Developers use continuous integration (CI) and continuous delivery (CD) tools to accelerate the creation of code that generates trustworthy data (i.e., passes all the tests). CI tools let large teams of developers work on branches of the same code set without creating coding conflicts. CD tools push code into production, testing system and software configurations to ensure error-free deployments. Both CI and CD tools, which often work hand-in-hand or as part of the same software suite, pull source code from code repositories, such as GitHub, that provide check-in/out and version control features.

Orchestration. Orchestration software coordinates the interaction of software that processes data in a pipeline. Essentially, it schedules and coordinates the flow of data among data pipeline components. With orchestration software, engineers define workflows, manage dependencies, and schedule jobs, storing the metadata in a repository. Orchestration software can also activate and run tests at each node in a pipeline and record results to maintain a continuous testing and monitoring environment.

Testing. As mentioned, continuous testing is another important component of the DataOps technology stack. Although developers can build tests using SQL or other scripting tools, many DataOps teams rely on test automation tools to provide a universal framework for evaluating code and data quality on a continuous basis. Testing is an integral component of continuous integration and continuous delivery processes. But it's also built into execution pipelines to detect any and all errors that might affect the user experience.

Monitoring. An integral part of continuous testing is performance management. Rather than test code and data quality, performance management tools monitor underlying systems and their impact on business users and applications. Using machine learning, they can automatically pinpoint the root cause of performance issues and outages, alert administrators to issues, and recommend actions to ensure compliance with service level agreements.

Platform Components. Most DataOps shops rely on platform products to manage DataOps processes. Among these are the following:

- Agile collaboration software (e.g., Jira) to manage user stories and sprints;
- A code repository (e.g., GitHub, mentioned above) for storing and versioning source code;
- A configuration repository for tracking server and operating system releases and code libraries used in development and delivery processes; and
- Container software to accelerate the deployment of components from development to test and production.

Data Management Components

While DataOps components manage the process of creating data pipelines, data management components manipulate the data running through the pipelines. Data management tools have been around for decades, although the technologies they employ continue to evolve rapidly.

Data Capture. The majority of data pipelines capture data in batch, using SQL queries or extract routines, but this is changing. Many companies now prefer a streaming-first architecture that captures and processes data in real time in a continuous stream. Streaming tools use change data capture, replication, and/or publish/subscribe messaging systems to capture events as they happen, move them into the data environment, and analyze them.

Data Integration. The heavy work begins once raw data lands in the staging area or data lake (or continuously streams through a real-time processing environment). Companies use a variety of tools to physically and semantically integrate data. Master data management and entity resolution tools harmonize or standardize master data (e.g., customer, product, location records) across source systems; extract, transform and load (ETL) tools and their ELT counterparts transform data into canonical models used largely within data warehouses and data lakes; data validation, data quality, and data formatting tools check data integrity and clean and standardize data; and data encryption, masking, and security software protects data from unauthorized access and usage.

Data Preparation. Data preparation turns raw or lightly integrated, subject-oriented data into business-ready data sets that business users can analyze for decision-making. Traditionally, business users have used Excel to acquire, manipulate, and mash up data. But today, data catalogs enable users to find relevant data sets (curated by the IT department or data stewards), and data preparation tools make it easy to manipulate and combine data for analysis. The best tools run on a corporate server or the cloud, creating a collaborative platform so data analysts can share ideas and output, and promoting code reuse and the documentation of tribal knowledge.

Data Analytics. The typical end of the line for data pipelines are business-facing analytic tools and applications, such as reports, dashboards, analytic models, and self-service visualizations. Increasingly, companies are embedding analytics into other applications, closing the loop between operations and analytics. They are also putting data science tools into the hands of data analysts (aka citizen data scientists) to create first-pass predictive models or simple models that don't require a data scientist to review.

Data Platform. Data management tools require a robust data and computing infrastructure that scales to meet business demand without impacting performance. Most companies now build a data lake to store raw data of any type from any system, often in Hadoop but increasingly in a public cloud (e.g., Microsoft Azure). They also build data warehouses to support performance reporting (e.g., reports, dashboards, and light analysis) on structured data and temporary sandboxes for testing, analysis, and prototyping purposes. These data environments require fast query processing, large storage capacity, elastic computing, in-memory caches, massively parallel query processing, columnar storage, and large in-memory data grids and computing clusters.

Design Patterns

Given the plethora of available tools, it's critical that data teams create "design patterns" to define the components individuals should use to create data pipelines. Without a data pipeline architecture, different groups will likely select unique components for each stage in pipeline development (i.e., ingestion, transformation, analytics), leading to fragmentation, overhead, and redundancy. Well-defined design patterns promote standardization, reuse, and economies of scale.

For example, Intel has 30 development teams working in a petabyte-scale Hadoop environment that pulls data from more than 150 sources. To operate efficiently at that scale, the team has numerous standardized data constructs and components that developers can reuse or tweak to accelerate the development of new data pipelines. “We create reusable design patterns that enable us to create a data pipeline quickly, change it as needed, and maintain reliability and consistency of the data output,” says Intel’s Martinez.

DataOps Tools

There are five categories of DataOps products on the market today:

- All-in-one tools
- Orchestration tools
- Component tools
- Case-specific tools
- Open source tools

All-in-One Tools. These products bundle most of the components necessary to build, test, monitor, and deploy data pipelines in a single integrated GUI-based environment. Although these products claim to support all DataOps processes and data management functions and platform capabilities (i.e., repositories, data lakes, containers), few do. But most are startups and making progress toward delivering a complete environment.

For example, StreamSets focuses heavily on ingestion and populating a big data lake with Apache Spark. Although it provides a graphical interface so data scientists can easily build their own data pipelines without IT involvement. Infoworks also provides a GUI-based environment for building and running data pipelines, but it goes a step further and integrates its output with various BI and analytic tools to simplify data access, security, and analysis.

All-in-one tools are ideal for organizations that want to standardize on a single integrated platform to build, run, and monitor data pipelines as well as port them to new environments (e.g., the cloud). These tools simplify management, accelerate adoption, and reduce costs. They also provide “one throat to choke” in case something goes awry. The downside of these tools, besides their relative immaturity, is that they may not have everything a customer needs or wants. The best products are extensible, making it easy for customers to plug in third party tools to supplement the platform or build extensions to deliver new functionality.

Orchestration Tools. Rather than try to be all things to all customers, orchestration tools focus solely on DataOps processes. These tools wrap around an organization’s existing data management products with a DataOps overlay that provides continuous testing and monitoring to ensure high quality and fast cycle times. Customers wrap the engines of existing products in containers so administrators can easily schedule, run, and test new code against old data (i.e., development environment) or old code against new data (i.e., execution environment) or create new development, test, and production environments in an automated manner.

Orchestration tools are great for companies that have invested large amounts of money and time into data management tools and don’t want to introduce another tool. The orchestration software applies continuous integration, continuous delivery, and continuous testing and monitoring to existing components, helping companies improve quality and cycle times without changing core development tools. For example, DataKitchen reduces analytics cycle time by monitoring data quality and providing automated support for the deployment of data and analytics.

Component Tools. Several dozen components are required to create, execute, and manage a data pipeline, and there are individual products for each component. DataKitchen has published an article that tracks component DataOps products, which now total 62 products in 10 categories. For instance, an organization could buy separate tools for continuous integration, continuous delivery, configuration management, performance management, and so on.

For example, Northwestern Medicine has a number of DataOps tools to foster collaboration and automation. It uses GitHub as a source control repository for data integration code; Jira to coordinate Scrum processes and manage user stories; TeamCity to facilitate code integration in a team-based development environment; and Octopus to deploy code from test into production.

One interesting component product is Unravel, which offers a performance management and monitoring tool that uses machine learning to automatically troubleshoot performance issues afflicting business applications and automatically recommends or executes fixes to comply with SLAs. Rather than assign expensive Spark developers to troubleshoot performance issues, Unravel continuously monitors the environment, detects problems, and informs administrators about the source of the issue and potential fixes.

Another component tool is Tamr, which provides data unification software that semantically integrates disparate data. It will scan multiple databases and match like records, such as customer, product, or supplier, to create a unified view of core data elements and master data. Companies can then use these views to create a canonical model for standardizing data sets, if they choose.

Case-Specific Tools. This category of DataOps tools is designed to support a specific “domain” of DataOps, such as data science (AiOps), data warehousing (DW automation), cloud migration (CloudOps), and so on. For instance, Attunity Compose (now owned by Qlik) is a metadata-driven data warehouse automation tool designed to automate updates to data marts and data warehouses. Also, Seldon is an AiOps tool that streamlines the data science workflow, with audit trails, advanced experiments, continuous integration, and deployment.

Open Source. Finally, there is a plethora of open source DataOps tools, many of which are quite popular, especially in the DevOps world. For example, Jenkins is a leading CI/CD tool and GitHub is the leading source code repository. Also, Apache Airflow is one of the most popular data orchestration tools on the market.

Evaluation Criteria

You should evaluate the following key criteria when selecting a DataOps tool. These criteria are best suited to evaluating all-in-one DataOps tools (see above.)

1. **Comprehensive.** Provides end-to-end creation, execution, and management of data pipelines from source to consumption.
2. **Self-Service.** Enables business users to create data pipelines without IT assistance and without deep training in data engineering, SQL, or production processes.
3. **Connected.** Can connect to any data source or application and ingest data for one or many data pipelines.
4. **Automated.** Can automatically move data and code from development to test and production environments with minimal conflicts or errors.
5. **Controllable.** Supports version control, release management, and rollback to preserve the integrity of data pipelines.
6. **Instrumented.** Applies tests to code and data to detect anomalies, errors, and drift.
7. **Intelligent.** Applies machine learning to detect semantic relationships among data elements, tables, and sources and automatically creates schema and join paths to facilitate queries.
8. **Graphical.** Provides a graphical interface for creating data pipelines with exits to write script or code, if preferred.
9. **Flexible.** Enables users to spawn new data processing environments on demand for development, test, prototyping, or analysis.
10. **Collaborative.** Allows users to view the work of others and share ideas and comments about those artifacts.
11. **Reusable.** Makes it easy for users to search for and reuse existing components and outputs rather than create new data pipelines from scratch.
12. **Secure.** Prevents users from viewing data they are unauthorized to view via encryption, masking, or data access controls.
13. **Portable.** Enables administrators to move all or part of a data pipeline to another data processing platform (i.e., on premises to cloud, cloud to on premises,

cloud to cloud) with no changes and minimal impact on system throughput and performance.

14. **Multi-Modal.** Supports multiple modes of operation, such as batch and stream-based ingestion and in-memory versus persistent data storage.

15. **Elastic.** Scales to handle peak processing without interruption to redirect or reallocate computing resources.

16. **Managed.** Enables administrators to monitor and manage the data environment, including the servers, networks, and core engines, to detect and fix performance issues before they affect business users.

17. **Auditable.** Logs usage and activity, creating easy-to-consume reports and dashboards about all activity on the platform.

18. **Traceable.** Tracks the lineage of all data objects and code, showing upstream sources and downstream consumption.

Conclusion

Although DataOps is rich in technology, it's the right combination of tools, processes, and people that make DataOps a compelling proposition for any data team that wants to do more with less and deploy code with confidence. DataOps ultimately is about creating a culture of continuous improvement. Data teams need to identify bottlenecks and ruthlessly attack them using tools, reengineered processes, and enlightened people.

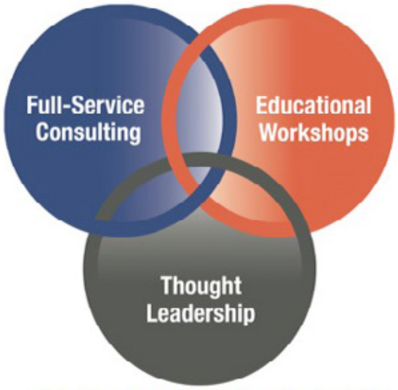
Rather than deploy code with hope and prayer, data teams that apply DataOps tools and techniques can sleep well at night, knowing that a full battery of tests protects their data pipelines and keeps data flowing continuously. Although moving from artisanal to industrial processes is not easy, it is the only path forward for data teams that want to succeed in the 21st century.

About Eckerson Group

Wayne Eckerson, a globally known author, speaker, and advisor, formed Eckerson Group to help organizations get more value from data and analytics. His goal is to provide organizations with a cocoon of support during every step of their data journeys.

Today, Eckerson Group helps organizations in three ways:

- **Our thought leaders** publish practical, compelling content that keeps you abreast of the latest trends, techniques, and tools in the data analytics field.
- **Our consultants** listen carefully, think deeply, and craft tailored solutions that translate your business requirements into compelling strategies and solutions.
- **Our educators** share best practices in more than **30 onsite workshops** that align your team around industry frameworks.



Get More Value From Your Data

Unlike other firms, Eckerson Group focuses solely on data analytics. Our experts each have more than 25 years of experience in the field. They specialize in every facet of data analytics—from data architecture and data governance to business intelligence and artificial intelligence. Their primary mission is to help you get more value from data and analytics.

Our clients say we are hard-working, insightful, and humble. We take the compliment! It all stems from our love of data and desire to help you get more value from data. Put an expert on your side.

Learn what Eckerson Group can do for you!

About Tamr

Tamr is the enterprise-scale data unification company trusted by industry leaders like GE, Toyota, Thomson Reuters, and GSK. The company's patented software platform uses machine learning supplemented with customers' knowledge to unify and prepare data across myriad silos to deliver previously unavailable business changing insights. With a co-founding team led by Andy Palmer (founding CEO of Vertica) and Mike Stonebraker (Turing Award winner) and backed by founding investors NEA and GV, Tamr is transforming how companies get value from their data. To find out more or register for a demo visit tamr.com.