Eckerson Group | White Paper

# Architecting and Automating Data Pipelines

## A Guide to Efficient Data Engineering for BI and Data Science

BY DAVE WELLS

RESEARCH SPONSORED BY MINITAB CONNECT

Minitab Connect™

# About the Author

Dave Wells is an advisory consultant, educator, and industry analyst dedicated to building meaningful connections throughout the path from data to business value. He works at the intersection of information management and business management, driving business impact through analytics, business intelligence, and active data management.

More than forty years of information systems experience combined with over ten years of business management give Dave a unique perspective about the connections among business, information, data, and technology. Knowledge sharing and skills building are Dave's passions, carried out through consulting, speaking, teaching, and writing.

# About Eckerson Group

Eckerson Group helps organizations get more value from their data through research, consulting, and education. Our experts each have more than 25+ years of experience in the field, specializing in business intelligence, data architecture, data governance, analytics, and data management. We provide organizations with expert guidance during every step of their data and analytics journey. Get more value from your data. Put an expert on your side. Learn what Eckerson Group can do for you!

Eckerson Group
RESEARCH CONSULTING ADVISORY
GET • MORE • VALUE • FROM • YOUR • DATA

# About This Report

To conduct research for this report, Eckerson Group interviewed numerous industry experts and viewed a dozen or more demos of data pipeline automation tools. The report is sponsored by Minitab Connect who has exclusive permission to syndicate its content.

# Table of Contents

# Executive Summary

Explosive growth in data volumes, users, and use cases causes pain for all data stakeholders, and especially for architects and data engineers. Demand for analytics-ready data far outstrips the capacity of data engineering groups to build and support data pipelines. This data supply-and-demand imbalance creates pressures for everyone who works with data, and creates a multitude of data management challenges.

Architecture is an essential first step along the path to data pipeline automation. Data pipeline architecture is the framework that establishes standards and conventions for data pipelines and supports definition of patterns and templates to achieve consistency and accelerate pipeline development. Well-architected data pipelines lead to marked improvements in data access, analytics value, data engineer and data analyst productivity, and ability to adapt to changing business, regulatory, and technical environments.

Despite common use of the term data engineering, most modern data pipelines are handcrafted with little attention to frameworks, standards, reusable components, and repeatable processes. Data pipeline automation shifts pipeline development away from handcrafting, moving closer to true engineering discipline. Data pipeline automation uses technology to gain efficiency and improve effectiveness for data pipeline development and operations. Data pipeline automation goes beyond simply automating the development process to encompass all aspects of pipeline engineering and operations including design, development, testing, deployment, orchestration, and change management.

## Key Takeaways

- Demand for analytics-ready data far exceeds the capacity of data engineers to produce data pipelines. And we can't hire and train enough engineers to close the gap.
- Data pipeline automation is a practical solution and the technology exists today.
- Automation doesn't work well without architecture. Pipeline architecture defines standards, patterns, templates, and reuse opportunities.
- Automation doesn't work without a repeatable process that defines the steps to define, build, and operate data pipelines. These steps are the things that can be automated.

## Recommendations

- Design data pipeline architecture that defines the standards, patterns, and templates that are needed to drive reuse and consistency of pipeline design and structure.
- Define a repeatable process to design and build data pipelines. Make it an output-driven process that begins with who needs data and why it is needed.
- Expect and plan for architecture and process evolution. Avoid having them become "shelfware" by actively seeking and responding to feedback from data engineers.
- Automate! Implement and use data pipeline automation technologies that will enable data engineers to produce reliable data pipelines at high speed and with minimum manual effort.

# Introduction

## The Science and Art of Data Engineering

Data engineering performs the lifecycle of work involved in assembling data for analytics. This means architecting, designing, building, operating and adapting the pipelines that ingest, process and deliver data from source to consumer. While the discipline was born in the world of SQL, ETL and data warehousing, data engineering evolved in recent years to address data science. Many data engineers now execute projects that support scripting with python and R, often to apply advanced algorithms to semi- or unstructured data.

While data engineering focuses on logical, left-brain tasks, like most "scientific" endeavors it also requires creative, even artistic right-brain thinking about what tasks matter, how to abstract specifics into reusable patterns and frameworks, and where to apply your grey matter. This report seeks to address both sides – both the science and the art – of data engineering.

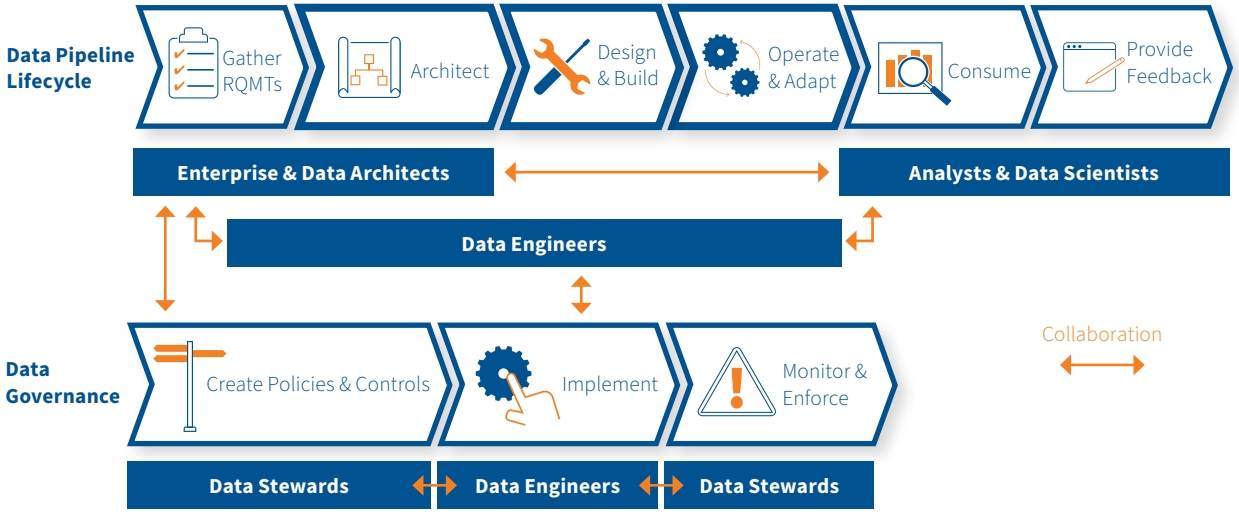
### The Data Pipeline Lifecycle

Throughout the lifecycle of a data pipeline—from concept, through development, and on to operations—building and managing a data pipeline involves many roles and many tasks.

- Enterprise architects, data architects, and data engineers **gather requirements** from the BI analysts, data scientists and business managers about the data that is delivered to them.
- These architects, as their name implies, then **architect** data pipelines by creating the guidelines to achieve quality and consistency in design and development.
- Architects and data engineers **design and build** pipelines. They identify their users (i.e., data consumers), define their use cases, and inventory their data sources. They apply design patterns such as ETL or streaming to manage the ingestion, processing and delivery of data to support those use cases.
- Data engineers **operate and adapt** pipelines. They maintain, monitor and tune various pipeline components, including sources, targets and processors, as well as their interconnections. In addition, they add and remove these components to answer changing business requirements. They also re-configure pipelines to manage the cascading effects of these changes.
- Analysts and data scientists **consume** the data emanating from their data pipelines, and **provide feedback** on data quality, SLAs, etc. to assist architects and engineers with ongoing architecture, design and adaptations.
- Architects and data engineers also help data stewards and other governance managers **create policies and controls**, which data engineers then **implement**. Data stewards **monitor** data usage and **enforce** those policies to maintain data quality.

Figure 1 illustrates the stages, roles and points of collaboration in this highly iterative and cross-functional process. While not shown here, architects and data engineers also collaborate with application developers to embed analytics output and functionality into business processes.

**Figure 1. Data Pipeline Stages, Roles and Collaboration**




This report explores the Architect, Design & Build, and Operate & Adapt stages, as well as the numerous touch points with other stages and roles.

# Data Pipeline Realities

Modern data pipelines are essential for data-driven enterprises and for digital transformation of business. Data pipelines offer many benefits and serve many use cases, yet they are fraught with challenges. Architecture and automation are the keys to overcoming data pipeline and data engineering challenges.

## Benefits of Well-Architected Data Pipelines

Enterprises need data pipelines now more than ever to manage explosive growth in both the supply of and demand for data. Pipelines that efficiently and effectively balance these forces, at scale, yield business benefits that include the following.

**Data access.** Well-architected data pipelines deliver data of high quality to more analysts and managers at greater speed and more reliably than would otherwise be possible. They enable a wide range of decision makers to access the right data, at the right time, in the right place.

**Analytics value.** Enterprises improve performance when decision makers, from line managers up to and including the C-suite, can access the right data at the right time and place. With trusted and timely analytics they can make fast, well informed, and risk-aware decisions to drive positive business results.

**Productivity.** Well-architected pipelines deliver more data to the business per unit of input, with a lower error rate, than is otherwise possible. Analysts answer more questions about business operations more thoroughly than they did before. Data scientists are able to build accurate models, trained on high volumes of high-quality data, while meeting tight timelines.

**Flexibility.** Modern data pipelines are built to accommodate growth, reconfiguration and component changes. They use elastic, scalable cloud infrastructure to handle growing data volumes, users and usage and to dynamically adapt to workload peaks and valleys. Data engineers can quickly reconfigure pipelines to address new requirements for governance, security, and performance.

## Adoption Patterns

Data pipelines are central to strategic data and analytics initiatives that are fundamental to becoming data driven and pursuing digital transformation of the enterprise. These inter-related initiatives seek both to improve the efficiency of data management and to amplify the benefits of analytics insights.

**Cloud migration.** Data teams complement and replace on-premises platforms such as databases, data warehouses and data lakes with Infrastructure as a Service (IaaS) offerings from cloud service providers. Cloud migration helps to achieve economic gains and operational flexibility.

**Data modernization.** As enterprises outgrow the inflexible architecture and proprietary formats of legacy data warehouses, they migrate old workloads and spin up new workloads on open, economic and elastic IaaS data warehouses. They convert legacy data warehouses to cloud and complement relational data with NoSQL databases to accommodate the expanding variety of data sources and data types.

**Self-service.** Many enterprises experience and encourage rapid growth in the number of line-of-business professionals who meet their own reporting, query, and data analysis needs with limited IT support. This trend increases the need for automated and flexible data pipelines that handle frequently changing users, datasets, and data requirements.

**Advanced analytics.** Businesses seek to compete effectively by pursuing analytics initiatives that rely on artificial intelligence (AI) and machine learning (ML), as well as new data sources such as Index of Things (IoT) sensors. AI/ML based applications drive business value through prediction, recommendations engines, and automation. These workloads typically require high volumes of well-processed data.

**Real-time data.** Enterprises replace legacy batch data ingestion and processing with real-time mechanisms such as change data capture (CDC) and message streaming. The real-time data trend applies both to new data pipelines and to reengineering of existing data pipelines to accelerate data delivery, improve processing efficiency, and reduce data latency.

## Use Cases

Analytics use cases vary as much as the myriad sources, users and components that comprise modern pipelines. They range from traditional business intelligence (BI), to data science, to full integration with automated business processes. To understand potential use cases, consider the various uses of data and the business value derived from each.

**Reporting** uses data to inform the business about what is happening when the business questions are known. BI analysts continue a decades-long tradition of tracking business operations by running scheduled or ad-hoc reports on various Key Performance Indicators.

**Discovery** uses data to inform the business about what is happening when specific questions are not known or asked. Data mining applications explore large volumes of data to identify patterns, trends, and anomalies that, with further analysis, lead to insights and new business understanding.

**Diagnosis** uses data to understand why things happen in the business. Diagnostic analytics uses techniques such as correlation and regression to identify connections between leading and lagging indicators, understand influences on business outcomes, and help managers and decision makers to identify the levers that they can use to change future business outcomes.

**Prediction** uses data to inform the business about what is likely to happen in the future. Predictive analytics helps business managers look over short, medium- and long-term horizons, and to make informed decisions based on probabilities.

**Prescription** uses data to recommend what to do in a business situation. As competition accelerates, decision makers need to reduce response times and use analytics to prescribe action.

**Automation** uses data in ways that enable machines and technology to take business actions by embedding analytics directly into business processes. Robotic process automation, for example, can approve a loan, provide an insurance quote, or trigger an automatic inventory request as part of a sales transaction—all occurring without human interaction.

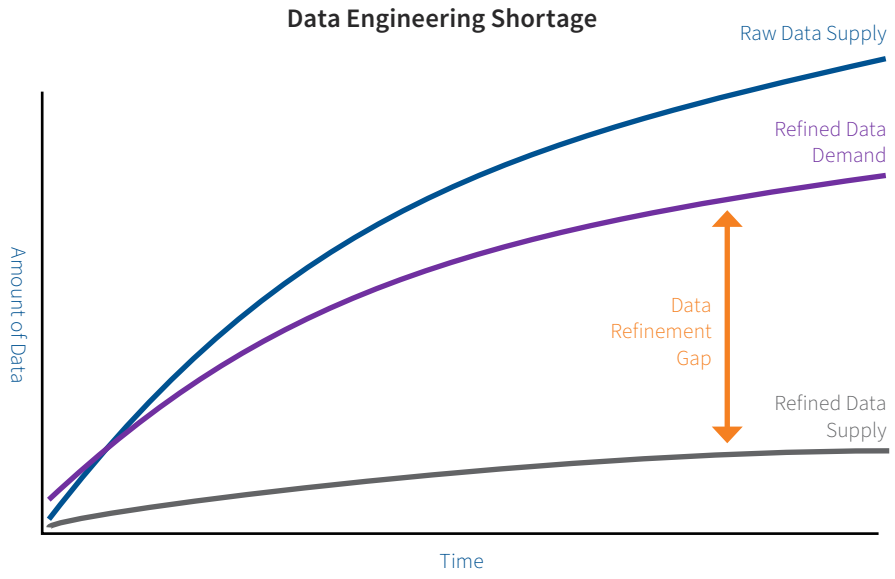
## Challenges: Pressure from All Sides

The explosion in data volumes, users, and use cases inevitably causes pain for all data stakeholders, and especially for architects and data engineers. A data supply-and-demand imbalance creates pressures for everyone who works with data, and creates a multitude of data management challenges.

## Data Supply and Demand

The explosive growth of data sources, volumes, and variety—widely known as big data—is at the heart of data supply-and-demand problems. "Data is the new oil" was a popular saying in the early years of the big data phenomenon. Building on that metaphor, consider the scenario where crude oil is plentiful, refinery capacity is severely limited, and the pressing need is for refined jet fuel. The realities would be unsatisfied demand for jet fuel, high cost of storing and managing unused crude, and much angst and unhappiness among oil stakeholders. Figure 2 illustrates these realities for data.

**Figure 2. The Data Supply and Demand Problem**




From the beginning of the big data era, the supply of raw data has grown rapidly, with similar growth of demand for data pipelines to refine data. Raw data is plentiful, yet analytics-ready data is in short supply. Due to a shortage of data engineers, the capacity to produce data pipelines that refine raw data and create analytics-ready data is severely limited. Increased capacity for data engineering is needed to fill the data refinement gap, but capacity is limited because we can't hire and train enough data engineers to meet the demand. Adding FTEs isn't a practical solution. Architectural discipline and intelligent automation are essential to overcome the limits, reduce the backlog, and meet future demand.

## Pressure from All Sides

The following challenges underscore the need for enterprises to ease the pain and dislodge bottlenecks with well-architected data pipelines.

**Data volumes, variety, and velocity explode.** Modern IT organizations often drown in data. Volumes rise as enterprises digitize, bringing more processes, stakeholders and touchpoints online. New types of sources, including IoT sensors, social media streams and mobile phones, introduce new data types and formats that IT struggles to manage and deliver to data scientists.

**Data consumers proliferate.** As business analysts and managers build data into their decisions, a rising portion of them needs direct access to BI tools as well as custom datasets and views. This explosion in data consumption generates requirements that cascade back through the pipeline: new BI user accounts, use cases, models, formats and sources.

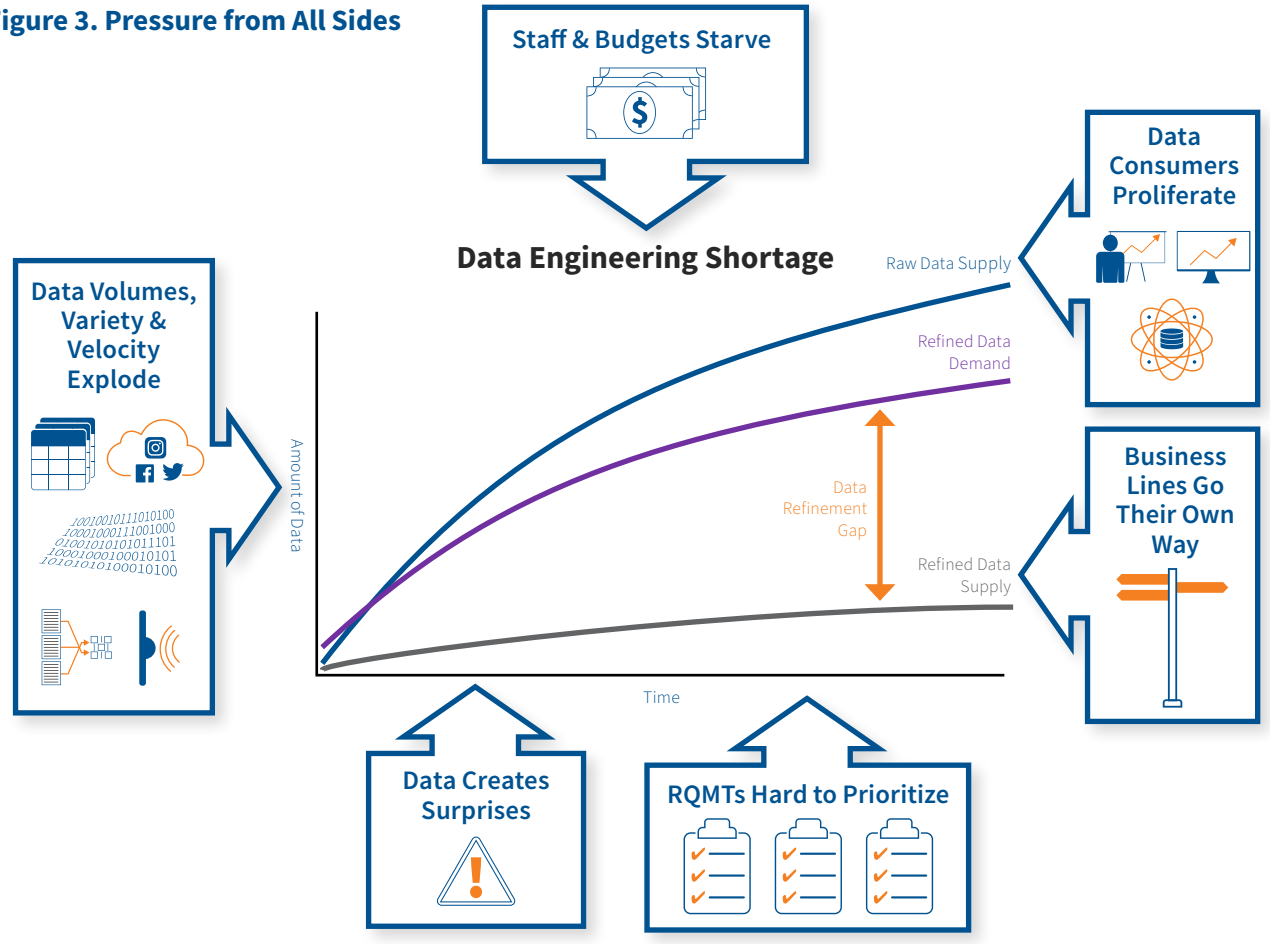
**Staff and budgets starve.** Data analytics leaders rarely receive budget increases to match the rising demands on their teams. Scarcity forces them to prioritize activities, ration resources and improve efficiency wherever possible.

**Lines of business go their own way.** Rather than waiting on IT, BI analysts or even data engineers within the lines of business often build their own data reports, formats, data sets and pipelines. Without guidance and a culture of managed self-service, these practices can undermine data quality, data governance and regulatory compliance efforts.

**Requirements become hard to prioritize.** As data and analytics activities become more diffuse, central IT departments lose visibility into line of business requirements. They struggle to regain control, prioritize requirements, and reconcile the overlapping needs of competing stakeholders. Conflict and inconsistency of priorities among IT and various lines of business leads to confusion, uncertainty, waste, rework, redundancy, and inefficient use of technical and human resources.

**Data creates surprises.** Decentralized data pipelines also raise the risk of data "surprises." These take the form of data quality issues, compliance gaps, or possibly ground-breaking insights. Whether positive or negative, such surprises have inter-departmental implications that need—but often do not receive—centralized attention. Without a coordinated response, the costs rise and benefits go unrealized. Figure 3 illustrates these challenges.

**Figure 3. Pressure from All Sides**



# Architecting Data Pipelines

Architecture defines the roles, structures, relationships, and rules by which a collection of components constitutes a cohesive whole – the glue that bonds individual parts into a system. Architecture is an early-stage design activity that precedes detailed design, specification, and construction. Effective architecture ensures that the things we build:

- Are suited to the purposes for which they are intended
- Fit gracefully into their environment
- Are structurally sound
- Comply with codes, regulations, and standards
- Are sustainable through their expected lifespan
- Are aesthetically pleasing

These fundamental principles hold true for architecture of many things—buildings, bridges, information systems, and more. They are equally important for data pipelines.

Data pipeline architecture (and really every aspect of data architecture) is ideally designed at two levels—enterprise architecture and applied architecture. Enterprise architecture establishes a referenceable framework for data pipeline developers that describes pipeline components and their relationships at a high level. It is a framework that establishes standards and conventions for data pipelines and supports definition of patterns and templates to achieve consistency and accelerate development of data pipelines. Applied architecture extends, adapts, and details the enterprise architecture to create high-level design for a single specific data pipeline.

## Data Pipeline Components

Data pipelines are complex applications that depend on many interrelated components to move data from a point of origin to a destination. An enterprise architecture view, as shown in figure 4, works well to illustrate the components and their relationships.

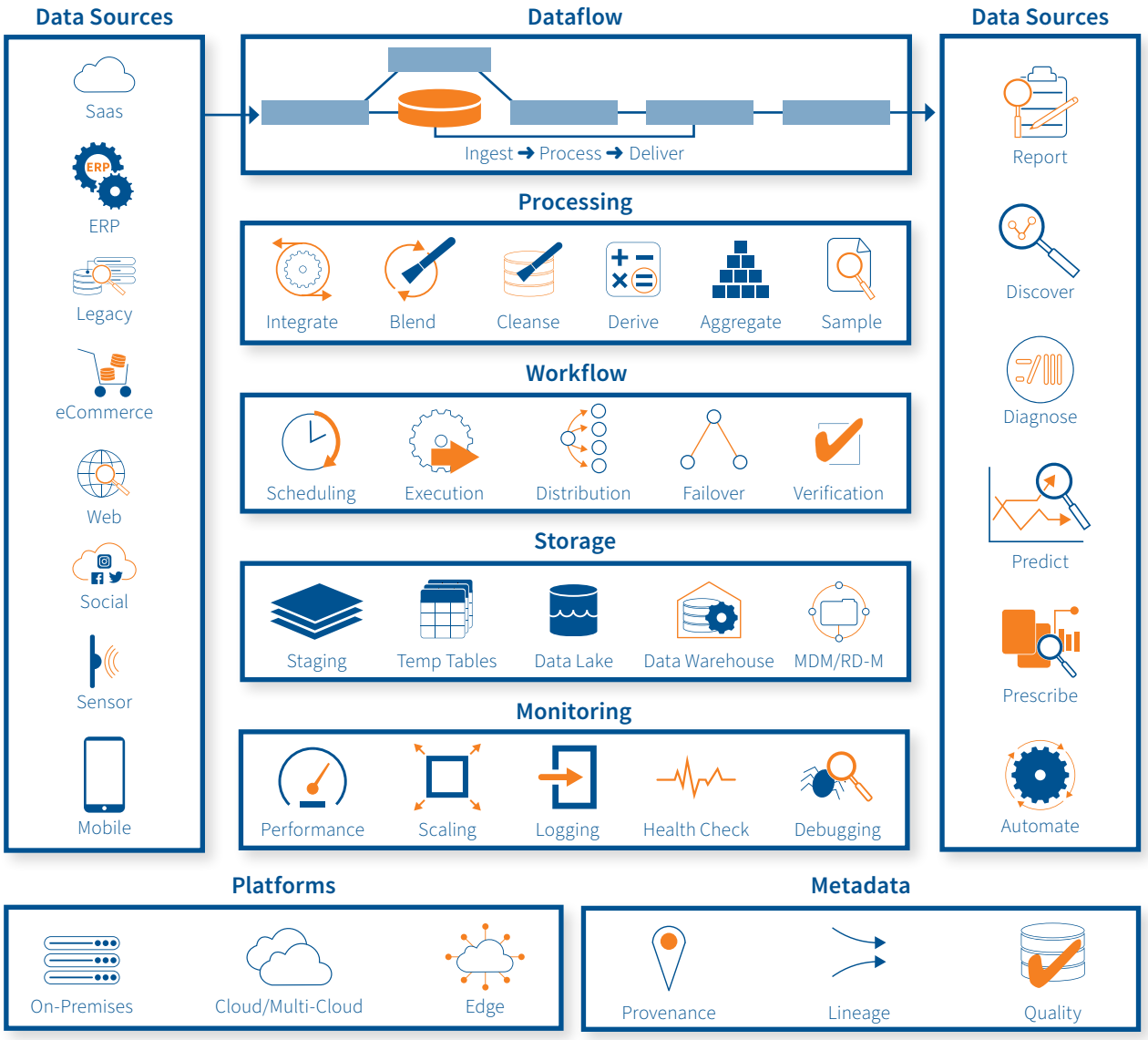
### Data pipelines consist of nine types of components:

**Data sources** are the ultimate origins of data when viewing data pipelines from an enterprise perspective. They are also the origin of many data pipeline complexities and challenges as described earlier with explosive growth in data volume, variety, and velocity. Enterprise data pipeline architecture is ideally designed to handle all types of data at any speed, from batch ETL to streaming in real time.

**Use cases** are the ultimate destinations for data. These are the BI and analytics applications where data is applied in ways that create business value. Enterprise data pipeline architecture must support all use cases, from information and decision support to prescriptive analytics and automation.

**Dataflow** describes the sequence of processes and data stores through which data moves to get from origin to destination. For a single pipeline instance, the origin may be an original data source or an ecosystem database such as a data warehouse or data lake. The destination may be a target business use case or an ecosystem database.

**Figure 4. A Data Pipeline Reference Architecture**




**Processing** encompasses all of the steps and activities that are performed to ingest, persist, transform, and deliver data. Much of pipeline processing is typically directed at data transformation—integration, blending, cleansing, data element derivation, aggregation, and sampling of data.

**Workflow** describes the sequencing of and dependencies among pipeline processes. Well architected workflow addresses both successful execution and handling of errors, exceptions, and processing failures. Well defined workflow is a prerequisite for data pipeline orchestration.

**Storage** includes all stores where data is persisted at various stages as it moves through the pipeline. This includes ecosystem databases such as data lakes, data warehouses, and MDM repositories as well as more transient zones such as data staging and temporary tables.

**Monitoring** provides the capabilities needed to observe data pipelines in operation and to ensure healthy and efficient pipelines. Monitoring also has an important role in data pipeline orchestration.

**Platforms** are the environments in which data pipelines are executed, including on-premises servers, cloud and multi-cloud, and edge computing. Hybrid pipelines operating across any or all of these platforms are becoming increasingly common. When designing cloud platforms, it is important to distinguish between cloud-hosted and cloud-native. Cloud native applications require a fine-grained design that can be built as a collection of microservices. Microservices architecture is also essential for edge computing where processing is done at or near the source of the data.

**Metadata** is an essential output of data pipelines. In addition to producing data for delivery to a destination, pipelines must also produce the metadata that describes data provenance, lineage, and quality. These types of metadata are fundamental for delivery of trusted data. The scope and types of metadata expands when pipeline automation is metadata driven. Execution schedules, for example, become essential metadata for orchestration. Similarly, schema metadata becomes critical when AI/ML is used to detect and adapt to schema changes.

## Data Pipeline Design Patterns

When designing and building a data pipeline, much of the work focuses on dataflow. When data engineers must support and maintain operating data pipelines, consistency among dataflow designs makes the job much easier. Design patterns as part of data pipeline enterprise architecture are beneficial for the architects and engineers participating who design, build, and support data pipelines. When designing enterprise data pipeline architecture, begin with the eight common patterns illustrated in figure 5, then expand them based on your experiences and needs.

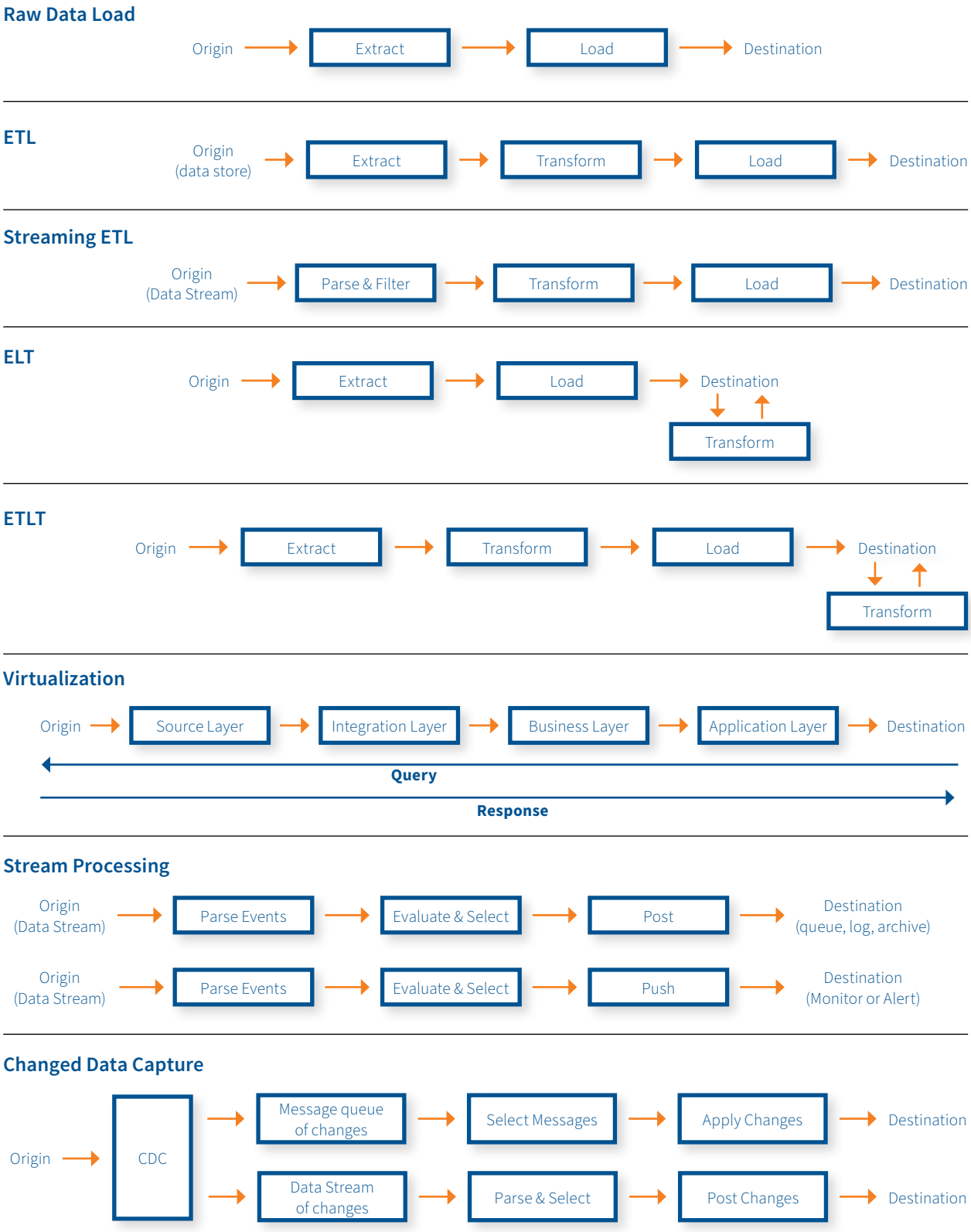
**Raw data load**, is a very basic two-step pipeline built to move data from one database to another. These pipelines perform the bulk data movement that is needed for the initial loading of a database such as a data warehouse, or for migration of data from one database to another—from on-premises to cloud, for example.

**Extract-Transform-Load (ETL)** is the most widely used data pipeline pattern. Data is extracted from a data source, then transformed to cleanse, standardize, and integrate before loading into a target database. ETL processing is executed as scheduled batch processing, and data latency is inherent in batch processing. ETL is well-suited for data integration when all data sources are not ready at the same time. Each individual source is extracted when ready with transformation and loading held until all extracts are complete.

**Streaming ETL** adapts the ETL pattern typically used with stored data to work with data streams. Instead of extracting from a data store, streaming data is parsed to separate individual events into unique records, then filtered to reduce the data set to only the events and fields of interest for the use case. Parsing and filtering are followed with transformation and loading. This pattern is particularly useful for machine learning use cases that often focus on only a few fields in much larger data sets.

**Extract-Load-Transform** (ELT is a variation on ETL used to offset the latency of pure ETL processing. Waiting for all transformations to complete delays availability of data for business use. Loading immediately after extract, then transforming in place, reduces the delay. ELT accelerates data availability when multiple sources that are not simultaneously ready would be held in a staging area with ETL processing. With ELT the data warehouse serves as data staging so each extract becomes available for use immediately. Data transformation is performed in place in the data warehouse once all extracts are loaded. Data quality and privacy are a concern with ELT processing. When

**Figure 5. Data Pipeline Design Patterns**



data is made available for use without transformation, it is exposed without first performing data cleansing and sensitive data masking or obfuscation.

**Extract-Transform-Load-Transform** (ETLT) is a hybrid of ETL and ELT. Each source is extracted when ready. A first stage of "light" transformation is performed before data is loaded. First stage transformations are limited to a single data source independent of all other data sources. Data cleansing, format standardization, and masking of sensitive data are typical kinds of first stage transformations. Each data source becomes available for use quickly but without the quality and privacy risks of ELT. Once all sources have been loaded, second stage transformation performs integration and other multi-source dependent work in place in the data warehouse or data lake.

**Data virtualization** serves data pipelines differently than the extract-based patterns. Most pipelines create physical copies of data in a data warehouse, data lake, or dataset. Virtualization delivers data as views without physically storing a separate version. Virtualization works with layers of data abstraction. The source layer is the least abstract, providing connection views through which the pipeline sees the content of data sources. The integration layer combines and connects data from disparate sources, providing views similar to the transformation results of ETL processing. The business layer presents data with semantic context, and the application layer structures the semantic view for specific use cases. Unlike ETL processing initiated by a schedule, virtualization is initiated by a query. The query is issued in application and semantic context, then is translated through integration layers and source layers to connect with the right data sources. The response reverses the path to acquire source data, transform and integrate, present a semantic view, and deliver an application view of the data.

**Stream processing** has two similar but slightly different patterns. In both patterns, the data origin is a continuous flow of event data in chronological sequence. Processing parses the stream to isolate each unique event as a distinct record. Individual events are evaluated to select only those appropriate to the use case. At the destination end of the data flow, the two patterns diverge slightly. Some use cases post events to a message queue where they become the input to a downstream data pipeline in which data consumption is somewhat latent. Other use cases push events to a monitoring or alerting application where information about the state of a machine or other entity is delivered in real time.

**Change Data Capture (CDC)** is a technique used to increase the freshness of data that is typically latent. Many operational data sources that flow into data warehouses and data lakes are processed using inherently latent batch ETL. CDC, when applied to operational databases, identifies data changes as they occur, then delivers information about those changes in either two forms. Pushing changes to a message queue makes them available for downstream mini- or micro-batch processing and substantially reduced data latency. CDC with streaming makes data changes available immediately, shifting from batch to real-time data ingestion.

## Designing Data Pipelines

A systematic approach to designing and building data pipelines is a necessity if we are to scale data engineering to catch up to and keep pace with accelerating demand. Ad hoc approaches and hand-crafting of pipelines is labor intensive, slow, and difficult to automate. Define a repeatable process to design and build data pipelines.

Make it an output-driven process that begins with who needs data and why it is needed. Start with the destination. Know where data is needed and why it is needed. Then look at origin to identify and understand the data that will enter the pipeline. With origin and destination understood, design the dataflow—the sequence of process and data stores through which data moves to get from origin to destination. This is the point at which you'll apply the data pipeline design patterns described above.

Next design data storage, both transient and intermediate datasets, and end point datasets where the pipeline destination is a persistent data store. Within the structure of origin, destination, dataflow, and storage you can now design the nuts-and-bolts of a data pipeline—the processing. Dataflow and data processing work together as the core of a data pipeline. The right processes executed in the right sequence turn inputs (origin data) into outputs (destination data). Much of the work of data pipelines is focused on data—i.e., getting the right data in the right forms and formats for intended uses. The three primary reasons for data transformation are improving data, enriching data, and formatting data.

Complete the pipeline design with operational context. Define the workflow that manages sequencing and dependencies of processes and tasks in the data pipeline. Consider pipeline monitoring needs and know how your data management technologies support them. Determine the execution platforms to describe where pipeline processing will be executed, both in server and software environments. Finally, build metadata requirements into the design. Know what metadata is desired and what metadata is required. Then identify the processes and the technologies responsible for metadata collection.

# Automating Data Pipelines

Despite the term data "engineering," most modern data pipelines are handcrafted with little attention to frameworks, standards, reusable components, and repeatable processes. Data pipeline automation shifts pipeline development away from handcrafting, moving closer to true engineering discipline. Data pipeline automation uses technology to gain efficiency and improve effectiveness for data pipeline development and operations. It goes beyond simply automating the development process to encompass all aspects of pipeline engineering and operations—including design, development, testing, deployment, scheduling, orchestration, monitoring, security, and change management. Most aspects of data management—including data discovery, ingestion, and preparation—also stand ripe for automation.

Each of these tasks involves a high percentage of repetitive work that software can execute faster, and with greater reliability and accuracy, than hands on a keyboard. Software vendors drive this automation trend by standardizing such processes and masking their complexity behind intuitive graphical user interfaces. Automation technologies for pipeline creation include functions for connecting to databases and other data sources, managing repetitive transformation tasks, managing schema, and monitoring data lineage. Automation of pipeline operation includes functions to schedule jobs, execute workflows, coordinate dependencies among tasks, and monitor execution in real time. Let's explore the impact and benefits of automation on key data pipeline stakeholders.

### The Architecture Perspective

As previously discussed, architecture is best designed at two levels—enterprise architecture to establish standards and conventions, and applied architecture to design specific data pipelines.

- Enterprise-level architects define and create standards, templates, and patterns to guide pipeline design and development for the organization. They embed architectural standards into automated tools, apply and adapt the templates and patterns that are built into tools to best fit enterprise requirements. Application-level architects apply the standards, templates and patterns of automated tools as they define and design data pipelines.
- By designing and applying architectural frameworks, architects at both levels accelerate the design process and make it easy to comply with architectural standards. They also create adaptable architectures and anticipate the impacts of future changes.

### The Engineering Perspective

Data engineers design and build data pipelines, often performing the applied architecture activities before undertaking detailed design and specification. The creative work of data engineering is in design and specification, not in time-consuming coding and scripting.

- Data engineers learn and adopt the architectural standards and conventions of automated tools. They can reuse rather than duplicate or reinvent components.
- Engineers benefit from eliminating the drudgery of handcrafting and scripting highly similar pipelines. They reduce backlog and remove the pressures of unsatisfied demand for data. They spend less time on repetitive scripting and more time on planning, process, and governance.

### The Operations Perspective

Managing data pipeline operations is a complex job to ensure that pipelines operate as planned and that data is delivered as expected and needed. Multiple dependencies of scheduling, workflows, and error handling are big challenges for this critical activity.

- IT operations managers learn and use automated orchestration and monitoring tools.
- They improve productivity with automated scheduling, managed workflow and real-time monitoring of pipeline execution. When things go wrong, they run impact analysis and recovery planning tasks. IT operations managers also can scan pipeline task histories to help problem solve and improve processes.

### Data Governance Perspective

Data governance in context of data pipelines focuses primarily on two areas—data protection and data lineage.

- Automation for data protection recognizes sensitive data, connects with policies for data protection, and applies those policies for data masking, obfuscation, or security protections. All data stakeholders benefit when data protection policies are applied reliably and consistently.
- Automation for data lineage builds metadata collection into data pipelines to maintain an end-to-end record of the sources, flows, and processes that shape the data as it moves from origin to destination. Data consumers have greater trust in data when lineage is fully known and described. Data engineers and operations staff depend on lineage metadata to trace and troubleshoot when things go wrong.

### The Business Perspective

Business stakeholders are the ultimate destinations and consumers of data so they clearly benefit from increased capacity to meet the demand for analytics-ready data. Perhaps equally important, but less recognized, they are the ultimate source of feedback that is needed to continuously improve data delivery processes.

- Business stakeholders are responsible to communicate clear data requirements and recognize similarities among those requirements. They also provide feedback to help engineers reuse components and achieve DataOps-like agility.
- Business stakeholders benefit from timely and comprehensive data access. BI analysts and business managers can further increase data access with self-service, performing data preparation and analytics tasks themselves. This reduces delays and enables data-driven decisions.
- The business value of analytics also increases as manual errors decline and data quality improves.

### Overall Benefits and Considerations

Automating the critical activities of modern data pipelines offers many enterprise-wide benefits—repeatable processes, reusable components, reliable results, consistency, maintainability, adaptability, and more. Automation also reduces risk and increases agility. Stakeholders commit to projects with newfound confidence that they will hit deadlines and budget targets. They execute more projects, faster than was possible with manually-intensive data pipelines. They respond rapidly to changing business requirements, re-configuring data pipelines and creating new pipelines with graphical low-code/no-code interfaces and reuse of existing and proven components. New hires among data and analytics teams onboard quickly when reuse and consistency shorten the learning curve.

While data pipeline automation has many benefits, it is not a silver bullet. Don't seek to automate everything and everywhere. Choose carefully where to automate, recognizing that complex business processes and corresponding data needs should not be oversimplified. Standards must always be tempered to accommodate the unusual, the exceptions, and the inevitable need to customize. Anticipate the need for custom scripting and specialized code, and expect data pipeline tools to interoperate with that code. When getting started with pipeline automation, begin with relatively straightforward pipeline requirements, gain experience, and add complexity as you learn and grow.

# Architect, Automate, and Tool Up

Rocketing growth in data supply and demand puts unprecedented strains on the data engineers who produce data pipelines, the data analysts who need analytics-ready data, and the business managers and decision makers who depend on data and analytics to do their jobs. Meeting the ever-expanding demand for analytics-ready data depends on automation to increase speed of development, reduce manual effort, improve quality, and optimize data engineering processes.

Automation, in turn, depends on architecture. It is impractical to automate without first defining the parts, roles, relationships, structures, standards, and patterns that are fundamental to reusable components and repeatable processes. Enterprise architecture describes the architectural foundation as an abstraction. Applied architecture uses the architectural concepts and principles to build, deploy, and operate data pipelines—to deliver that data that is needed for many data use cases—reporting, discovery, diagnosis, prediction, prescription, and automation.

Designing, building, and operating data pipelines is complex, daunting, and difficult. The difficulties are compounded and amplified by exceptional demand that far exceeds data engineering capacity. But it is not a hopeless situation and the path forward is clear—architecture and automation built on the concepts and guidelines described in this report and implemented with the smart automation technologies offered by forward thinking software vendors.

Plan and navigate your path to data pipeline automation with these recommendations.

- Design your data pipeline architecture taking into account all of the pipeline components and defining the standards, patterns, and templates that are the basis for reuse and consistency.
- Define your repeatable process to design and build data pipelines, ensuring that every pipeline is an instance of applied architecture. Start with the destination, knowing who needs data and why. Next understand data origins—the sources—before designing dataflow, storage, processing, etc. Use design patterns and seek opportunities to reuse existing pipeline components.
- Listen and respond to feedback from data pipeline developers. Evolve and adapt both architecture and process based on experience and in response to business and technical change.
- Tool up! Seek out the pipeline automation technologies that enable your data engineering teams to produce reliable data pipelines at high speed and with minimum manual effort. Evaluate the tools based on their ability to minimize complexity, integrate with heterogeneous environments, connect with many data sources, allow custom scripting when needed, and support the full lifecycle of data engineering—design, development, operation, and orchestration.

# 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 expert guidance during every step of their data journey.

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. We share best practices that align your team around industry frameworks.
- **Our consultants** listen carefully to craft tailored solutions that translate your business requirements into compelling strategies and solutions.
- **Our educators** share best practices in consulting workshops or external conferences on 30+ topics.

Our experts each have more than 25+ years of experience in the field. They specialize in 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 by using their extensive experience.

Our clients say we are hard-working, insightful, and humble. It all stems from our love of data and our desire to help you get more value from analytics—we see ourselves as a family of continuous learners, interpreting the world of data and analytics for our clients and partners.

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

Eckerson Group

RESEARCH | CONSULTING | ADVISORY

GET • MORE • VALUE • FROM • YOUR • DATA

# About Minitab Connect


Access, blend, and enrich data from all your critical sources for meaningful business intelligence and confident, informed decision-making with Minitab Connect™. Feed analytics initiatives and foster organization-wide collaboration with self-serve tools for data integration, automation, and governance. Data users from across the enterprise can effortlessly blend and explore data from databases, cloud and on-premise applications, unstructured data, spreadsheets, and more. Flexible, automated workflows accelerate every step of the data integration process, while powerful data preparation and visualization tools help yield transformative insights. Discover how you can accelerate your digital transformation with Minitab Connect at **minitab.com/connect**.

For nearly 50 years, Minitab has helped companies and institutions spot trends, solve problems and discover valuable insights in data by delivering a comprehensive and best-in-class suite of data analysis and process improvement tools. Combined with unparalleled ease-of-use, Minitab makes it simpler than ever to get deep insights from data.

Thousands of businesses of all sizes and industries worldwide, including the Top 10 Fortune Companies and 85% of the Fortune 500, use and trust Minitab® Statistical Software, Companion by Minitab®, Minitab Workspace™, Minitab Connect™, Quality Trainer® and Salford Predictive Modeler® to make better, faster and more accurate decisions to drive business excellence.