CHECKLIST REPORT

NOVEMBER 2016

Building a Data Lake with Legacy Data

By Krish Krishnan

Sponsored by:

syncsort

tdwi
Transforming Data With Intelligence™

NOVEMBER 2016

TDWI CHECKLIST REPORT

Building a Data Lake with Legacy Data

By Krish Krishnan

TABLE OF CONTENTS

tdwi

Transforming Data With Intelligence™

555 S. Renton Village Place, Ste. 700
Renton, WA 98057-3295

T 425.277.9126
F 425.687.2842
E info@tdwi.org

tdwi.org

FOREWORD

One of the biggest developments for data managers in the last decade has been the creation of the enterprise data lake. A data lake is usually thought of as the collection and collation of all enterprise data from legacy systems and sources, data warehouses and analytics systems, third-party data, social media data, clickstream data, and anything else that might be considered useful information by the enterprise. Although the definition is interesting, is it actually possible? Do we have the infrastructure and skills to make this work? Will we end up with better insights and opportunities than we have today, or will they just wind up submerged in the lake?

These concerns are greater at organizations with extensive legacy data sources—such as mainframes and data warehouses—that can store decades' worth of critical information. Unlike startups, or even the Bay Area giants that found the earliest success with data lakes, most of the Fortune 500 have to make sure all their established data sources are connected while maintaining security and governance protocols. These challenges are valid, but fortunately, they are also addressable.

Several key requirements make a data lake successful:

- A flexible infrastructure
- A scalable architecture
- Extensible metadata and semantic libraries
- Master data and other reference data
- Multi-format data processing capability
- Data integration for all datasets
- Efficient tools with minimal overhead for ingesting and processing data

There are also several key challenges in building a successful data lake with legacy data:

- Acquiring the necessary new skills through training or hiring employees
- Keeping up with rapidly evolving big data tools and ensuring interoperability among them
- Managing heterogeneous data formats, especially with mainframe systems
- Ingesting the data quickly and efficiently enough to meet SLAs with business stakeholders
- Assuring clean data lineage, especially with varied ETL methods
- Data quality management

This checklist will help you and your team plan and launch successful data lake projects with your legacy data sources. It reviews the critical success factors for these projects as well as the risks and issues to mitigate.

NUMBER ONE

VERIFY THE QUALITY OF YOUR EXISTING DATA ECOSYSTEM

The goal of the data lake is to be the single integrated platform for generating insight. It is created when data is freed from historical siloes—such as mainframe systems, UNIX-based databases, legacy Web databases, and flat files—to form a new, connected ecosystem. This data ecosystem is vital for the success of all subsequent data intelligence and analytics programs. Therefore, it is important to address the following areas to ensure you are starting with a strong foundation.

- **Data quality** is the top priority of business users and, therefore, must be top priority for IT. Bad source data leads to bad reports, inaccurate analytics, and poor executive decision-making processes. Some organizations prefer to ensure the quality of data in the source systems prior to introducing it to the data lake, while others bring raw data from sources into a "landing zone" and use the Hadoop cluster to create a "clean data zone." Regardless of when and where it happens, data quality must be ensured.
- The level of **data granularity from legacy systems** often varies widely, depending on the type of system and the file format. Too fine a level of detail can be as much of a problem as too coarse an aggregation if it comes at the expense of analytic consistency. Therefore, make sure you verify and validate the granularity of data across the full range of legacy systems to account for any variations.
- **Data ownership** refers to the responsibility for the various source systems. Many enterprises assign specific owners to segments of data. Such assignments ought to be part of the initial process to approve the release of data, provide encryption and masking requirements, determine which users will have access to which data, and describe the data life cycle—especially with respect to mainframe systems, as they are often assumed to be separate from other IT systems.
- **Data access rules**—especially for data from legacy systems—need to be well documented and meticulously implemented. Both the rules and the tools used in this process must also be documented. Data access rules are to be implemented at data extraction, ingestion, and usage points. Issues with this task often only emerge later when the data lake is formed and tested, so keeping it in mind up front will avoid later pain.

It should be noted that although data quality is listed as a separate item in our checklist, data quality will be involved with every other item in this checklist.

NUMBER TWO

ALIGN YOUR DATA FORMATS

The subject of data formats can disrupt nearly any discussion. Consider, for example, that according to the Microsoft Developer Network most structured enterprise data is stored as VSAM files—a file format introduced by IBM in the 1970s for use on its mainframes. However, the issue is not whether a given format is right or wrong, but rather whether the details of data in that format—field separators, line terminators, fixed vs. variable length data, special formats for images and video links, encryption, missing content, security, metadata, and semantic links—are fully identified.

Tools can provide a flexible set of graphical options to export data in different formats, but ingesting files into the data lake may require specific metadata and formats that you'll need to identify. Additionally, moving from the lake into the destination systems may also need the data format to be understood because importing data relies on the format(s) of the source files.

Here are the key areas to pay attention to

- **Field separators and terminators:** Separators and terminators are a small but important portion of data. For example, using common separators, such as commas or tabs, can cause issues because they can often be found in the data itself. One reliably unique separator in use is three consecutive pipe symbols ("lll").
- **EBCDIC-to-ASCII formats:** For mainframe systems, this is critical because the global nature of enterprise data means the source data may be encoded in a number of different character sets. For use cases where you want to keep the data in its original EBCDIC format—a compliance requirement in some industries—it is important that you can still work with the data once it's in the lake. For use cases where the data will be blended with other formats, you will need to convert it to ASCII when bringing it into the data lake. This is a critical feature from a tool perspective when implementing a data lake.
- **Header and trailer formats:** Although there is no format issue for data from databases, mainframe data needs specific header and trailer formats to be created and managed.
- **Metadata:** If you do not have metadata in place in your current environment, you are not ready to transition to a data lake. Metadata is essential to integrating and creating a data lake across any enterprise and should be implemented in the source rather than target systems. (See Number Three.)

NUMBER THREE

DEFINE AND USE ACCURATE MASTER DATA AND METADATA

Master data and metadata are important factors in any data integration project, but they are essential when building and maintaining a data lake—both for its initial creation and especially for updates post implementation.

Master data is the set of corporate data definitions used across all enterprise systems, including legacy systems. The original concept of a data lake did not make use of master data, but the companies involved were quite different from the large *Fortune* 500 organizations now implementing lakes. For organizations with large legacy data stores, master data provides a source of reference data that can be used to tag the data upon ingestion, providing avenues for later auditing and verification. Master data can also be integrated with custom taxonomies to enable searches across the data lake. You may also find tools available that will make use of master data, but you will want to test those in a proof-of-concept to ensure they're useful in your particular environment.

Metadata is technical information integrated when data is originally brought into storage. (There is also business metadata that is added after the fact by various reporting tools, but we're speaking only of technical metadata here.) With respect to legacy data, technical metadata can provide a useful check between the definition of the data at its source and its integrated definition in the data lake. It can also be useful in creating rules for integration, verifying data lineage, and documenting any adjustments to file format.

Five key aspects should also be considered when looking at master data and metadata:

- **Usage:** Does the necessary information exist in the source systems? If not, what needs to be added to the source data prior to extraction?
- **Maintenance:** What is the maintenance cycle for this information? How often is it refreshed? Is the cycle triggered by the source applications? If so, how will it be triggered in the target data lake?
- **Availability:** Is the information readily available for all source data? If not, where can it be retrieved from, and who owns the security and user access?
- **Interoperability:** Can all your source system metadata be leveraged in all your target systems? Do you have Mainframe COBOL Copybook data that needs to be mapped? Are you able to share metadata that originates in various source systems?
- **Stewardship:** From a governance perspective, who are the data stewards for this information? Are there standards and policies for it?

NUMBER FOUR

DATA LAKES STILL NEED DATA GOVERNANCE

Data governance cannot be overlooked by any organization, but especially not by those with extensive legacy data stores, which are often maintained for regulatory and compliance reasons as well as long-term analytics use. Therefore, your existing data governance policies and best practices regarding the sources, rules, processes, metrics, and stewardship for data should be extended to the data lake, not left on the shore.

Making the data lake part of your existing governance program will:

- Ensure executive presence, participation, and sponsorship in the data lake program
- Simplify creation of any necessary committees to ensure governance of the data lake is managed and implemented
- Establish metrics and processes for accuracy and compliance as well as to identify issues and recommend corrective action
- Integrate stewardship of the data lake within the larger enterprise ecosystem; this is especially important if source data needs to be manipulated before being brought into the lake and any exceptions documented
- Confirm that existing security and access control policies, such as those concerning authentication and authorization, can be extended to the data lake
- Allow data from the data lake to be more easily processed and integrated into any executive dashboards used for reporting, compliance, or other governance tasks

As with metadata, be sure you have an effective data governance program in place before launching any data lake program.

NUMBER FIVE

ARCHITECT FOR LEGACY DATA INCLUSION

The intent of the data lake is to improve future operational and business analytics. This involves addressing some of the same architecture issues—network performance, bandwidth congestion, and data processing costs—that hamper traditional data management platforms in addition to the issues of managing and moving the large data sets extracted from legacy sources. There are several infrastructure issues an enterprise must consider.

Data Size, Complexity, and Format

These issues have been discussed earlier, but we return to them here with an eye toward architecture. For example, extracting mainframe data has direct impact on network performance and scalability that aren't necessarily problems when working with database data. This impact will need to be documented and mitigated with the right tools.

This complexity also highlights the requirements for compression of the data, which has impacts on network and resource performance from source to target. There are three key resources that must be balanced to prevent bottlenecks: I/O (including network traffic), CPU, and memory. Compression reduces I/O but increases CPU, which is helpful if I/O is the bottleneck.

The data formats of mainframe sources may impact the network as well when you must convert formats or unpack fields. Document any processing requirements to ensure that appropriate design-time modifications can be incorporated into your final architecture.

Network

Your network is the underlying infrastructure critical to creating and implementing your data programs. For the data lake, there are three key focus areas for networks:

- Requirements: Do we need 10GB/s or 100GB/s network capacity? What capacity is available? How many firewalls, DMZs, and switches do we need to traverse?
- Network risks: Be sure you consider such problems as data breaches, packet losses, repeat of data packets, and failure at a switch. Document these risks and develop mitigation strategies for each.
- Bandwidth: Specify how much bandwidth your network will need. Although tools for exporting mainframe data sometimes include tools for sizing files to ensure smooth performance, this needs to be documented and possibly tested in a proof-of-concept (POC).

Cost

Your estimate of the costs of processing data from legacy systems needs to include resources, network usage, and archival data volumes. Because many organizations implement a data lake with the specific goal of reducing the costs, it's important to accurately account for all your new costs to ensure a good ROI. Costs may be soft and the task of calculating ROI may often be forgotten in the rush to complete the project. Make this a critical step in your project checklist.

Processing Costs

There are many processing costs incurred in processing data for the data lake at both the source and target layers. These costs become impactful when your applications don't account for additional processing overhead or general inefficiencies. Tools that generate inefficient code to perform cleansing or transformation operations are just one example of unnecessary processing overhead. Finding a tool with an engine that is purpose-built for efficiency easily mitigates this risk, as well as lowers overall project costs. Although the impact on source systems can be managed or minimized, the target system needs to be optimized to account for each additional source.

The data lake target system (Hadoop) also imposes several layers of overhead with its libraries, use of Avro or Parquet, YARN, MapReduce, Impala, HIVE databases, and HDFS file management. Such performance factors need to be considered, discussed, and optimized. Several tools in the ecosystem are available that have specific design constraints that must be validated and verified in a POC.

Although discussed here as part of the architecture and planning stages, this area has the most relevance to post-deployment adoption and successful self-service analytics initiatives.

NUMBER SIX

LOOK AT ALL ASPECTS OF DATA SECURITY AS DATA MOVES TO THE LAKE

One of the primary ongoing concerns with implementing data lakes is how to ensure the security of critical enterprise data. Considerable discussion occurs about encrypting "data at rest" as well as "data in motion," among other best practices. However, legacy data often comes with years'—if not decades'—worth of established security protocols, so how does an enterprise manage security issues such as redundancy, semantic data, and masking and encryption when that data is moved to the lake?

- **Redundancy:** After the initial extraction from legacy systems (whether mainframe or databases), often incremental extractions or snapshot exports are done to retrieve data changes. This change data capture keeps the data lake current with still-used source systems, but often leaves behind staged data sets and duplicate copies. Removing this redundant data ensures that unnecessary security risks are also removed, as every copy of data must be protected.
- **Semantic data:** Data is more than just about individual fields. Semantic data, data that specifies two data elements as well as the relationship between them, can be added when you integrate data into a data lake. Data we discuss here has several relationship elements that can be interpreted as key-value column pairs, multiple columns associated with one key, multiple tables represented with complex relationships, graph databases, datasets with external joins to latitude-longitude codes. Managing this data with programming in the ETL or ELT or ETLTLTL process is cumbersome, which is why it a feature of the automated tool you use to extract data from legacy sources.
- **Masking and encryption:** A major concern with accessing legacy source data and moving it into a data lake is the masking and encryption of data. TDWI research repeatedly shows the importance of securing data in all phases—both at rest and in motion. Depending on the particular regulatory environment you operate in, there may be data that cannot be sent to the data lake even if it could be sent securely.

  For example, Canadian law prohibits the transfer of student data across national—and sometimes even provincial—boundaries. Therefore, a multinational provider of education services would not be able to send Canadian student data to their U.S.-based data lake, no matter how secure the transfer. Similar compliance requirements may exist in other segments of data and verticals. (See Table 1.) In addition, smaller pieces of data may need to be masked before being moved.

Given the availability of tools to address these concerns, it is unlikely you would need to develop custom code in languages such as Java or Python for this exercise. By validating the capabilities of available tools and conducting some proof-of-concept experiments, you ought to be able to address these concerns, as enterprises have found when conducting successful mainframe-to-Hadoop projects.

INDUSTRY	COMPLIANCE REGULATION
Credit card processing	Payment Card Industry Data Security Standard (PCI DSS)
Healthcare	Healthe Insurance Portability and Accountability Act (HIPAA)
Life sciences	Genetic Information Non-Discrimination Act (GINA)
Financial services	Sarbanes-Oxley Act (SOX), Dodd-Frank Act, Security and Exchange Commission (SEC) Rule 17a-4
Media and entertainment	The Motion Picture Association of America's security requirements for content movement
Government	Federal Information Security Management Act (FISMA) for U.S. government agencies

Table 1: Examples of industry-specific regulations.

NUMBER SEVEN

TAKE ADVANTAGE OF AVAILABLE TOOLS BUT EVALUATE THEM THROROUGHLY

Modern data management tools are generally able to manage the creation and maintenance of the data lake, with most vendors in the database ecosystem adapting to these new architectural implementations. However, these questions are good to keep in mind when evaluating tools.

- How well does it handle data ingestion? Can it access data from all your legacy systems—including mainframes and RDBMS—as well as newer sources, such as NoSQL and streaming sources like Kafka? How fast can the tool move large volumes of data? Can it easily ingest many tables at once or is it limited to one table at a time?
- Is it truly multiplatform? Does the tool work on premises, in the cloud, or in a hybrid architecture? Do you need to purchase separate tools or licenses depending on the platform? Will it work in a Spark framework as well as a MapReduce framework? Does it do parallel processing in the cloud at all?
- Is it reusable? Can you reuse components you have designed and developed, including reusing the same components for both data extract and transformation? Can you use the same jobs and tasks across new compute frameworks without recoding?
- Is it scalable? The complexity, volume, and variety of data from legacy sources can often impose scalability-related performance issues. Although tools for data lake creation may include features such as platform compatibility, recommended and suggested performance optimization, and compression, they're not always well equipped to do so with complex legacy data. They may handle it with clumsy workarounds such as by padding out variable length files, which bogs down performance significantly.

  It's also wise to verify that the tool is scalable with respect to the equipment it will run on. Can it scale up performance from a 10-node test cluster to a 1000-node production cluster without reconfiguring or recoding?
- Does it meet compliance requirements? As addressed above, security is of paramount concern, so the tool you choose must support all major security protocols. Evaluate how the tool handles metadata. Does it preserve the data lineage from all sources, including mainframes? Is the metadata locked inside the tool or is it available to your other metadata management environments?
- Does it help you meet your success criteria? The main benefits of modern tools over hand-coding are ease-of-use and greater efficiency. Look for an easy-to-use graphical user interface and a simple approach that allows you to design jobs once and deploy them on any platform and across multiple compute frameworks. This greatly reduces skills requirements and staff time needed for coding, tuning, and adjustments over time. How many of these features does the tool you are evaluating have? Does it deliver on its promises?

If multiple vendors claim key make-or-break capabilities, a focused POC or paid pilot might be in order to put those claims to the test. An effective POC will have a well-defined scope that allows you to evaluate the vendor and supporting features in the most efficient way possible. Be sure to keep careful notes as you conduct your POC, as well as links to any other related documentation, because this will form the basis of your implementation road map later on.

NUMBER EIGHT

DEFINE SUCCESS *BEFORE* YOU BEGIN

We've covered many best practices in the prior seven checklist points. Now we leave you with the last and arguably the most important. While all successful projects are closely aligned to—and in support of—defined business objectives, creating a data lake with extensive legacy data also requires a definite set of technical objectives to measure against.

- **Outcomes:** Is all the data in the data lake? Was the data validated and checked for its metadata and interconnectedness? Was the depth of the data verified by the appropriate business teams?
- **Measures:** Were the base measures from the data lake achieved? Did they meet all the definitions of data governance? Examples of measures include data lineage reports, consistent loading and transformation time for processes, stewardship compliance, metadata definitions compliance, and security and access rules implementation and use.
- **Data quality:** Is the overall quality of data in the data lake satisfactory? Was all the necessary information from the mainframe systems correctly translated into the data lake?
- **Data access:** Have the necessary data access rules been implemented? Are users satisfied with the granularity of access and usability?
- **Data availability:** Is data available with recency and accuracy? Does the refresh cycle update data in the cycle time allotted?
- **Security:** Is security implemented with a combination of authorization, authentication, data access (accomplished in the Hadoop ecosystem by Knox & Ranger/Hortonworks, Sentry/Cloudera, and other vendors)? This is a complex subject and needs to be monitored and managed carefully.
- **Network:** Have there been performance issues? Are there network-specific requirements that need to be documented?
- **Encryption:** Is data that needs to be encrypted when stored sufficiently encrypted? Has this encrypted data been tested and validated? This topic is a learning lesson for all implementations because encryption on storage and adding access rules to the data can degrade performance. All data is never encrypted, and compliance rules can cause portions of data to be never brought to the data lake.
- **Adoption:** Has the data lake gained user acceptance? Is this the new enterprise data source for analytics? Has the governance mechanism and process been completed for this data lake?
- **Metrics:** Are operational and business analytics being run from the data lake? Are they accepted to provide more insights?

These items are a part of the post-implementation phase focus areas that provide feedback to the organization and opportunities for optimization. It should be assumed that these will need to be updated and reported on regularly for at least a reasonable period after implementation, if not for the life of the data lake. This will also allow you to measure the level of adoption and use of the data lake, and the overall success of the data lake program.

FROM OUR SPONSOR

www.syncsort.com

Syncsort is a provider of enterprise software and global leader in Big Iron to Big Data solutions. As organizations worldwide invest in analytical platforms to power new insights, Syncsort's high-performance software harnesses valuable data assets while significantly reducing the cost of mainframe and legacy systems. Thousands of customers in more than 85 countries, including 87 of the *Fortune* 100, have trusted Syncsort to move and transform mission-critical data and workloads for nearly 50 years. Now these enterprises look to Syncsort to unleash the power of their most valuable data for advanced analytics. Whether on-premises or in the cloud, Syncsort's solutions allow customers to chart a path from big iron to big data.

Syncsort offers DMX-h high-performance data integration software to simplify big data integration by making it easy to collect batch and streaming data from every source across the enterprise—from mainframe and RDBMS to NoSQL and Kafka—and populate your data lake efficiently, reducing development time from weeks to days. With DMX-h software, you can visually design your jobs once and deploy them anywhere—MapReduce, Spark, or standalone servers—with no changes or tuning required, using the data integration staff you already have.

For more information about Syncsort, visit www.syncsort.com.

ABOUT THE AUTHOR

KRISH KRISHNAN is the founder and president of Sixth Sense Advisors, Inc. He is an expert in the strategy, architecture, and implementation of big data, text analytics, and high-performance ata warehousing. He wrote Building the Unstructured Data Warehouse with Bill Inmon, and has since published two more books on social media analytics and big data.

Krish teaches at TDWI and speaks at many conferences worldwide. You can follow him on Twitter: @datagenius.

ABOUT TDWI CHECKLIST REPORTS

TDWI Checklist Reports provide an overview of success factors for a specific project in business intelligence, data warehousing, or a related data management discipline. Companies may use this overview to get organized before beginning a project or to identify goals and areas of improvement for current projects.

ABOUT TDWI RESEARCH

TDWI Research provides research and advice for BI professionals worldwide. TDWI Research focuses exclusively on data management and analytics issues and teams up with industry practitioners to deliver both broad and deep understanding of the business and technical issues surrounding the deployment of business intelligence and data warehousing solutions. TDWI Research offers reports, commentary, and inquiry services via a worldwide membership program and provides custom research, benchmarking, and strategic planning services to user and vendor organizations.