









Fundamentals of Data Engineering Chapter 2

<input checked="" type="checkbox"/> Favorite	<input type="checkbox"/>
<input checked="" type="checkbox"/> Archived	<input type="checkbox"/>
<input checked="" type="checkbox"/> Fleeting	<input type="checkbox"/>
↗ Area/Resource	
↗ Project	
▼ Type	
📅 Review Date	
📎 Image	
🔗 URL	
🕒 Created	@February 21, 2023 10:02 PM
🕒 Updated	@February 21, 2023 10:02 PM
🔍 Root Area	
🔍 Project Area	
Σ Updated (short)	02/21/2023
↗ Pulls	
🔍 Resource Pulls	
🔍 Project Archived	
Σ URL Base	
Σ 🔍 Recipe Divider	🥗🥗🥗 RECIPE BOOK PROPERTIES 🥗🥗🥗
☰ 🔍 Recipe Tags	
Σ 📖 Book Divider	📖📖📖 BOOK TRACKER PROPERTIES 📖📖📖
☰ 📖 Author	

  Date Started	
  Date Finished	
  Book Status	
  Rating	

The Data Engineering Life cycle



A core message in this book is that due to the increased technical abstraction of tools, data engineers will increasingly become data life cycle engineers, thinking in terms of data life cycle management. The data engineering life cycle is the central theme of the book.

What is the data engineering life cycle?

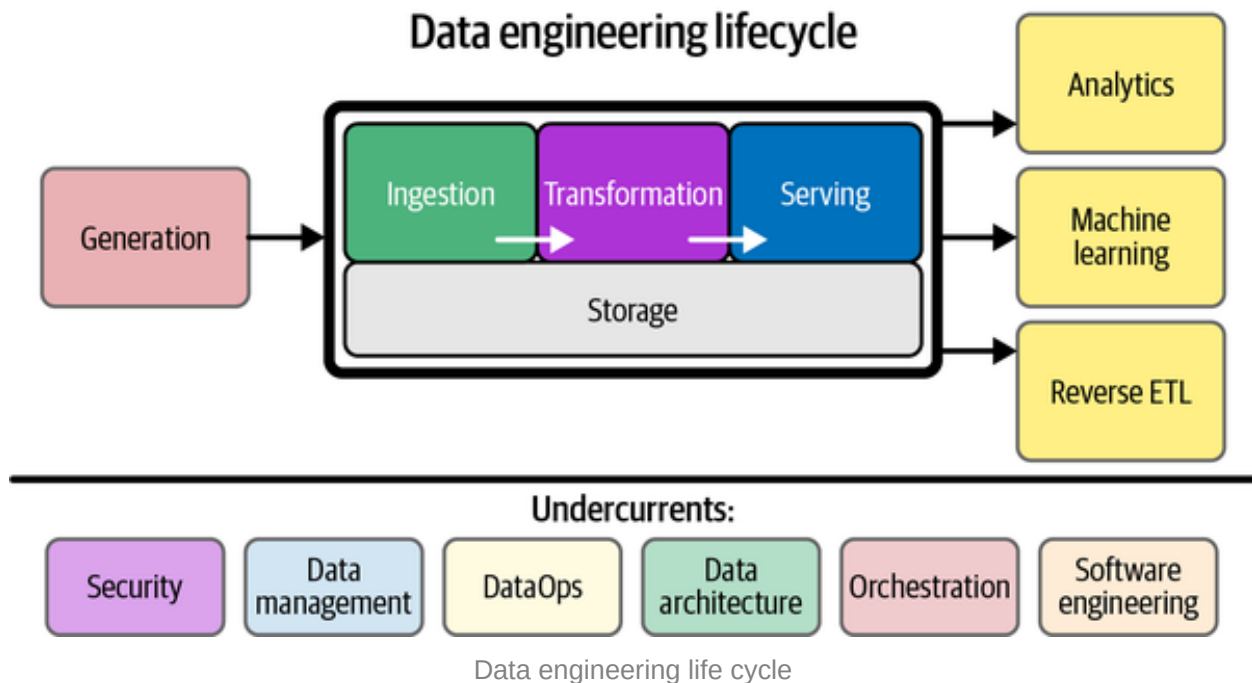


The data engineering life cycle comprises stages that turn raw data ingredients into a useful end product, ready for consumption by analysts, data scientists, ML engineers and others.

Data products



Data products can be thought of as self contained data containers that directly solves business problems or are monetized. Examples include: dashboards, ML models, a table or view.



The life cycle starts by getting data from the source and storing it. It is then transformed so that it can be ready for the final goal of serving it. Storage occurs at several stages throughout the life cycle hence it underpins the ingestion, transformation and serving stages.



The undercurrents are core components to the cycle as it cannot function properly within a business context without these undercurrents.

Difference between data engineering life cycle and the data life cycle



The data engineering life cycle is a subset of the data life cycle. It only focuses on the stages a data engineer controls.

☐ Look into the data life cycle

Stages

Generation



A source system is the origin of the data used in the data eng life cycle. The source is what captures these point in times events, this can be an smart camera, automated billing reports, etc... The data engineering should have working knowledge of how the systems work, the way they generate data as well as the velocity, frequency and variety of data they generate.

☐ Evaluating source systems

☐ Schemas

Storage



Choosing a proper storage solution is key to success in the rest of the data life cycle. Storage runs across several sections of the data pipeline, with storage systems crossing with other systems, ingestion, transformation and serving.

☐ Evaluating storage systems

☐ Understanding data access frequency

Ingestion



Once you understand the data source, the characteristics of the source system, and how the data is stored you need gather the data. The next stage is data ingestion from source systems.

Batch versus streaming



Virtually all data we deal with is inherently streaming. Data is nearly always produced and updated continuously at its source. Batch ingestion is just a specialized way of handling this streaming data by gathering it in large chunks in certain time intervals

- ☐ Separation of storage and compute
- ☐ Key considerations
- ☐ Push versus pull

Transformation



Once the data is ingested and stored it must be transformed for it to be useful in downstream use cases. Batch transformations are very popular, but we expect the popularity of streaming transformations to continue growing.

Serving



This stage deals with getting value from the data. This means something different for different users. Data can be considered valuable when it has practical purposes. This is where you can apply ML, analytics etc...

- ☐ Analytics
- ☐ ML
- ☐ Reverse ETL

Major undercurrents across the data engineering life cycle



Data engineering is rapidly maturing. Previous cycles of data engineering focused on the technology layer, the continued abstraction and simplification of tools and practices have shifted this focus. DE now encompasses far more than tools and technology, and the field is now shifting up the value chain, incorporating traditional enterprise practices such as data management and cost optimization and dataOps. These practices have been term undercurrents.

- Security
- Data Management
- DataOps
- Data architecture
- Orchestration
- Software Engineering

Security



This should be top of mind for data engineers. DE must understand both data and access security, exercising the principle of least privilege. All individuals who have access to data must understand their responsibility in protecting the company's sensitive data and it's customers.

Should know:

- IAM roles
- policies
- network security
- password policies
- encryption

☐ Expand on this

Data management



Due to technical abstraction DE are moving up the value chain to the next run of best practices. Data management is the development, execution, and supervision of plans, policies, programs, and practices that deliver, control, protect, and enhance the value of data and information assets throughout their life cycle.



Data governance refers to the process of acquiring, storing, organizing, protecting, maintaining and utilizing an organization's data assets. It encompasses all activities related to managing data throughout its entire life cycle. Effective data management involves developing and implementing policies, procedures and tools to ensure that data is protected from unauthorized access, loss, or corruption, and it is available to those who need it.

- The Data Management Body of Knowledge, is the definitive book for enterprise data management.
- Without a framework for managing data, DE's are simply technicians in a vacuum. DE needs a broader perspective of data's utility across the organization, from the source systems to the C-suite.

Facets of data management:

- Data governance, including discoverability and accountability
- Data modeling and design
- Data lineage
- Storage and operations
- Data integration and interoperability
- Data lifecycle management
- Data systems for advanced analytics and ML

- Ethics and privacy

Data governance



Data governance is a set of processes, policies, standards and guidelines that ensure the proper management and use of an organization's data assets. It encompasses the management of data throughout its entire lifecycle, including the creation, collection, storage, processing, analysis, dissemination and disposal of data.

The primary goals are:

1. Ensure data is accurate, consistent and reliable
2. Ensure data is accessible, and available to those who need it
3. Ensure data is protected from unauthorized access, loss, or corruption
4. Ensure compliance with applicable laws, regulations and industry standards related to data management and protection



Data governance engages people, processes and tools within an organization to maximize the value of their data assets in driving business value while protecting data with appropriate security controls.

Discoverability



In a data-driven company, data must be available and discoverable. End users should have quick and reliable access to the data they need to their jobs. They should know where the data comes from, how it relates to other data and what the data means.

Metadata



Metadata is data about data, and it underpins every section of the DE lifecycle. Metadata makes the data discoverable and governable. It can be divided into human generated and auto generated. Tools such as Airbnb's data portal should provide a place to disclose data owners, data consumers, and domain experts. Documentation and internal wiki tools provide a key foundation for metadata management, but these tools should be combined with automated data cataloging.

- Data portal: <https://atlan.com/airbnb-data-catalog-dataportal/#:~:text=Dataportal is a data catalog,data with the appropriate context>

Categories of metadata:

- Business metadata
- Technical metadata
- Operational metadata
- Reference metadata



Business metadata relates to the way data is used in the business, including business and data definitions, data rules and logic, how and where data is used and the data owners.

- Used to solve questions like who, what, where and how



Technical metadata describes the data created and used by systems across the DE lifecycle. Includes the data model, and schema, data lineage, field mappings, and pipeline workflows. A DE uses technical metadata to create, connect, and monitor various systems across the data DE lifecycle.

- Pipeline metadata (Produced in orchestration systems)
- Data lineage
 - Origin and changes to data, and its dependencies over time

- Schema
 - Structure of data that is stored



Operational metadata describes the operational results of various systems and includes statistics about processes, job IDs, application runtime logs etc... It is used to determine whether a process succeeded or failed.



Reference metadata is data used to classify other data. Also known as look up data. It is a standard for interpreting other data. Such as internal codes.

Data accountability



Assigning an individual to govern a portion of data. The responsible person then coordinates the governance activities of other stakeholders.

Data quality



Data quality is the optimization of data toward the desired state and orbits the question, “What do you get compared with what you expect?”. Data should conform to the expectations in the business metadata. i.e. does the data match the definition agreed upon by the business?

- A DE ensures data quality across the entire data engineering life cycle. This informs performing data-quality tests, and ensuring data conformance to schema expectations, data completeness and precision.

Data quality is defined by three main characteristics:

- Accuracy
 - Is the collected data factually correct? Are there duplicate values? Are the numeric values accurate?

- Completeness
 - Are the records complete? Do all required fields contain valid values?
- Timeliness
 - Are records available in a timely fashion?

Data modeling and Design



The process of converting data into a usable form for deriving business insights from data is known as data modeling and design. Data engineers need to understand data modeling best practices as well as develop the flexibility to apply the appropriate level and type of modeling to the data source and use case.

Data lineage



Data lineage describes the recording of an audit trail of data through its lifecycle, tracking both the systems that process the data and the upstream data it depends on. How else can we know which system affected the data or what the data is composed of as it gets passed around and transformed?

☐ Look more into data lineage

Data integration and interoperability



This is the process of integrating data across tools and processes. As we move away from single stack approach to analytics and towards a heterogeneous cloud environment in which various tools process data on demand, data integration and interoperability occupy an ever-widening swath of the DE's job. This is where orchestration comes in.

Data lifecycle management



This concerns what happens to data at the end of its data engineering lifecycle. Why discard data when you can add more storage ad infinitum. Data is stored on the cloud where you can manage its lifecycle, moving it from hot to cold data storage, and new privacy laws require data engineers to actively manage data destruction to respect user's right to be forgotten.

Ethics and Privacy



Data engineers should ensure that their data is consistent with the new rules, laws and regulations regarding their data.

DataOps



Maps the best practices of Agile methodology, DevOps, and statistical process control to Data. Data ops aims to improve the release and quality of software products. A data engineer must understand both the technical aspects of building software products, and the business logic, quality and metrics that will create excellent data products.

- DataOps borrows from lean manufacturing and supply chain management, mixing people, processes and technology to reduce time to value.

DataOps is a collection of technical practices, workflows, cultural norms and architectural patterns that enable:

- Rapid innovation and experimentation delivering new insights to customers with increasing velocity
- Extremely high data quality and very low error rates
- Collaboration across complex arrays of people, technology and environments
- Clear measurement, monitoring and transparency of results

DataOps core technical elements:

- Automation

- Monitoring and observability
- Incident response

Automation



Automation enables reliability and consistency in the DataOps process and allows data engineers to quickly deploy new product features and improvements to existing workflows.

Observability and monitoring



Look into incorporating Statistical Process control, which ensures that a process operates within its desired parameters.

Incident response



Incident response is about using the automation and observability capabilities mentioned previously to rapidly identify root causes of an incident and resolve it as reliably and quickly as possible. This enables DE's to be prepared for a disaster and ready to respond as swiftly and efficiently as possible.

Data Architecture



A data architecture reflects the current and future state of data systems that support an organization's long-term data needs and strategy. A DE should first understand the needs of the business, and translate them into a design that balances cost and operational simplicity. This means knowing the trade-offs with design patterns, technologies, and tools in source systems, ingestion, storage, transformation and serving data.

Orchestration



The process of coordinating many jobs to run as quickly and efficiently as possible on a scheduled cadence.

Airflow

Software engineering



Software engineering is still critical to data engineering.

- Core data processing code
 - SQL
 - Spark
 - Python
 - Scala
- Development of open source frameworks
- Streaming
- Infrastructure as Code
 - Know about modularity
- Pipelines as code
- General purpose problem solving