# Chapter 1 - Data Engineering a Description

# Describing the Data Engineer

Partial Definition of Data Engineering

Data Engineers conceptualise the data architecture in an organisation with the goal of creating value for data consumers in mind. They therefore need to harmonise their architectures with the structure and processes of the surrounding organisation. This means they need to understand and communicate the operative (e.g. financial) and domain-specific (e.g. client requirements) aspects and implications of how the data is being used.

## Historical Development

**1980s to present**

### [[Data Warehouse]] Engineering

Data Warehouse Engineers, ETL Developers or BI Engineers would use ETL pipelines to populate a Data Warehouse of the most important business numbers for analysts to write reports and decision-makers to make informed decisions

**1990 to present**

### Web applications emerge

Web applications establish servers, databases and storage as backend tools to power applications and software.

**early 2000s until 2010s**

### The "Big Data" and Hadoop era

More and more firms try to leverage big data amounts, running large and complicated computer clusters operating with technologies like Hadoop, MapReduce, ... In this era, many firms were frustrated by Big Data, because many Data Engineers spent their time administrating and managing the infrastructure rather than creating business value. The architecture often was too expensive, too inefficient, and too labour-intensive.

**from 2010 on**

**The modern data stack**

Decentralised, modularised, and managed data tools mostly outsourced in the cloud

Thus, the older role of a Data Engineer used to be centred around understanding "a handful of powerful and monolithic technologies" like Hadoop, Spark, Teradata, Hive and others which involved "a detailed understanding of software technology, networks, distributed systems, storage and other system-related segments", including e.g. administration of clusters and managing pipelines and transformations. In short, it was a very technical role revolving around machines and data only.

In contrast, nowadays Data engineers are less focused on the minute technical details and additionally focused on more business-related aspects. They are co-responsible for creating value out of data which involves, among others,

- communicating with and understanding data consumers
- monitoring and minimising the cost of the architecture
- documenting, explaining and improving the architecture
- Conceptualising the architecture embedded into and not separate from general business structures and practices like domains, functions, or departments

## Data Engineers as enablers

Historically, Data Engineering as a discipline of its own often stemmed partly from the need of Data Scientists to take care of data wrangling, data quality, or data ingestion and

automation first. In the wake of the 2010s Data Science hype, many firms employed data scientists and machine learning engineers to build ML models for them and deal with "big data". But as it often turned out, these data scientists spent a great deal of their time NOT working on models but instead performing tasks now subsumed under the label "Data Engineering"—namely preparing all necessary steps for analysts and data scientists to perform their actual work.

> Partial definition of Data Engineering
>
> Data Engineers perform all necessary actions, procedures, and communications to enable downstream actors to create value, such as Data Analysts, Data Scientists, Software Engineers, or Business Analysts.

The authors of the book themselves describe themselves as "recovering data scientists" in this sense.

Data Engineering often is visualised as the bottom half / the fundament of the **Data Science Hierarchy of Needs**, i.e. from bottom to top: "ingest", "transfer and save", and "explore and transform". Only from this point can the other needs be fulfilled such as "aggregate and label", "learn and optimise", and then finally "AI and Deep Learning".

# Data Engineers as providers of infrastructure and enrichers of data

Ultimately, Data Engineers provide fresh, clean, and enriched data. To do so, they usually build stable **data pipelines**, i.e. they set up these transfer and transformation processes in a way that new raw data can pour in every day or hour with predictable, stable results. They also store the processed data in a stable manner where it can be accessed directly by anyone who is authorised. Data Engineering thus provides durable, repeatable, and resilient procedures and storage—in this sense they represent the **data infrastructure** of a company. The infrastructure metaphor borrows from physical engineering where infrastructure can be roads, water pipes, or street lamps. They, too, are freely available for authorised users and are stable and dependable structures that allow users to either perform certain tasks faster (you can brush your teeth with water from the tap rather than having to fetch your own water from the well) and often safer (your utilities provider will check your tap water regularly for germs and bacteria).

> Partial Definition of Data Engineering
>
> Data Engineers build stable, dependable, and safe structures through which users can perform certain tasks faster or safer. In this sense, they are building **infrastructure** just like a civil engineer would in the form of water pipes, roads, bridges, or electricity cables.

## Software Engineering skills to ensure infrastructure quality

To ensure the quality and safety of data pipelines and ultimately the users' trust in the Data Engineers' work, they both need softer skills like subject matter knowledge or a good understanding of the concrete needs of the downstream users but also harder skills and crafts that enable them to catch errors, keep operational costs low, and make the infrastructure modular and abstract. This skill is mostly found in Software Engineering and its practices. Data Engineers not only spin up cloud instances, they also write code.

## The Data Warehouse

The other historical precedent of Data Engineers besides Data-Scientists-turned-Engineers were Data Warehouse Engineers: The datawarehouse as a concept emerged in the late 80s and was formalised in the late 80s and early 90s as a central end locus for transformed and validated data ready for analysis and value creation. Various professions like BI Engineers, ETL developers or database engineers emerged that ran the processes behind the data warehouse. They supervised the operation of the database, they wrote the code that ingested and transformed the data, and they modelled and categorised the data on a higher level. Still today, Data Engineering touches upon many of these roles and data warehousing is still a relevant concept.

> Partial Definition of Data Engineering
>
> Catering to the needs of reports and downstream consumers often includes creating or maintaining a **data warehouse** where data are stored in a format ready to use for Analysts. The extent to which data engineers are involved in more downstream processes like data transformation and modelling depends on the team setup and whether the team distinguishes between Data Engineers and Analytics Engineers. In some cases (e.g. ELT procedures), Data Engineers only ingest data into the Datawarehouse and Analytics Engineers take it from there.

## The web (application) and the cloud (infrastructure)

The internet plays a two-fold role in the development of Data Engineering: On one hand, internet-based business models were the first to produce the amount of data that merited intensive data processing and enriching as a means of value creation for a company. The companies surviving the dot-com bubble in the early 2000s were the ones who heralded the 2010s paradigm of "big data" by breaking up the hitherto mostly monolithic data architecture into more distributed clusters operated through Hadoop, map reduce and other parallel computation techniques.

On the other hand, the aforementioned separation of computation into several clusters ushered in the second role of the internet: It not only *produces* the amounts of data to be analysed but it also *processes* this data throuh the **cloud**: Storage and computation are performed on remote servers that are controlled by and operated through the internet. Data is not stored and computed on premise anymore but stored and computed through some

external service provider like AWS or Google Cloud, enabling fast scaling and allegedly reducing overhead costs for database management, catastrophe recovery, or data security.

> Partial Definition of Data Engineering
>
> Because most of today's data are either directly produced in the internet or at least stored and computed in the cloud (i.e. the internet), Data Engineers need a good understanding on how to operate and connect cloud services in their daily work.

## From batch processing to data streaming

A third aspect of how the internet influenced and shaped Data Engineering was the possibility and the necessity for data streaming: Some applications need almost real-time data processing and this extremely low-latency requirement requires special expertise to execute properly. **That still does not mean that batch processing is going away any time soon, but that data streaming is another skillset that a modern Data Engineer needs.**

## The Data Engineering Lifecycle

With the Modern Data Stack freeing up Data Engineers from having to care for minute details of their data operations, they can now take care of higher-order processes that are more directly related to creating business value: Security, data management, DataOps, data architecture, orchestration, and data lifecycle management.

The goal of Data Engineering has shifted from "who has the most data?" to "who can treat their data the best to create value with them?"

To properly describe the process from data ingestion to value-creation, the authors use the concept of the "Data Engineering Lifecycle" to outline all relevant steps. This concept will be fleshed out in chapter 2.

# Data Maturity

Since modern Data Engineering is deeply interwoven with the business practices behind the consuming organisation, Data Engineers need to pay close attention to the level of Data Maturity present in the surrounding organisation.

> Data Maturity
>
> Data Maturity describes the manner in which data is used by organisations as a **competitive advantage**. It involves
>
> 1. the **volume** of data being used in the organisation but also
> 2. the extent to which data practices are **embedded** in an organisation—the more teams regularly work with data and base their decisions on them the more data-

mature.

The authors opt for a simple, three-stage model of Data Maturity here:

| | Stage 1: Starting with Data | Stage 2: Scaling with Data | Stage 3: Leading with data |
|---|---|---|---|
| **Architecture goals** | unclear, loosely defined or no goals at all | Where to scale and where to invest is defined | Maintenance and improvement of architecture go maintain and extend its competitive advantage |
| **Acceptance and usage of data** | - low or non-existent acceptance<br>- reports lack structure<br>- How to create value from data is poorly understood | - acceptance in areas where data has created value<br>- impacted areas develop structure in their reporting<br>- Some functions in the organisation understand how to create value from data | - Teams can self-serve analyses and machine learning<br>- the company is data-driven<br>- Data creates measurable business value |
| **Data Team** | small | Growing in size, increasing specialisation into different data functions | Similar to stage 2, but bigger and more formalised |
| **Role of the Data Engineer** | - generalist, takes on multiple roles simultaneously e.g. Data Scientist or software engineer<br>- Needs to prove him-/herself through creating business value | - Has proven their value,<br>- needs to scale the initial success to more domains, functions and application contexts<br>- Needs to keep educating others in the organisation both about how to use data and how the current architecture works | - Implement checks and balances that ensure availability and reliability of data services for all users |
| **Data or feature requests** | ad-hoc, unstructured, personal | Ordered and structured, formal data procedures | similar to stage 2 |

| | Stage 1: Starting with Data | Stage 2: Scaling with Data | Stage 3: Leading with data |
|---|---|---|---|
| **Data Engineering goals** | - Get buy-in from most import stakeholder groups including management about increasing data proficiency and data culture in the organisation<br>- Define the optimal data architecture (mostly by yourself realistically speaking) focussing on how to reach business goals and how to gain competitive advantages through data architecture<br>- Identify and validate the data that support the most important business initiatives and that would work within the drafted architecture.<br>- Create a solid foundation for later Analysts or Engineers to create useful reports or models. Until they arrive you might have to prototype these reports and models by yourself. | - establishing official data practices<br>- creating scalable and robust data architectures<br>- applying DevOps and DataOps procedures<br>- building systems that support Machine Learning<br>- Document and educate | - building custom-made tools and system that use data as competitive advantage<br>- concentration on the "entrepreneurial" aspects of data like data management (including data governance and quality) and DataOps<br>- Building proper documentation: data catalogues, data lineage documents and metadata catalogues<br>- Connect strongly with software engineers, ML engineers, analysts and other roles |

| | Stage 1: Starting with Data | Stage 2: Scaling with Data | Stage 3: Leading with data |
|---|---|---|---|
| **Remarks and tips** | - To keep motivation for building data architecture high, tangible and visible results need to be achieved. This, however, raises the probability of accruing technical debt through quick wins. Draft an exit strategy for legacy solutions to not be cornered by them.<br>- Go out and talk to people instead of sitting in your isolated data team bubble. You don't want to be working on something nobody needs.<br>- Avoid custom-built solutions, unneccessary complexity, or pointless initatives. Build ready-to-use solutions! | - Keep avoiding unneccessary efforts and keep avoiding custom-built solutions where unnecessary<br>- Avoid blindly adopting architecture elements and practices that work for other firms and keep focussed on what creates value for your business.<br>- Often the crucial factor for scaling is less about technology and more about the team. Choose technologies that enable your team to do more. | - Self-complacency is the biggest threat in this phase. As soon as companies reached phase 3, they have to focus on maintenance and improvement constantly to not fall back to a previous stage.<br>- Another big threat is focussing on toy or hobby projects that do not deliver value. Just because you could doesn't mean you should. |

# Two archetypes of Data Engineers

The authors distinguish two types of Data Engineers which are—in allusion to Type A and Type B Data Scientists—Type A and Type B Data Engineers

- **Type A Data Engineer** stands for "abstraction": This role is more operative and hands-on and uses standard products, managed services and tools and avoids all too complex data architectures. They are hired first when building up data infrastructure to get started
- **Type B Data Engineer** stands for "build" and they build more bespoke architectures and solutions that are more scalable and tailored to the business case of the company. They are mostly found in data maturity stages 2 and 3 when certain fundaments have been laid already.

# Miscellaneous random pieces of knowledge

- Data Analysts and other data consumers are excellent conversation partners for Data Engineers in their user research. Keeping a close connection to the consumers of data

is the best practice to ensuring value is created.

- Thus, gauging how certain decisions will impact the business is a core skill for modern Data Engineers. They need to know how to gather and implement business and product requirements in their data products.
- Data Engineers also need to be familiar with a lot of best practices from related fields like Software Engineering, DevOps, or sometimes even Finance. Concepts like Agile, CI/CD, version control, or dependency injection are as important as minimising running costs or GDPR and other data protection frameworks. They also need to be familiar with languages like SQL, Python, bash, Java or Scala.