



# Data Engineering Introduction and Epochs

# Table of Contents

<b>Data Engineering Explained</b>	<b>5</b>
<b>The 50s and 60s: Business Intelligence is (re)Born</b>	<b>6</b>
The Birth of "Computer Science"	6
ALGOL (ALGOrithmic Language)	7
COmmon Business Oriented Language (COBOL)	8
Modern Business Intelligence Emerges	8
Hierarchical Database Management Systems (DBMS)	10
<b>The 70s and 80s: Information Engineering Comes to Life</b>	<b>11</b>
Codd Redefines Large Dat	11
Naur Returns	12
Information Engineering	13
Data Processing (DP)-driven Information Engineering	13
Business-driven Information Engineering	14
From Information to Business	15
<b>The 1990s: The Rise of the Internet and the Democratization of Data</b>	<b>16</b>
<b>The 2000s to Today: The Emergence and Mainstreaming of Big Data</b>	<b>18</b>
Big Data as a Practice Area	18





Every aspect of our lives is impacted by data, and the use of data is continually making its way into every nook and cranny of everything we do. From the standardized tests taken by school children to assess their teachers to financial scoring tools used to judge creditworthiness, data is being used to make better decisions all the time.

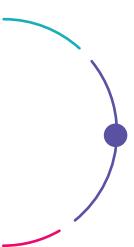
But, data has expanded beyond these obvious use cases to products and services that we use every single day. Some of these uses are critical and some are just for fun. On the critical front, as the world moves inexorably toward automated transportation in the form of self-driving vehicles, data is used to make literal life and death decisions. On the fun side, services such as Netflix and Amazon study your habits and history in order to make product and entertainment recommendations to you. For you, this use of data helps you live a fuller (if poorer!) life since there is so much cool stuff to watch and buy. For the companies that leverage this data, it's all about the bottom line. For Amazon and Netflix and the like, by making the right recommendations, they are hoping for the upsell that can help their customers make the leap to click the Buy button a few more times.

Of course, there are far more mundane—but still really important—uses of data, too. Every company on the planet does its utmost to use information to make good decisions. This data-driven decision making process has, at its core, a deep need to ensure that the company is collecting the right data.

Perhaps the biggest challenge inherent in these so-called Big Data projects is ensuring that there are underlying tools and infrastructure components – whether on-premises, in the cloud, or even in the car – to store data in ways that make sense and that enable timely analysis of that data in order to turn it into information and, ultimately, action.

For decades, information experts have attacked this problem. In fact, our current efforts around data engineering can be traced back to the 1950s and 1960s. In this paper, you will go on a journey of discovery. You will travel back to the early days of electronic data and trek through the decades all the way to today. Along the way, you will learn about the pioneers and products that have gotten us to where we are today.

So, let's begin!



# Data Engineering Explained

There are a number of disciplines that converge to enable the miracles that are achieved with data. Like so many aspects of IT, though, the data world is broken up into multiple segments, with each segment handling, in general, discrete tasks. For data, the two primary staffing segments are data scientists and data engineers. Data scientists are the wizards that can look at buckets of raw data and turn them into beautiful imagery, from which action springs forth.

But they can't do it alone. Data scientists need a cauldron in which to brew their data spells. For them, this cauldron is a conglomeration of computational and storage resources. The compute power enables number-crunching while the storage resources keep data available until such time as it's called to serve. On top of these resources, there are tools, such as databases, that help to keep data consistent and accessible by the analysis tools used by data scientists.

Of course, infrastructure doesn't manage itself. In fact, for modern data analysis needs, on-premises infrastructure requires dedicated and skilled people to maintain everything so that it remains available and usable by others. The people that are maintaining this underlying infrastructure are often referred to as data engineers. Simply put, data engineers are the experts on which data scientists depend in order to be able to work their magic.

Whereas data scientists tend to toil away in analysis tools such as R, SPSS, Hadoop, and other similar tools, data engineers are focused on the products with which those tools integrate. For example, a data engineer's arsenal may include SQL, MySQL, NoSQL, Cassandra, and other data organization services.

This clear distinction between data scientists and data engineers is really important to understand as you journey through the rest of this paper. Even though the general focus of this paper is on data engineering as a discipline, in some cases, it can be pretty tough to stay on just that topic, so you may see some elements of data science jump in from time to time. Moreover, over the years, terminology and roles have changed, so it's practically impossible to completely separate these two disciplines as we look at a history of what we now refer to as data engineering.

# The 50s and 60s: Business Intelligence is (re)Born

The Beatles, bell bottoms, and business intelligence. These trends defined the 1960s. One went on to influence our music culture, one (thankfully) died, and one has morphed over the years and has now shapes the modern business landscape.

During the 1960s, the business world was forever changed thanks to the beginnings of mainstream introduction of centralized computing systems. Never had such computational power been so concentrated. Of course, by today's standards, the systems of the 1960s can't even compare, but, at the time, these behemoth computers revolutionized how business was done and ushered in the information age.

## THE BIRTH OF "COMPUTER SCIENCE"

Although the discipline of computer science existed long before the 1960s, the emergence of mainstream computing popularized the term and it quickly became a part of the public lexicon. One of the most important aspects of traditional computer science has been the development of various programming and scripting languages. Programming languages made it possible for more people to harness the power of computing systems and wrangle data into submission. During the 1950s and 1960s, three primary computer languages were developed and brought into business: ALGOL, COBOL, and Fortran.



## ALGOL (ALGORITHMIC LANGUAGE)

If you've spent any time at all in IT or at developer parties, you've heard of programming languages such as Pascal and C. After enjoying a very long life as a learning language, Pascal has mostly fallen out of favor, but development in C continues to this day. These languages have a common progenitor in a language known as ALGOL (ALGOrithmic Language). The very first ALGOL specification was introduced in 1958, with new specifications released in 1960 and 1968.

ALGOL wasn't without its challenges. The language did not have facilities for regular input/output operations, so its usefulness for business applications was less than stellar. ALGOL was a game changer, though, in that it included syntax elements that were somewhat more natural (mathematically speaking) than were available in some other languages available at the time. One of the original dreams behind the initial ALGOL effort was to create a single computing language that could work across a variety of systems. Although this did not happen, and even though that ALGOL is now considered a dead programming language, it's spawning of a number of newer languages—some still in use today—demonstrate that some vestigial elements of the language remain relevant.



### Peter Naur (1928-2016) Syntax and Data Science Pioneer

A key contributor to ALGOL, particularly the ALGOL 60 variant, was Peter Naur. Naur's name is a storied one in the annals of data science. In fact, although he was a professor of computer science at University of Copenhagen from 1969 to 1998, Naur was a staunchly opposed to the term "computer science" and preferred the term "data science." Mr. Naur was also a key contributor to what became known as the Backus–Naur form (BNF), which help to form the design of ALGOL 60 and which is still used today to describe language syntax. These types of contributions to the industry helped to propel computing forward.

**Photo Credits:** The original uploader was Erikj at English Wikipedia - Transferred from en.wikipedia to Commons by Sozi, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=4030914>

## COMMON BUSINESS ORIENTED LANGUAGE (COBOL)

While ALGOL and other early languages often focused on needs of the scientific research community, it was COBOL that brought business-centric computing mainstream. COBOL's English-like syntax and easily identifiable data structures made it a prime tool in business process automation and data analysis.

COBOL's business-centricity is actually somewhat ironic, given the language's origins, in part, as a segment of a Department of Defense effort to modernize the military and, by extension, the government. The first version of COBOL was released in 1960, and, since then, three new standards have been released, coming in 1968, 1974, and 1985.

In fact, COBOL proved to be so popular that it remains in use even today. Although it's not generally used for new application development, millions of lines of COBOL code remain in production systems even today. In fact, although other languages were complicit, COBOL code was a big part of the concern around the "Year 2000" hullabaloo. It wasn't COBOL itself, but the underlying data that was the concern. Back when digital storage capacity used to be exceedingly expensive, to save space, developers stored years with two digits. That two bytes of data savings was a lot, and it became a pervasive practice.

One of COBOL's biggest legacies has been its representation of data layout. Businesses and government agencies used COBOL's data layout scheme for decades after COBOL was released, particularly in situations where data sharing had to take place. COBOL's data layouts were very easy to read and support.

## MODERN BUSINESS INTELLIGENCE EMERGES

We're starting our data engineering journey in the 1960s era, but, way back before the modern era of computing was even the stuff of dreams, some of the terms that we use even today were invented. Business Intelligence is one such term. Today, this now-common term carries with it an implicit assumption that there are practically limitless underlying computing and software resources that can be wrangled into submission with an end result being a beautifully represented depiction of the business' operational status and the competitive business landscape in which that business operates.

It's hard to believe that the term business intelligence actually first appears way back in 1865 in a book entitled Cyclopædia of Commercial and Business Anecdotes. Of course, over time, the way that this term is used has morphed to fit the capabilities of the day, but the general underlying intent remains intact.

In 1958, Hans Peter Luhn unleashed unto the world a paper simply entitled A Business Intelligence System. Little did he know the change that would be ushered in through the thoughts shared in this paper. The following quote is the abstract that describes the contents of the paper:

“

An automatic system is being developed to disseminate information to the various sections of any industrial, scientific or government organization. This intelligence system will utilize data-processing machines for auto-abstracting and auto- encoding of documents and for creating interest profiles for each of the 'action points' in an organization. Both incoming and internally generated documents are automatically abstracted, characterized by a word pattern, and sent automatically to appropriate action points. This paper shows the flexibility of such a system in identifying known information, in finding who needs to know it and in disseminating it efficiently either in abstract form or as a complete document.<sup>1</sup>

”

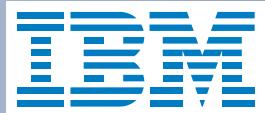
The idea espoused in the paper holds true even today. With the paper and through the subsequent development and deployment of tools and technologies that support this vision, the modern era of business intelligence was born.

<sup>1</sup>Source: <http://altaplana.com/ibmrd0204H.pdf>

## HIERARCHICAL DATABASE MANAGEMENT SYSTEMS (DBMS)

Although databases themselves existed long before the 1960's, as the information age continued to engulf more and more organizations, pioneering companies in this space came to realize that there needed to be new products brought to market to help businesses leverage the computers that were continuing their insidious penetration of the business.

Enter IBM.

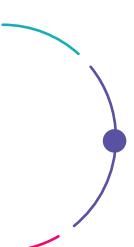


### IBM's Long and Storied History

IBM wasn't always IBM. Way back in 1888, a small company named The International Time Recording Company came on the scene. Shortly thereafter, in 1891, another company, the Computing Scale Company was founded. In 1911, both companies were merged with a third company, called the Tabulating Machine Company, as well as a couple of smaller companies, to form the Computing-Tabulating-Recording Company, which was based in Endicott, New York. In 1924, the company changed its name to International Business Machines and, in 1947, adopted the moniker by which the company goes today, IBM. Today, IBM's is based in Armonk, NY and the company serves 170 countries providing cloud computing, data center equipment, and enterprise IT consulting services.

Of course, IBM was a huge, well-known company and was recognized as a leader in the nascent computing era, but in 1969, the company extended its portfolio through the introduction of the IBM Information Management System (IMS/360) product. The 360 portion of the product name indicated that IMS ran on the IBM 360 mainframe. IMS is recognized as being the first commercially available hierarchical database management system on the market. In a hierarchical DBMS, data is organized into now-familiar tree-like structures. In such a system, each record has just a single parent record (except the root, which has no parent) but may have many child records.

Perhaps the primary problem with these early hierarchical DBMS tools is their rigidity. By their very nature, such systems are limited to one-to-many or one-to-one scenarios. In a world in which many-to-one and many-to-many relationships need to exist, one-to-many hierarchical systems only covered a subset of the needs of the business world and, while useful, newer structures eventually superseded these early efforts.



## The 70s and 80s: Information Engineering Comes to Life

Where the 1950's and 1960's created a foundation, the 1970's and 1980's built the framing for the data information engineering structure. It was during this era that business computing went mainstream. Further, some of the weaknesses in 1960's-developed structures began to undergo correction, which aided in analysis of more and larger data sets.

### CODD REDEFINES LARGE DATA

For example, in 1970, IBM's Edgar F. Codd published a paper detailing what would ultimate become the now-common relational database management system (RDBMS). The paper carried the title A Relational Model of Data for Large Shared Data Banks and opened with the following:

“

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation).

”

This was an early admission that, in order to be truly useful, data use and access had to also be simple. Ordinary users could not be expected to memorize complex data structures and should be able to interact with normal business processes in a consistent and straightforward way.



## Edgar F. Codd (1923-2003) RDBMS Innovator

Edgar F. Codd was a mathematician working at IBM during the dawn of the modern information age. Codd's 1970 paper, A Relational Model of Data for Large Shared Data Banks went on to define an entirely new industry and revolutionized databases, leading to a multi-billion dollar market and reshaping the business landscape.

After the release of the paper, Codd went on to receive the Turing Award (1981), which is awarded to "individual[s] selected for contributions of a technical nature made to the computing community". In 1994, Codd was inducted as a Fellow of the Association of Computing Machinery.

## NAUR RETURNS

In 1974, the data engineering world is also reintroduced to Peter Naur. During this year, Naur released a paper entitled Concise Survey of Computer Methods. This paper outlined a number of data processing methods and applications. Perhaps most importantly as it relates to this paper Naur uses the term "data science" a number of times throughout the paper.

Naur also defines the basic principle of data science, which helped to define the market that came after. This data science principle was:

"The data representation must be chosen with due regard to the transformation to be achieved and the data processing tools available. This stresses the importance of concern for the characteristics of the data processing tools."

Simply put, Naur said that it's important for there to be a thought process around data, from beginning to end. By understanding how data is to be used, appropriate structures can be created from the beginning that support the intended outcomes associated with capturing that data.

However, Naur was but one voice in the 1970's information revolution. Clive Finkelstein, now considered the "father of information engineering" and his work created a path to modern information architecture. Finkelstein, over a period of years, developed a series of concepts that were eventually published by Computerworld in 1981.

## INFORMATION ENGINEERING

Building on the work of Naur and Codd, among other, Finkelstein initially focused his information engineering research efforts on data analysis and database design, but the term quickly expanded to include software engineering efforts as well. Originally, however, information engineering was laser focused on data.

Over time, information engineering underwent something of a splintering, with two distinct variants emerging: data processing-driven and business-driven.

## DATA PROCESSING (DP)-DRIVEN INFORMATION ENGINEERING

The Data Processing variant of the general Information Engineering track was originally developed by Database Design, Inc. (later renamed to KnowledgWare, Inc.) and Texas Instruments and de-emphasized the data centric nature of information engineering in favor of process-centricity. In other words, rather than strict focus on data, the focus shifted to overall information systems development, which was defined to consist of four phases:

- Planning
- Analysis
- Design
- Construction

This very intentional and interview-driven development methodology—whereby data processing experts interviewed end users in an effort to glean wisdom regarding business processes—was intended to help ensure that data was captured with purpose and that there was completeness of vision and requirements definition so that the end result was a usable system. This was particularly important as more and more organizations began early efforts at automation of what were traditionally manual business processes.

## BUSINESS-DRIVEN INFORMATION ENGINEERING

Along the way, though, an organization in Australia named IES determined that DP-driven information engineering efforts were resulting in very long wait times by the business. As data processing departments pondered design the business waited... and waited... and waited. Often, wait times were long enough that, by the time data processing departments completed their efforts, the business had moved on. In 1986, the business-driven variant of information engineering was introduced in an effort to correct some of the challenges inherent in a data processing-driven focus.

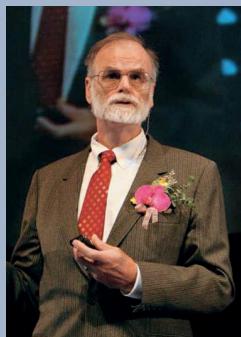
The business-driven variant required the inclusion of functional end users in the information management and development lifecycle. With a realization that business knows business and data processing/information technology knows computers, the goal was to bring the groups together to enable faster and more accurate results.

Rather than focusing on history-driven processes, the business-centric variant instead looked to an organization's strategic plans. So, rather than looking to the past to create a system, the business could them create an information system that was built for where the business was going.



## FROM INFORMATION TO BUSINESS

With all this research and development, the world had entered the information age and business was doing everything possible to capitalize on it. During the 1980's internetworking became more mainstream as commercial Internet Service Providers (ISPs) began to open up for business. Although large businesses and academic and research institutions were already connected to a common network, the commercialization of the worldwide Internet proved to be radically transformational, including where data was concerned.



### Jim Gray (1944-2007/2012) Information Theorist

Jim Gray was a researcher who originally developed three database principles—atomicity, consistency, and durability — a concept that would, in 1983, be expanded by Andreas Reuter and Theo Härder and morphed into the ACID principle upon which most relational databases today are modeled. Gray worked for such companies as DEC, Tandem, IBM, and Microsoft. Gray was also awarded a Turing Award by the Association of Computing Machinery. In 2007, while working for Microsoft as a distinguished engineer and as Microsoft's Bay Area Research Center, Gray was lost at sea and was presumed dead in 2012.

**Photo:** By Microsoft Research [CC BY-SA 3.0 us (<http://creativecommons.org/licenses/by-sa/3.0/us/deed.en>)], via Wikimedia Commons



# The 1990s: The Rise of the Internet and the Democratization of Data

The period from the 1960's to 1980 was focused on centralized control of information and computing resources. However, in the 80's, the world began to see the rapid rise of interconnected systems. Alongside the development of these various networks—both inside companies and connecting organizations to one another—was the rise of the microcomputer. No longer was computing power solely the realm of data processing, management information systems, or information technology departments. The desktop computer revolution had begun and the age of the mainframe was closing. As the 1990s approached, it was exceedingly clear that life—and business—would never be the same.

With the rise of the commercial Internet, the ability for organizations to share data and to capture new data about their customers grew exponentially. Moreover, underlying computing resources continued to evolve and became far more adept at supporting burgeoning data capture, consumption, and analysis needs.

For example, in the 1990s, the first Fibre Channel specifications were developed. While the microcomputer was waging a full-on assault of the mainframe in an effort to unlock computing resources, the opposite war was being conducted in the world of storage. Rather than continuing to support a myriad of different storage islands, businesses across the globe were looking for ways to reduce their storage-related expenses and make their data more easily accessible by more people.

Why is this important in terms of data engineering? According to IBM Systems Journal, 1996 was the year that digital storage became more cost-effective than paper data storage. So, beyond being far more accessible, and manipulable than paper, digital data also became cheaper. And, with the elimination of cost as an excuse for not storing data elements, the sky was the limit on what people were willing to do with data.

There was something else that happened in the 1990s that is important as changed the way that people view data... for better or for worse. On September 4, 1994, Jonathan Berry published an article with a very simple title: Database Marketing. This key line from his article neatly summarizes how business were turning their attention from the use of data as an internal process enhancement to using it as a way to boost top-line revenue:

"A growing number of marketers are investing millions of dollars to build databases that enable them to figure out who their customers are and what it takes to secure their loyalty."

The efforts at collecting as much data about customers didn't stop in the 1990s. Those efforts have continued to accelerate.





# The 2000s to Today: The Emergence and Mainstreaming of Big Data

In the late 1990s, John Mashey, chief scientist at Silicon Graphics (since then renamed SGI and then gobbled up by Hewlett Packard Enterprise in November of 2016) began to use the term big data in his various talks and discussions. In fact, the title of Mashey's April 1998 talk was Big Data and the Next Wave of InfraStress. In this talk, Mashey sought to educate the audience about how the impending tsunami of data would have the potential to cripple underlying infrastructure.



## Disagreement Around Origins of “Big Data”

Although the term Big Data had been used here and there prior to Mr. Mashey's usage, the late 1990s are when the term began to spread. However, attribution for truly launching what we consider the modern big data phenomenon is often ascribed to Roger Moulalis from O'Reilly Media, who used the term in 2005 to describe sets of data that were practically impossible to manage using traditional business intelligence tools. Moreover, there were other interim uses of the term that some consider just as important. So, it's left to the reader to make a final determination.

## BIG DATA AS A PRACTICE AREA

Big data practitioners seem to like the letter v. There are a number of characteristics that describe big data and, strangely enough, people have found v-words for all of them. There are articles all over the Internet discussing the “n v's of Big Data”. Sometimes, it's three v's. Sometimes, it's five. And other times, there are a whopping seven!

## Volume

As was described in John Mashey's 1998 paper, the sheer volume of data is a key characteristic of big data. Data sets used to be measured in terms of megabytes, but it's very common today to have data sets that measure in terms of gigabytes, terabytes, and petabytes. And that's just for single data sets. As you start to look at data across industries, you need to start thinking in terms of exabytes and zettabytes. The sheer volume of data creates analysis challenges that require specialized tools to solve. As time goes on the rate by which data sets increase in size also continues to accelerate.

## Velocity

As of 2016, every day, Facebook generated 4.3 billion (yes, with a b) messages, Twitter generated 500 million tweets, and, according to some estimates, 2.5 exabytes of new data was produced. It wasn't all that long ago when these estimates were an order of magnitude smaller and it won't be long before they're an order of magnitude larger. This second v outlines the characteristic of big data that demonstrates that the amount of data captured continues to increase each and every day. That creates a new set of ways to store this data, which impacts storage and networks alike. It also means that data engineers and scientists need to find ways to process and analyze data far more quickly.

## Variety

At the same time that you're getting more data more quickly, you're also getting all kinds of different types. Recently emerging trends, such as the Internet of Things (IoT) will just exacerbate this situation. There are mass quantities of unstructured data hitting corporate databases each and every day.

## Variability

The data that you collect today may not have the same meaning tomorrow. Ensuring that you're able to assign consistent meaning to that data, even as underlying conditions change, is increasingly necessary.

## Veracity

You've heard the phrase "garbage in, garbage out". The next v in the big data realm stands for veracity and its intended to describe a characteristic that requires that data be valid. In other words, for big data systems to be reliable and usable, you need to be able to ensure that the data is also accurate.

## Visualization

People like pretty things. Today, there are entire industries that revolve around turning raw data elements into beautiful images that the viewer can immediately grasp. Whether that's through the use of charts, infographics, or spreadsheets, the ability to make the underlying information visually accessible has become a critical component of the big data movement.

## Value

The end goal for any big data undertaking is to drive some kind of value for the business. Whether the initiative is intended to streamline processes and reduce costs or it's intended to grow revenue, big data projects, like every other business project, need to be able to show some kind of return on investment.

Panoply.io provides end-to-end data management-as-a-service. Its unique self-optimizing architecture utilizes machine learning and natural language processing (NLP) to model and streamline the data journey from source to analysis, reducing the time from data to value as close as possible to none.

© 2017 by Panoply Ltd. All rights reserved. All Panoply Ltd. products and services mentioned herein, as well as their respective logos, are trademarked or registered trademarks of Panoply Ltd. All other product and service names mentioned are the trademarks of their respective companies. These materials are subject to change without notice. These materials and the data contained are provided by Panoply Ltd. and its clients, partners and suppliers for informational purposes only, without representation or warranty of any kind, and Panoply Ltd. shall not be liable for errors or omissions in this document, which is meant for public promotional purposes.