DATA SCIENCE MEI/1

University of Beira Interior
Department of Informatics

Hugo Pedro Proença,
hugomcp@di.ubi.pt, **2021/22**

Data Infrastructure for **Data Science**

- This is the software infrastructure that allows to both store and access a company's data that **is needed from the start.**
  - Typically, that's what **Data Engineers** do: they build and maintain consistent the data infrastructure, guaranteeing its availability to **Data Scientists**, who will actually analyze it and make it useful to a company.
- Generally speaking, Data Engineers are particularly needed in the early stages of the Data Science project.
  - It is important to keep scalability in mind. If the project is supposed to grow, then the whole architecture should be scalable.
- Often, Data is stored on multiple servers, which creates challenges for engineers to integrate data so that it may be analyzed properly. Data processing is a challenge as powerful computers, programs, and a lot of preparatory data engineering works are required to create the appropriate "data sets".

Data Infrastructure for **Data Science**

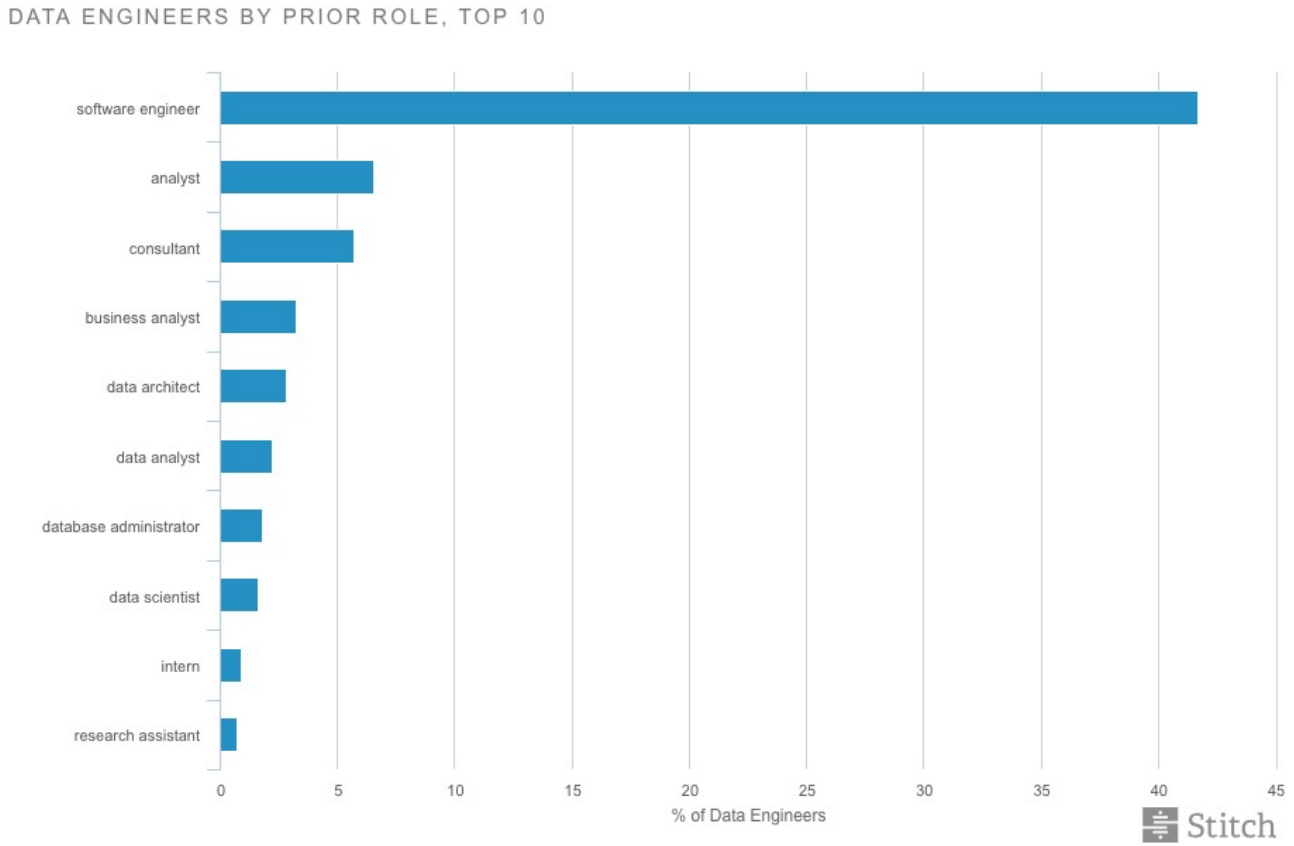
- In summary, the task demanded to a **Data Engineer** is to prepare the "Data Sets" to be used by Data Scientits and Data Science processes.
  - In this sense, the tasks carried out by the Data Engineer at this stage may (or may not) be considered part of the Data Science chain itself.
  - Proficiency in Database Solution languages, especially **SQL**, Programming Languages like **Python**, **R**, **C++, Hadoop** are some of the skills that a Data Engineer should have

Data Engineer: Main Tasks

- **Build and maintain the organization's data pipeline systems**:
  - Data pipelines encompass the journey and processes that data undergoes within a company. Data engineers are responsible for creating those pipelines.
  - Creating a data pipeline may sound easy or trivial, but at big data scale, this might imply bringing together 10–30 different big data technologies.
  - More importantly, a data engineer should be the one who understands and chooses the right tools for the job.
- **Clean and wrangle data into a usable state:**
  - Data engineers make sure the data the organization is using is clean, reliable, and prepared for further analsis, i.e., support queries run against it by data scientists.
  - Data wrangling is about taking a messy or unrefined sources of data and turning then into something useful. Once the data is intially parsed and cleaned, the data sets are usable. Typically, a report is produced, derscribing the features available (scales, type of data, domains, missing values,...)

Data Engineer Background and Prior Job

- **Data engineering** is a specialization of software engineering. The majority of data engineers working in the field used to be software engineers.
  - Due to the high demand, other roles such as **DBAs**, system admins, data analysts and scientists are also transitioning.



Source: https://www.stitchdata.com/resources/the-state-of-data-engineering/?thanks=true

Data Engineer Skills

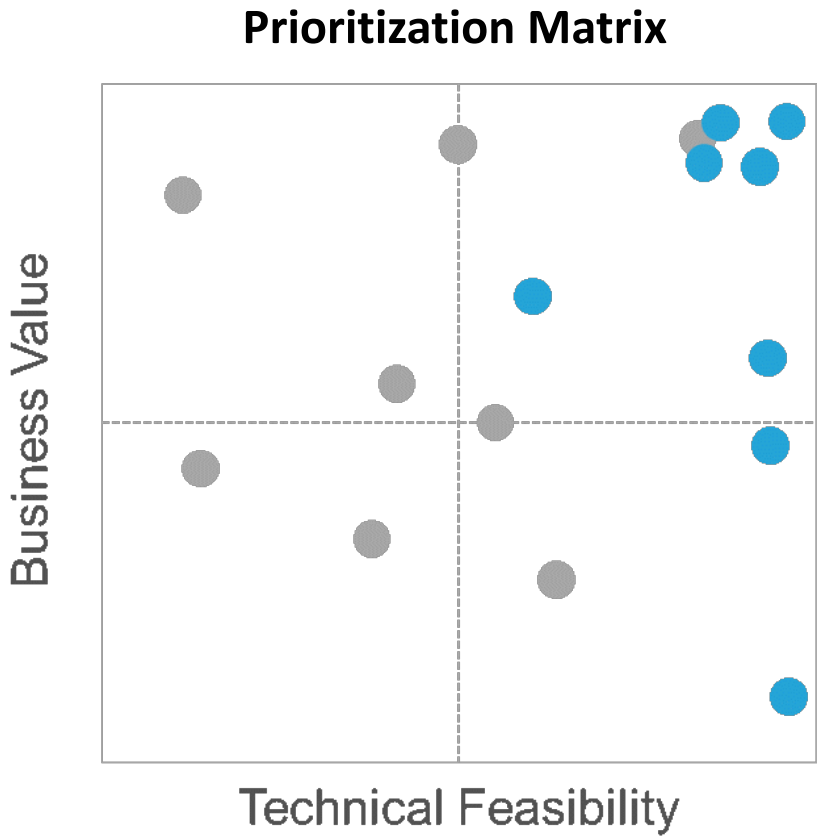
- ☐ **Linux,** command line and scripts creation/execxution/debugging
- ☐ Relative experience programming in at least **Python** or **Scala**/**Java**.
- ☐ Deep understanding of relational and non-relational databases (**SQL**).
- ☐ Some understanding of distributed systems in general and how they are different from traditional storage and processing systems.
- ☐ Deep understanding of the ecosystem, including ingestion (e.g. **Kafka**, **Kinesis**), processing frameworks (e.g. **Spark**, **Flink**) and storage engines (e.g. **S3**, **HDFS**, **HBase**, **Kudu**).
- ☐ Know how to access and process data.
- ☐ A holistic understanding of data is also important. This means thinking and acting not only like an engineer and (the supplier of the goods (data), but also as the data scientists (the data clientes). It is importante to anticipate their requirements and prioritties.

Data Infraestructure: Define the Data and Analytics Strategy

- Before tackling any kind of project, consider questions like:
  - Do we have a data and analytics strategy?
  - What is the company's overall corporate strategy?
  - What is the business reason behind the need for analytics?
- It will be important to define what technology, processes, and people will be used to put in place, in order to meet the analytics goals.
- **Understand Context:**
  - What is the long-term analytics vision, and how does it fit into the overall business strategy of the company?
- **Capture the Current State:**
  - This includes interviewing stakeholders, evaluating data sources, and reviewing technologies
- **Develop an Analytics Plan:**
  - This is a detailed plan that provides a full roadmap between the current and final states.
- **Deliver Results:**
  - Results (even if intermediar) should be delivered in a phased approach, with short cycles, so that the clients (Data Scientists) can provide feedback throughout the process and check the effectiveness of data along the way.

Projects Prioritization

- Without prioritization, the different projects may take turns never intended. Well-communicated priorities help to align projects and programs to its strategies.
  - This uncreases the success rates of strategic projects.
  - Increases the alignment and focus of management around strategic goals.
  - Clears any doubts for the operational teams when faced with decisions.
  - Builds an execution mindset and culture.

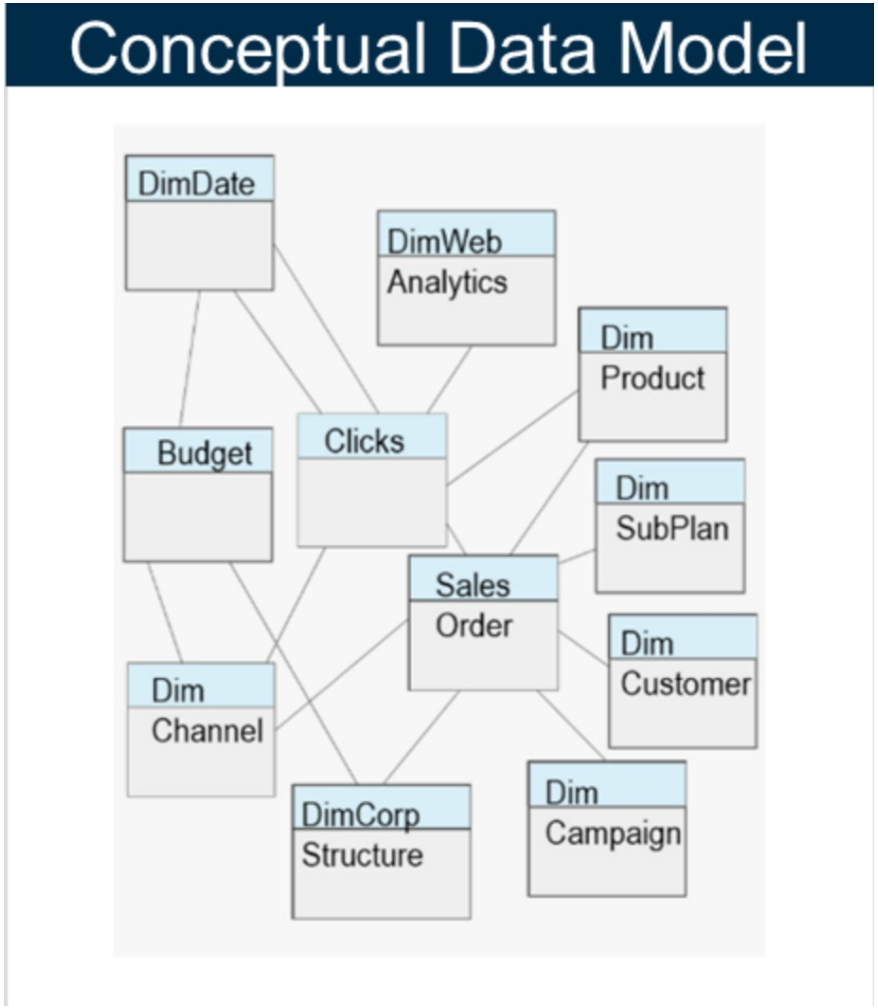


Evaluate Environments

- ☐ Where within the technology stack the environments Setup is needed the most? Consider how to move data through the stack.
  - ☐ The whole system will run smoother if this is set up well.
- ☐ Some things you should start documenting when evaluating your environments include:
  - ☐ Security setup considerations
  - ☐ Data load/storage strategy
  - ☐ Architecture diagram
  - ☐ Change management strategy
  - ☐ Ensure your environment is setup thoughtfully.
  - ☐ Look for redundancies: Make sure the system is eficiente.
  - ☐ Evaluate your environment: Consider what's best for your organization (on premises versus the cloud, etc)
  - ☐ Will multiple environments be needed? Do we have *Dev*, *QA*, and *Prod* environments, or is that overkill?
  - ☐ Refresh data: in case of distributed data source systems, it is needed to ensure the data is refreshed, so that updated versions are at each node

Build Flexible Data Models

- A data model creates the structure where the data lives.
- It determines the model flexibility and the final system performance.
- It also defines how things are labeled and organized, which determines how the data can be used and the kind of information possible to be inferred.
- Finally, a data model helps to define the problem, enabling to consider different approaches and choose the best one. (the Data Scientist might have an opinion here).



Document Data Lineage

- This phase is probably the most tedious one, but still is necessary.
  - Without the knowledge of how the data goes from origination to its destination, one could end up having to rebuild things later.
- By producing appropriate documentation...
  - We provide knowledge about what data is available, its quality, and correctness
  - We provide knowledge from the head of the ETL developer
  - We provide more transparency about what's going on with the data.
  - We give users more detail about what they're using in their reports.
  - We Understand the impact of changes made on a source system.
- This phase is similar to the creation of a ohysical model of a database, but with extra details

Column Name	Datatype	Description	Unknown	SCD Type	Schema	Table Name	Source Field Name	Leagacy Table	Legacy Source field	ETL Rules (If blank
EmployeePerson_PK	int	Auto-generated row ID number	-1							
EmployeeNumber	int				Staging	Infor_PAEmppos	Employee	Prod_Employee	person_Id	
Department Name	varchar			Type2	Staging	Infor_PRSystem	r_name	Prod_DepartmentTemp	Dscr	Join to Infor_PAEm
Department Code				Type2	Staging	Infor_PAEmppos	Process_Level	Prod_EmployeeOccupation	Dept_No	
Division Name	varchar			Type2	Staging	Infor_DeptCode	r_name	PROD_Payroll_Center	Dscr	Join to Infor_Emplo
Division Code				Type2	Staging	Infor_PAEmpPos	Department	prod_Employee	org_unit_id & PR_Center_Id	
Union Name				Type2	Staging	Infor_Pcodes	Description	prod_BargainingUnit	Dscr	Join to Infor_Emplo
Union Code	int			Type2	Staging	Infor_PAEmpPos	Union_Code	prod_Employee	Bargaining_Unit_Id	
WorkingJobTitle	int	WinAD maintained title		Type2						Need to use LDAP c
CityPositionName					Staging	Infor_PAPosition	Description	Prod_Occupation	Title	Join to Infor_Emplo
CityPositionCode	int	Generic Union assigned title		Type2	Staging	Infor_PAEmpPos	Position	Prod_EmployeeServiceHistory	Occ_Code_Id	For Prod_Employee
EmployeeStatus		ex: a5="Active Classified", t1 "Terminated"		Type2	Staging	Infor_EmStatus	Description	Prod_Employment_Type	Dscr	Join to Infor_Emplo
EmployeeStatusCode	int			Type2	Staging	Infor_HRhistory	Emp_Status	Prod_Employee	EMPLOYEE_TYPE1_ID & EMPLOYEE_TYPE2_ID & EMPLOYEE_TYPE3_ID & EMPLOYEE_TYPE4_ID	

Assess Performance

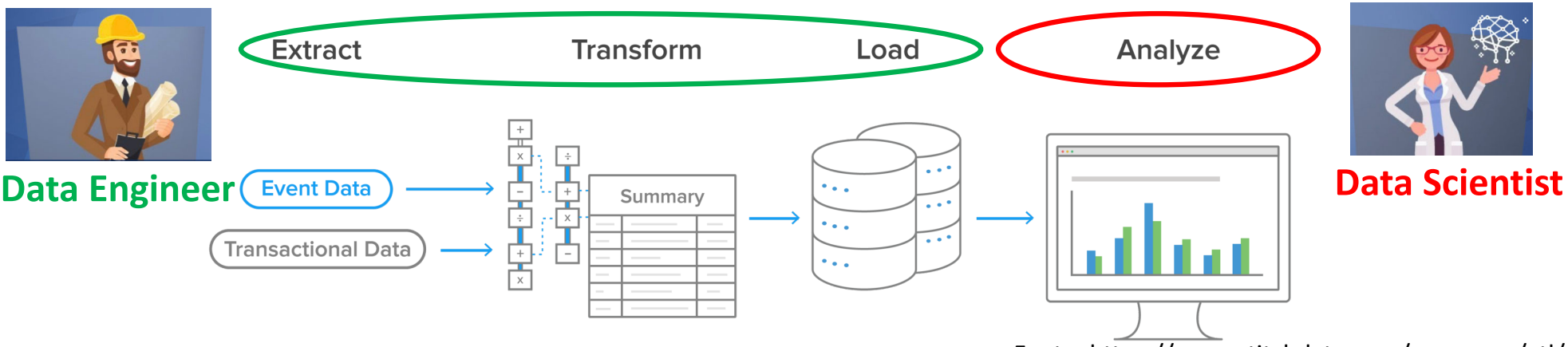
- Once the model has been defined and documented, it is time to perform the first performance assessments on the data source:
  - We want to consider performance needs for both front-end user experience and backend infrastructure
    - Taking time to do this doing the development process will help ensure optimal performance.
  - Here are some questions you can ask when assessing performance.
    - **User Experience: (From the Data Scientist Persoective)**
      - How long does it take to run reports?
      - What factors are affecting performance?
      - Are those services really too expensive?
    - **Backend Performance: (From the Database Administrator Perspective)**
      - How often does the data need to be refreshed?
      - Are you using incremental loads?
      - Are you uploading data that nobody uses?
      - How is ETL performance?

Design/Implement a Data Governance Plan

- With a properly implemented data governance program, we can gain consistency, get faster time to delivery, lower the maintenance needs, get more quality data, increase user adoption.
- It's a critical piece to your data and analytics solution, but one that is often overlooked.
  - Typically, it is build from a top-down persoective: it is importante to define who wil have "Administration" previliges within the organization and then broadcast the plan across all persons that will interact with the data, assuring that it will mantain consistente.
    - Are there any downtime periods, for mantainance expected?
    - Are there regular backup/mirroring processers schedules, that possibly will decrease the performance of the whole system at some times?
    - Who can access to which part of the data sources?
    - Is the Data governance plan according to Data Protection National Reulations?

ETL – Extract, Transformation and Load

- This is a general procedure of copying data from one or more sources into a destination system which represents the data differently from the source(s) or in a different context than the source(s). The ETL process became a popular concept in the 1970s and is **often used in data warehousing**.
- "**Extraction**" involves extracting data from homogeneous or heterogeneous sources;
- "**Transformation**" processes data by data cleaning and transforming them into a proper storage format/structure for the purposes of querying and analysis;
- "**Loading**" describes the insertion of data into the final target database such as an operational data store or a data warehouse



Fonte: https://www.stitchdata.com/resources/etl/

ETL – Extraction

- This step envolves copying data from one or more source systems. In many cases, it is the most important aspect of ETL, since extracting data correctly sets the stage for the success of subsequent processes.
- Most data-warehouses combine data from different sources.
  - Each separate system may also use a different data organization and/or format. Common data-source formats include relational databases, XML, JSON and flat files.
- An intrinsic part of the extraction process involves data validation to confirm whether the data pulled from the sources has the correct/expected values in a given domain (such as a pattern/default or list of values).
- If the data fails the validation rules, it is rejected entirely or in part. The rejected data is ideally reported back to the source system for further analysis to identify and to rectify the incorrect records.

ETL – Transform

- In the data transformation stage, a series of rules or functions are applied to the extracted data, to prepare it for loading into the end target.
- An important function of transformation is data cleansing (filtering), which aims to pass only "proper" data to the target.
- In other cases, one or more of the following transformation types may be required to meet the business and technical needs of the server or data warehouse:
  - Selecting only certain columns to load: (e.g., avoiding null columns)
  - Translating coded values: (e.g., if the source system codes male as "1" and female as "2", but the warehouse codes male as "M" and female as "F")
  - Encoding free-form values: (e.g., mapping "Male" to "M")
  - Deriving implicit values: (e.g., sale_amount = qty * unit_price)
  - Joining data from multiple sources (e.g., lookup, merge) and deduplicating the data

ETL – Load

- The load phase loads the data into the end system, which can be any data store including a simple delimited flat file or a data warehouse.
- This is the most varying phase:
  - Some data warehouses may overwrite existing information with cumulative information; updating extracted data is frequently done on a daily, weekly, or monthly basis.
  - Other data warehouses may add new data in a historical form at regular intervals — for example, hourly.
  - More complex systems can maintain a history and audit trail of all changes to the data loaded in the data warehouse.[6]
- As the load phase interacts with a database, the constraints defined in the database schema — as well as any existing“triggers” activated upon data load — apply (for example, uniqueness, referential integrity, mandatory fields).