NOVA IMS
Information Management School

MAA

Mestrado em Métodos Analíticos Avançados
Master Program in Advanced Analytics

Internship report at SDG Consulting Group

Filipa Delgado Bizarro Cerqueira

Business Intelligence: Development of a performance monitoring dashboard in a pharmaceutical company

Internship report presented as partial requirement for obtaining the Master's degree in Advanced Analytics with a specialization in Business Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

BUSINESS INTELLIGENCE: DEVELOPMENT OF A PERFORMANCE MONITORING DASHBOARD IN A PHARMACEUTICAL COMPANY

by

Filipa Delgado Bizarro Cerqueira

Internship report presented as partial requirement for obtaining the Master's degree in Advanced Analytics with a specialization in Business Analytics

Advisor: *Flávio L. Pinheiro*
External Advisor: *Vincent Benet*

September 2021

DEDICATION

This report is dedicated to my family, in particular to my parents Alexandra and Carlos, for always giving me the motivation and support to go after what I want.

‘Listening to the data is important... but so is experience and intuition. After all, what is intuition at its best but large amounts of data of all kinds filtered through a human brain rather than a math model?’

- *Steve Lohr*

ACKNOWLEDGMENTS

First, I would like to thank my thesis advisor, Flávio L. Pinheiro, for his availability in providing quick feedback and assurance during the whole process. His teachings and advice have been essential for the completion of this report.

I would also want to express my gratitude to everyone at SDG for creating an incredible environment to work in. Specifically, José Bagina, for believing in my capabilities and giving me my first work experience in the Business Intelligence field. Additionally, to David Charles, Marc Farre, and Vicent Benet, for onboarding me on the team and being accessible to any doubts and questions I may have had.

I would also like to acknowledge my family. My parents Alexandra and Carlos, for their unconditional support and valuable bits of advice. My sisters, Rita and Francisca, and grandmothers Anabela and Alzira for their presence in my life. And my uncles Zé and Vera for graciously giving me a place to stay when I first moved to Lisbon. This would not have been possible without all of you.

Lastly, I would like to thank my friends. Both my college friends that helped me truly enjoy this time and make the best of it. As well as my hometown friends Cláudia, Gena, Inês Damasceno, Inês Faro and Rita, for their long-lasting friendship and continuously being there throughout all steps in my life.

ABSTRACT

Business intelligence is a concept that has been around for over 150 years. Although, only for the past 10 years has it become an idea wildly known in the technology world. Now, it is a tool used to help organizations handle big amounts of data and transform it into real-time information and ultimately make better and more effective decisions.

In this context, the report describes my internship experience working as a Business Intelligence consultant at SDG. Which is an established consulting firm with offices in almost every continent. SDG helps companies handle their most important challenges as well as discover new opportunities with the use of advanced analytics and data-driven business models.

The report specifies one of the projects that was accomplished during my time at SDG, with the role of developer. The goal of this project was to develop a dashboard using Qlik that would help two pharmaceutical companies measure the performance of a new drug. To do so the necessary data sources were provided from both sides of the company, and using an ETL approach, the data was integrated and ready to be used in the project. The companies provided an initial mock-up of the expected visualizations, and with some changes considering the initial discoveries, the data model was created. Following, the KPI's were defined and implemented in the dashboard's visualizations. The project was deemed successful, and it fulfilled the clients' expectations.

KEYWORDS

Qlik Sense; Business Intelligence; SDG; SQL; Data Visualization; Dashboards

INDEX

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

BI	Business Intelligence
SDG	Strategy Decision Governance Consultancy
IMS	Information Management School
OLPT	Online Transaction Processing
OLAP	Online Analytic Processing
ETL	Extract, Transform, Load
Pharma	Pharmaceutical
Config	Configuration
Dev	Development
EAMO	Engineering Asset Management Organizations
US	User Stories

1. INTRODUCTION

Business Intelligence has become more than essential for all companies, particularly for the Pharma Industry. Considering the nature of the business and how much time, money and research goes into each product's launch, it's extremely important to make smart decisions that will help accelerate the growth of the company. A BI implementation will provide the information and justifications on what steps to take.

SDG is a European leader in the business performance management and knowledge management field. The firm's mission is to transform businesses with its in-depth analysis and cutting-edge technology. Alongside this, their innovative solutions and impressive client repository are the main reasons I decided to conduct my internship in this company.

This report is part of the Data Science and Advanced Analytics Master program, with a specialization in Business Analytics carried out at Nova IMS. In it, I detail one of the projects I worked on as a developer.

The project consisted in the creation of a dashboard that measures the performance of a new drug that resulted in a collaboration of two of the biggest pharmaceutical companies. SDG's expertise focuses on extracting values from the data provided and displaying it in a way for clients to better understand it and make smarter decisions. As with any new product, it is always necessary to closely monitor how the market is reacting to it. For this purpose, a dashboard was built to ensure the initial sales and order numbers are fulfilling the planned objectives.

The project goals consist of identifying the product's growth over time, compare its performance between the two companies, territory analysis, visualize the current and forecast sales, and evaluate the percentage of sales by category (face to face, calls, meetings, …).

The paper presents my experience as a Business Intelligence Consultant, describing the main tasks I was assigned to and the limitations and challenges I faced during this time. Which involved not only creating the displayed visualizations but also all the back-end part. Including, gathering and cleaning the data, constructing the data model, and creating the measures.

Therefore, the following chapters are expected to answer these business questions:
1. How can business intelligence be used to retrieve information?
2. What are the necessary steps that go into creating a dashboard?

2. COMPANY DESCRIPTION

2.1. SDG GROUP

SDG Group, which stands for Strategy Decision and Governance, is a consulting firm specialized in Business Analytics, Corporate Performance Management, and Advanced Business Architectures.

The group was created in 1994, and ever since then, it has had recognition for its innovation and development of planning and controlling methodologies and practices. As well as the design and implementation of Decision Support Systems[1].

Currently, SDG is present around the world, having 23 offices in Europe, North Africa & Middle East, and in both North and South America, employing more than 750 consultants.

The firm has over 400 active clients, and although it operates in many different industries it strives in Healthcare & Life Science. It distinguishes itself from other firms by understanding the industry-specific needs and its competitive nature. It has a close relationship with pharma institutions to continue to understand more about the business and provide the best practices and most tailored solutions.

SDG's focuses on six main key points. **Sales**, sales analysis is essential to determine the current approach effectiveness. It can provide information such as the number of new customers acquired by each salesperson, sales by product, customer segment or territory, and much more.

Medical, which is similar to sales, but because of privacy reasons does not show the sales and campaign data. It is more focused on medical performance. **Market access**, which covers Key Account Manager (KAM) and Market Access Responsible needs. KAM refers to the role of retaining important customers and nurturing their relationship, and Market Access ensures patients get access to the right treatment/product at the right price.

Marketing, where data from different channels is gathered to identify, anticipate, and provide solutions for customer requirements. **Customer Services**, where Customer feedback is collected, from surveys, and analyzed to discover valuable insights and better understand the client's wants and expectations. Ultimately, resulting in higher customer satisfaction and retention. And finally, **Compliance**, which analyzes and monitors data to ensure compliance rules are being followed.

Currently, SDG has proven its expertise in maintenance, support, operational, and evolution of reporting, with hundreds of Qlik projects delivered. The firm has earned the status of 'Certified Partner' as an Elite Solution Provider & Global System Integrator of Qlik solutions.

[1] https://www.sdggroup.com/pt-pt/about-us/sdg.

These accomplishments made 'Market Scope for Business Intelligence and Information Management Services in Western Europe' quote SDG's clients praising its ability to understand business requirements, flexibility, and strong technical competencies.

They have had a sustainable growth in profit, amounting to around 65 M € in gross sales. Additionally, the company promotes a good environment between workers, with activities such as Weekly Meetings, Happy-Hour, SDG Socializing, and Workshops. During the pandemic remote work was applied, however, the company managed to adapt almost seamlessly, and these social events continued over Zoom.

2.2. Daiichi-Sankyo

This internship focuses on the creation and maintenance of a dashboard regarding one of SDG's clients, Daiichi-Sankyo.

Daiichi Pharmaceutical Co. was founded by Dr. Katsuzaemon Keimatsu and five others, in 1915 in Takatsuki, Japan. Gradually expanding to other Japanese locations, as well as Beijing, China.

Sankyo was founded in 1899, through a joint investment by businessmen Matasaku Shiobara, Shotaro Nishimura, and Genjiro Fukui. They first established a plant in Shinagawa, Japan, and eventually expanded to the US and Europe.

In 2005, the two companies merged. Now, Daiichi-Sankyo operates in more than 20 countries around the world, has around 16,000 employees and over 100 years of scientific and technology expertise. Allowing them to define themselves as a 'drug discovery-oriented company'. For their 2025 Vision, they expect to become a 'Global Pharma Innovator with a competitive advantage in oncology.'

They are focused on improving the quality of life around the world by creating innovative pharmaceuticals. They expect to respond to unmet medical needs, provide stable supply products, improve access to pharmaceuticals and ESG (Environmental, social, and governance) activities & respond to other requests. Currently, Daiichi-Sankyo considers its three main strengths[2]: Science & Technology, Global Organization & Talent, and its Presence in Japan.

The company is focused on in-house drug discovery, made possible by its cutting-edge science & technology and collaboration with academic institutions.

Additionally, it strives on providing an environment for optimal human resources by having stretch goals, challenging tasks, and opportunities for overseas transfers and study programs.

Finally, Daiichi-Sankyo is proud of the trust and relationship it has created in the Japanese market. It has a business model for sustainable growth that focuses on launching innovative products and acquiring licenses for promising ones, having superb sales capabilities.

Currently, the organization uses SGD services to perform detailed analyzes and reporting.

[2] https://www.daiichisankyo.com/about_us/mission-strength/strengths/

3. LITERATURE REVIEW

This section provides relevant theoretical background applied during the internship. In this matter, an introduction to Business Intelligence, Data Modelling, and other interconnected concepts are presented below.

3.1. BUSINESS INTELLIGENCE

Despite the term Business Intelligence having been around since 1989, there is still not a consensus on a unique definition. In fact, it is many times described as an umbrella term. But it can be typically defined as a 'data-driven Decision Support System that combines data gathering, data storage, and knowledge management with analysis to provide input to the decision process' (Solomon & Gray, 2008).

This can be translated, in most cases, as the collection of data from different sources and its transformation into useful insights presented in a user-friendly way, such as dashboards. Ultimately, creating value and support better, faster, and more accurate decisions.

3.1.1. BI COMPONENTS, REASONS, AND BENEFITS

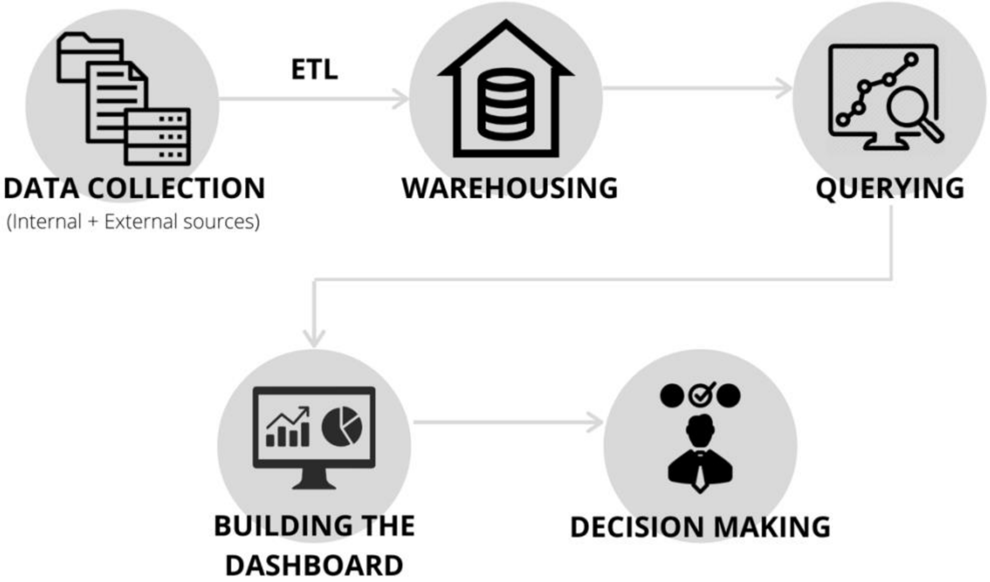


Figure 3.1: Business Intelligence process

In figure 3.1, some of the BI necessary components are shown. Before implementing a BI solution, it is necessary to have the data stored. Typically, this is done in a data warehouse, where data from multiple sources is combined in a form ready for integration, cleaning, aggregation, and querying.

Afterward, the data can start to be analyzed. Considering the business requirements and the initial discoveries, the BI team can start creating queries and measures. Different tools can be used to manipulate and mine the data. Once finished, the next step is to start building the user interface, used to visually display the KPIs and metrics produced. The interfaces are typically dashboards, reports, or online portals. When properly constructed, they allow the data to tell a story. Finally, the ultimate goal of BI can be completed, to make decisions from the information that has been given.

As mentioned, the demand for BI has increased tremendously over the past few years. Several factors have had an impact on that. First, with the era of globalization and easy internet access. There has been an overwhelming amount of information being continuously created. This, however, has not been a synonym for better information since only a small part is transformed into actual knowledge.

Organizations are now working across countries, and sometimes even industries. So, the language and format (structured or unstructured - e-mail, social media, …) can be different. Without an effective way to successfully integrate the data, and from it retrieve information, this can be an almost impossible mission.

Additionally, it's not just the volume of data that has changed, the speed at which companies receive information and are expected to respond to it has also increased excessively. So, the BI solution needs to help reduce the time in which decisions are made. Finally, the progress made in information technology has resulted in effective tools and processes that make BI more useful for companies.

Nowadays, BI can help companies with these new arising challenges, and it can improve the overall performance of a company. It allows for stakeholders to not have to go through all the data, but only the relevant extracted information. Which is displayed in a user-friendly way. Also, BI makes it possible to have real-time data shown, so faster decisions can be made.

In more detail, a successful Business Intelligence implementation brings organizations very valuable benefits. It provides access to new insights. Determining and analyzing important KPIs provides necessary information about the company's strengths and weaknesses.

Moreover, by better understanding customer behaviors and patterns, companies can tailor their approach to anticipate customer needs and wants. Ultimately increasing customer satisfaction. Additional to the analysis of internal data, companies can gain insights into their competitors by using external data to understand their positions. The integration of data across all departments also allows for a more holistic view of the company.

Finally, BI can increase revenue, and profit margins. Although this is caused by all the points made above, an increase in revenue is expected since companies have more information about themselves, and the market they are in. Additionally, when the correct metrics are provided, it is easier to know what needs to be improved.

All these benefits contribute to the main goal of a BI implementation, to make more accurate and informed decisions. Data from multiple sources are combined and converted into intuitive dashboards. By presenting this information using visual reporting, efficiency and visibility are improved. This way only the relevant information is highlighted, so it takes less time to understand it. Allowing for a better decision process, made easier even for less tech-savvy users.

3.1.2. Successful BI Implementation

A lot goes into guarantying a successful business intelligence implementation. “60% of BI projects failed due to inadequate planning, poor project management, undelivered business requirements or those that were implemented were of low quality” (Moss & Atre, 2003). Due to this high risk, it is extremely necessary to make sure the implementation is correctly done.

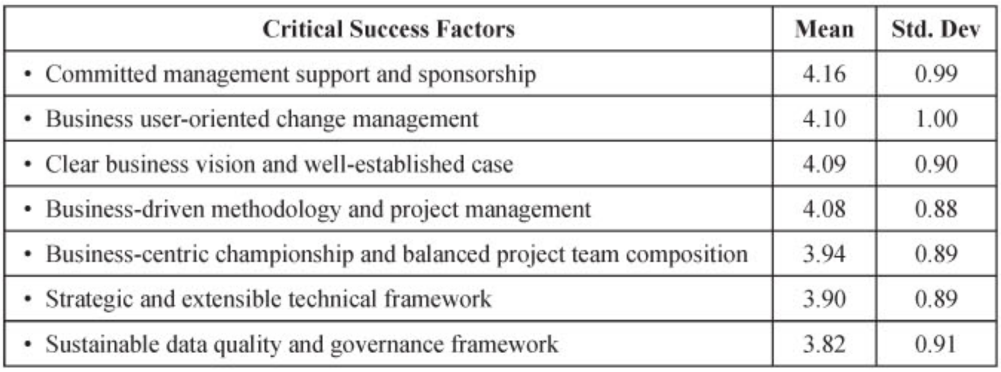
In the paper ‘Managing the Implementation of Business Intelligence Systems: A Critical Success Factors Framework’ done by Yeoh, Koronios & Gao (2008), seven critical success factors (CSF) were rated with the intention of better understanding what influences the success of a BI implementation.

To rate those critical success factors the Delphi method was used. This is a technique that intends to create more informed decision-making by relying on the consensus of the opinion of experts.

For this study, the panel consisted of fifteen BI system experts in EAMOs (Engineering Asset Management Organizations), who have been previously involved in the implementation of a BI system. The method was divided into three parts. Firstly, one-on-one interviews were conducted, in them a list of predefined factors was presented to discuss (although this is not the common Delphi practice).

The answers were analyzed to identify the factors that were most frequently brought up and the importance they were given. Secondly, those factors were combined into a list and given to the participants to confirm it. Lastly, the participants received a questionnaire to grade each factor on a scale from 1 (“not important”) to 5 (“critically important”). From the initial list, only the ones with an average rating of higher or equal to 3.5 were considered.

In general, the opinions of the participants didn’t vary much between each other. In descending order of importance, the seven most critical factors considered were:



Critical Success Factors	Mean	Std. Dev
• Committed management support and sponsorship	4.16	0.99
• Business user-oriented change management	4.10	1.00
• Clear business vision and well-established case	4.09	0.90
• Business-driven methodology and project management	4.08	0.88
• Business-centric championship and balanced project team composition	3.94	0.89
• Strategic and extensible technical framework	3.90	0.89
• Sustainable data quality and governance framework	3.82	0.91

Figure 3.3: Ratings of critical success factors by Delphi participants

As it can be observed, the majority of factors are related to organizational issues, as opposed to technical ones. The participants gave great importance to the business executives' support (‘committed management support and sponsorship’, ‘business user-oriented change management’, and ‘business-centric championship and balanced project team composition’).

This is mainly because they are responsible for defining the resources (budget, time, and people) to be expended on the project.

It is also necessary to have the project's scope, approach, and framework well defined. As well as alignment with the organizations' vision ('clear business vision and well-established case', 'business-driven methodology and project management' and 'strategic and extensible technical framework'). To have a well establish case it is important to clearly define the objectives, consider the risk involved, detail who will benefit from the implementation (employees, customers, or stakeholders), and look for case studies of similar organizations and their applications. When determining these factors, it's imperial to maintain realistic.

Finally, the project is dependent on the quality of the data, so if the company doesn't already have quality data, the information gathered from it can be also inaccurate ('sustainable data quality and governance framework').

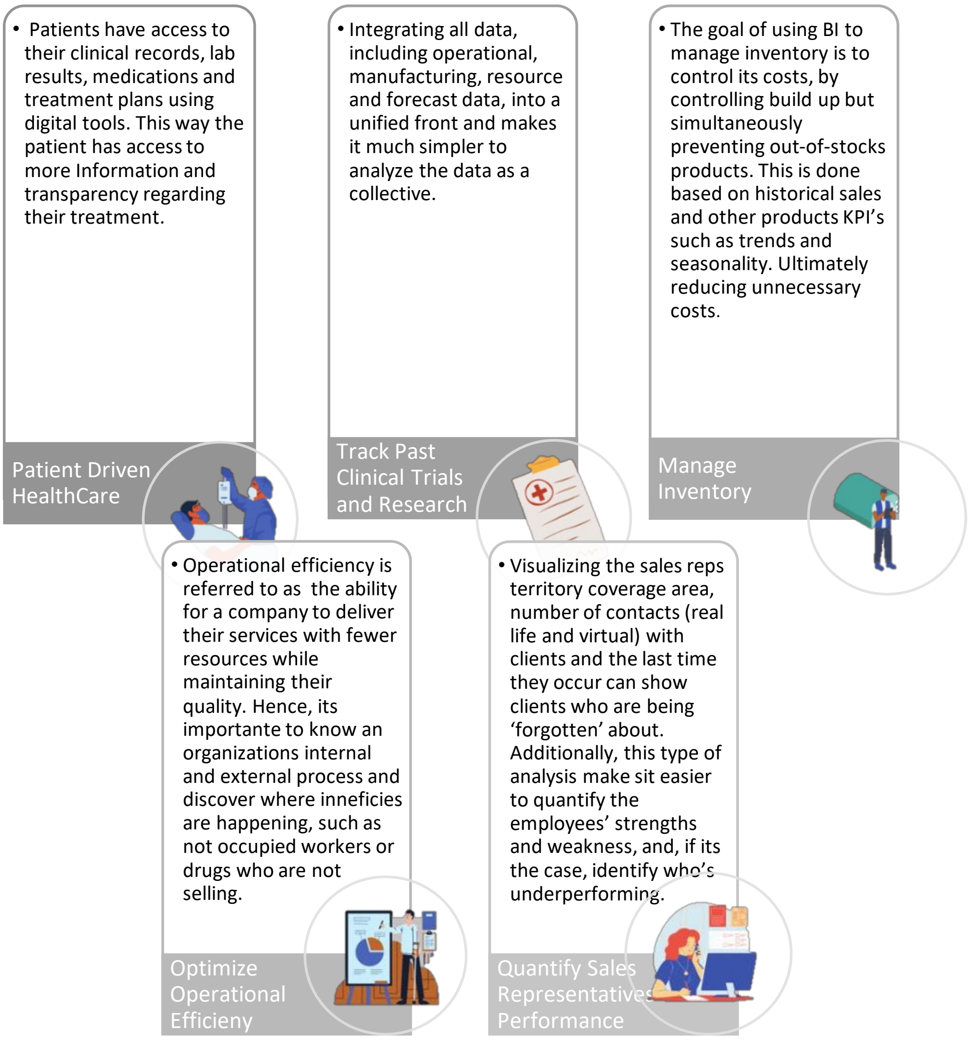
In addition to all the difficulty in applying these critical success factors, many reasons can limit a BI implementation and discourage companies, mostly small ones, to consider it. Firstly, there are very high implementation costs associated, and the implementation period is very long. Even requiring a continuous service, which also contributes to the high cost. It can also be hard to justify an implementation, considering the difficulty in providing complete objective benefits (actual numbers) as opposed to just general positive outcomes (better decision making).

3.1.3. BUSINESS INTELLIGENCE IN THE PHARMACEUTICAL INDUSTRY

In the chapter above, the BI concept was introduced, as well as other components necessary for its implementation. However, considering the scope of this report, it makes sense to go deeper into BI's role in the pharmaceutical industry.

According to the "Healthcare Business Intelligence Global Market Report", the global market for Healthcare BI was estimated at US$5.7 Billion in the year 2020, and it is projected to reach US$15 Billion by 2027. Alan S Louie, the Research Director of IDC Health Insights USA, claims this growth is partially due to "the increasing concerns over the expiry of patents for major blockbuster drugs, weak product pipelines, heightened awareness of drug safety, globalization, competition from generics and growing value considerations with regards to access to and reimbursement of new specialty therapeutics" (Louie). Consequently, Pharmaceutical companies have started relying on digital analytics to alleviate pricing pressures.

In Figure 3.4, it is presented some specific applications of Business Intelligence to the Pharma industry and that can bring the most benefits.



- Patients have access to their clinical records, lab results, medications and treatment plans using digital tools. This way the patient has access to more Information and transparency regarding their treatment.

Patient Driven HealthCare

- Integrating all data, including operational, manufacturing, resource and forecast data, into a unified front and makes it much simpler to analyze the data as a collective.

Track Past Clinical Trials and Research

- The goal of using BI to manage inventory is to control its costs, by controlling build up but simultaneously preventing out-of-stocks products. This is done based on historical sales and other products KPI's such as trends and seasonality. Ultimately reducing unnecessary costs.

Manage Inventory

- Operational efficiency is referred to as the ability for a company to deliver their services with fewer resources while maintaining their quality. Hence, its importante to know an organizations internal and external process and discover where inneficies are happening, such as not occupied workers or drugs who are not selling.

Optimize Operational Efficieny

- Visualizing the sales reps territory coverage area, number of contacts (real life and virtual) with clients and the last time they occur can show clients who are being 'forgotten' about. Additionally, this type of analysis make sit easier to quantify the employees' strengths and weakness, and, if its the case, identify who's underperforming.

Quantify Sales Representatives Performance

Figure 3.4: Applications of BI in the Pharma Industry

TechValidate, a 3rd-party research service, conducted a case study of Mission Health System based on Qlik customers. The study displays a more concrete example of how Business Intelligence can be used in the Pharma industry. Mission Health System is a large medical enterprise that was struggling with the volume of data they had, and how it came from various sources. Because of this, they needed a BI solution to integrate their data.

In the case study it is stated that by implementing Qlik, employees are now able to make faster and more confident decisions. Additionally, the company was able to reduce costs and risks. With data integration, departments can use the necessary data from each other. They have also seen that employees who use Qlik are finding new ways to approach business issues. In the case study, it is concluded that "they have improved their decision-making agility by 50% – 74% with Qlik." [3]

[3] https://www.techvalidate.com/tvid/1BF-84A-E1A

3.2. Data Warehousing

3.2.1. Data Source

A key factor in the quality of a BI implementation is the data. In most cases, organizations use both outside and transactional data. Transactional data can be defined as 'the information stored to track the interactions, or business transactions, carried out by an organization' (Larson B., 2016).

Companies can store the data in different ways. One method is the OLTP (Online Transaction Processing), a system that stores day-to-day transactions as they happen. This process is more focused on data processing. On the opposite side is OLAP (Online Analytic Processing), which is used to analyze data and, from that, provide business decisions. Information can be processed from multiple databases at the same time. This process is more focused on data analysis.

OLTP has the necessary raw data to create the metrics needed for BI. However, some factors can make that process difficult. Firstly, in most cases, the databases are relational, and the data is normalized, so multiple transactions can run simultaneously. However, in BI, aggregated data is needed (not individual transactions) since we want to compare records between each other. Creating aggregates with OLTP would take a lot of time and processing.

Because the system is created to deal with day-to-day operations, it normally doesn't save data for long. And when it does, the system can become slower and less efficient. With BI, the more data there is, especially historical, the more accurate the metrics will be, so this is a big impediment.

Organizations can often have more than one OLTP system, each regarding different operations and/or departments. As with BI, even with different data sources, the data needs to be combined and treated as a whole.

Despite this, by storing the OLTP information in a different data store, some of these difficulties can be solved. This can be achieved with a data mart, which stores historical data, often segments of the organization's data warehouse, like a specific area (e.g. finance). Another approach is to create a data warehouse, which is larger than a data mart, and all the historical information of the organization is stored.

However, this approach has its drawbacks, mainly due to the OLTP limitations themselves. Because OLTP's store data throughout the day, it requires periodical data loads, meaning it's not dealing with real-time data. Furthermore, the cleaning and consolidation process can be made harder when multiple OLTP systems are combined. However, this is already the most time-consuming step in most approaches.

OLAP, on the other hand, is presented specifically to help users retrieve information more quickly and easily. The data is normally stored in a data warehouse or in a data mart using cubes - multidimensional data structures where the users can access it in many ways to extract a subset of values that can answer specific business questions.

Because it is built for this purpose it has advantages over OLTP. The aggregates are stored preprocessed and can be read directly from the database. Moreover, it is much simpler for end-users since the structure is meant to match the structure of the organization, and therefore users are more familiar with creating queries and using it.

3.2.2. ETL (Extract, Transform, Load)

ETLs has become a major component of Business Intelligence because it allows for data from different sources to be grouped into a single usable format while guaranteeing quality and consistency. It is "everything between the operational source systems and the DW/BI presentation area" (Kimball & Ross, 2013).

Despite not being visible to end-users, it easily consumes 70 percent of the resources needed for the implementation and maintenance of a typical data warehouse. So, it is very important to understand this concept and how to implement it.

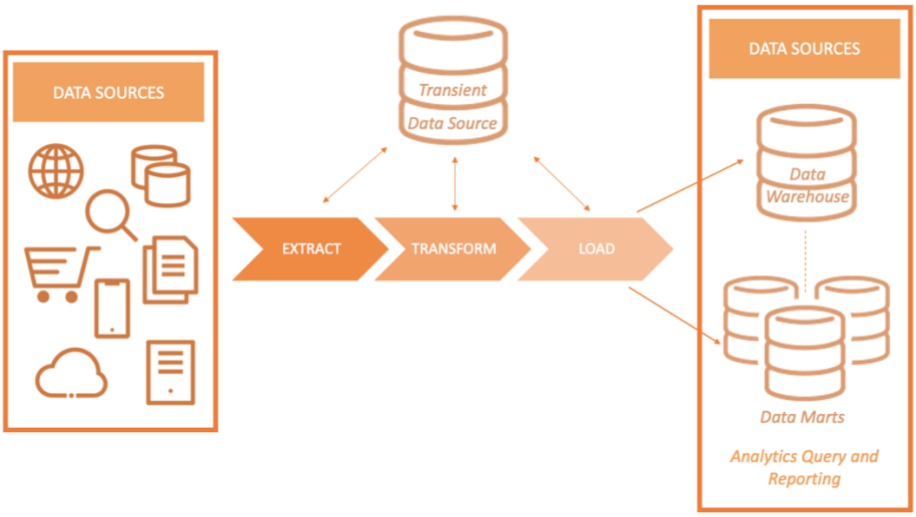


Figure 3.5: The ETL Process (Adapted from Sharda R., Dursun D., Turban E. Business Intelligence, Analytics, And Data Science: A Managerial Perspective, p.148)

The first step in ETL is **extraction**, which is the process of taking data, structured or unstructured, from different data sources and combining them. In this process, it is very common for the data to be somewhat incompatible. So there needs to be a logic to define the relationships between the different sources. To increase efficiency and ensure data integrity, after the extraction the data can be placed into a staging area. Which is a temporary storage used to quickly extract data from its data sources, to not compromise performance.

Afterward, the data is **transformed** into a format that can be analyzed. Some of the tasks involve data cleaning, such as correcting misspellings or modifying the data into the standard format, combining data from multiple sources, and removing duplicated data. This step is still performed in a staging area because it reduces the likelihood of data corruption.

And finally, structuring and l**oading** the data into the end target. This normally means loading all the data, followed by period loads of the data changes, incrementally. Full data loads are also done to replace and refresh the data, however, this happens much less often.

In the end, a properly designed ETL system provides a lot of benefits. It enforces data quality and integrity, since data from the different multiple sources now have a consistent structure. It delivers faster results. Most processing, such as eliminating missing data and format mistakes, is already done once loaded, so it’s ready to use by end-user tools.

3.2.3. Data Warehouse

A data warehouse (DW) is a centralized data storage system that integrates data from multiple locations into one, providing the enterprise with business analytics to help with decision making. In ‘Building the Data Warehouse’ (2008), Inmon attributes four fundamental characteristics to a data warehouse. In this way, a well-constructed data warehouse should be **subject-oriented**, meaning the data is organized by subject, such as sales. This is useful for businesses because it provides a more comprehensive view of the organization.

In addition, it's expected for the data to be **integrated** into a consistent unified version. To get to this point there must be naming, format, and coding conversions. The data should also be **time-variant**, meaning historical data is used for detecting trends and forecasting. And finally, **nonvolatile**, once the data is stored it cannot be modified, updated, or erased, but rather the changes are recorded as new records.

In general, data warehousing allows the processing of large amounts of complex data more efficiently.

3.3. Best Practices in the Data Model and Dashboard Creation

Having a well-designed data model is an essential step for creating a dashboard. Without the correct relationships, the insights can be very limited. Hence why it is so important to follow some common practices. During the internship, the project managers provided a guide with some steps that needed to be respected during the project's creation:

1. Reducing snowflakes

A Snowflake schema consists of a fact table surrounded by multiple dimension tables that are also connected with a foreign key to other dimensions tables via a many-to-one relationship.

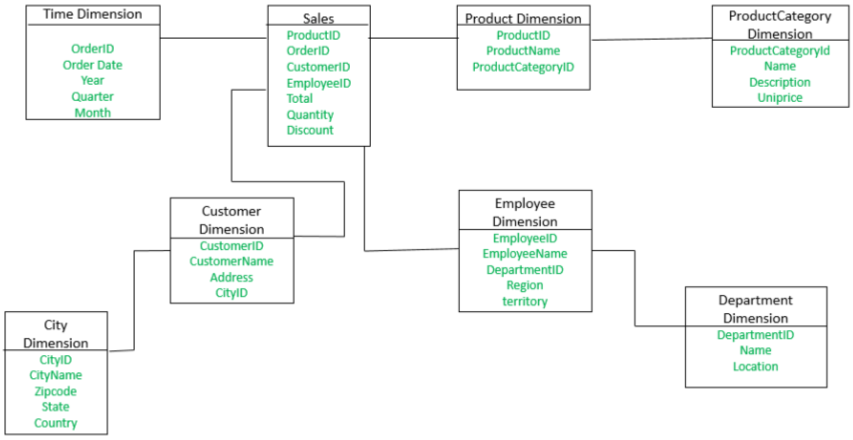


Figure 3.6: Example of a Snowflake Schema

This schema doesn't result in duplicated data, so it is easy to maintain. However, when dealing with big data volumes, which is the case for most SDG projects, it's normally not the best option. It is harder to design compared to a star schema and the queries can be more complex and slower because of the number of joins between the tables. Which are required by the primary and secondary dimension tables. As a result, the query processing time is higher, and performance is worse.

In addition, to optimize the data model, it is recommendable to have one single Fact table, and all Master tables around. This way, the number of jumps between tables is reduced. However, due to the elaborate models that need to be created, this is very frequently not possible.

2. Removing Synthetic Keys

They are generated when there are two or more common fields between two different tables. They can be damaging to the functionality of the model because they provide redundancy. Generally, this is undesirable since it causes problems maintaining consistency after updates and ends up using an excessive amount of time and/or memory. This isn't always the case, but for the most part, the cons outweigh the pros, and it is advised to avoid them.

To remove them, first, it is necessary to determine if there is a need to have the field in both tables. If not, the solution is quite simple, deleting the field where it is not needed. If yes, it could be more beneficial to create a key with the concatenation of both fields and delete or comment the fields in common. To make the model more efficient we should apply the auto number function and convert the alphanumeric fields into numeric ones.

Qlik creates a warning when a synthetic key is created, so in most cases, they are easy to identify and fix.

3. Keep the files clean and readable

When working on a team, it is important to maintain the files as clean as possible, so when another person takes over or needs to reuse them on their own project it's not confusing. To help with that it's advised to always comment on newly implemented logic, use the same naming convention, and be consistent with the format (capital letters, paragraphs, ...).

Additionally, unused fields and tables need to be removed from the data model. This significantly reduces the size of the application. And improves performance and readability.

4. Think about scalability

A lesson frequently brought up during the internship was to make sure the task was scalable. Scalability is achieved in the ability to handle changing demands. This means, not just thinking about the 'quick fix' that will work for right now, but rather coming up with solutions that will remain correct and usable, even if slight changes occur.

For data model creation, this can mean, for example, considering all possible date formats using a *pick match* - a function similar to an *if* but with better performance in Qlik. This is important since it is very frequent for them to change and their format will impact how the calculations, like extracting the month, are done. However, it goes much further than that.

The measures used in the dashboard are kept in excel and stored as variables. This way, in case of a change in the measure's definition, instead of having to change all the dashboards every time the logic is used, it only needs to be changed once.

5. Simple is better

When creating a dashboard, it's important to not have too many visualizations. Otherwise, it can become too crowded and take away from its actual purpose: to make it easier to draw conclusions.

In the end, it is important to note the end-users are, normally, not from a data background, and for the most part they are part of the sales or management team. When creating a dashboard, data literacy needs to be considered. Everything needs to add value and contribute to better decisions. This is important to keep in mind because creating complicated visualizations can be quite timely but if the user isn't able to understand them, they will probably not end up in the final work and it will just be 'wasted' work and time.

3.4. Dashboards

Today, dashboards are an integral part of BI. A dashboard is an interactive data visualization and analysis tool, that presents critical KPIs, visualizations, metrics, and key data points in an organized way. It allows easier and quicker analysis that enables users to deeply explore the data directly within the dashboard.

Dashboards can have many purposes, for this specific report a performance dashboard was created. According to Eckerson performance dashboards allow a company to monitor the business activities, analyze where the problems are coming from and provide information that can help resolve them, and manage people and processes to contribute to an overall increase in performance (W. Eckerson, 2005).

As mentioned, the goal of a dashboard, especially in BI, is to provide information that is easy to view and interpret and to make information easily accessible. For this reason, a good dashboard should straight away draw attention to the most important information.

Additionally, it should only contain relevant information. There needs to be a reason why the data is displayed, and it needs to be shown in the correct format. The type of information will determine how it should be presented. For example, line charts are better for displaying patterns, whereas bar charts are better at comparing items in the same category.

To prevent the dashboard from being too cluttered drill-downs or drill-throughs are a great way to show a higher level of detail without compromising the design of the dashboard.

A "drill down" is a feature that allows the user to go from a more general field (Country) to a deeper one (City). And a "drill through" is a feature where the user passes from one report to another, with the same data.

In figure 3.7 some examples of how to correctly display the data are shown, considering its purpose.

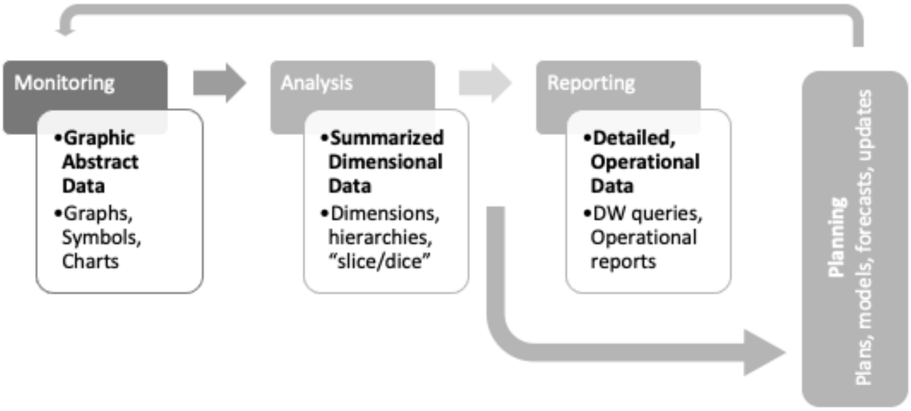

Monitoring
•Graphic Abstract Data
•Graphs, Symbols, Charts
Analysis
•Summarized Dimensional Data
•Dimensions, hierarchies, "slice/dice"
Reporting
•Detailed, Operational Data
•DW queries, Operational reports
Planning
Plans, models, forecasts, updates

Figure 3.7: Most distinctive features of a dashboard. (Adapted from Eckerson W., 2005, Performance Dashboards: Measuring, Monitoring, and Managing Your Business)

3.5. QLIK SENSE

For the duration of the internship, the toll used to create dashboards was Qlik Sense. The platform is known for allowing organizations to become data-driven and, make better decisions from that.

One of the benefits of Qlik Sense is great visualizations. Even being recognized in 2020 as a leader for both Business Intelligence and Data Discovery and Visualization by Trust Radios. Additionally, because it is designed for end-users, it is easy to use and permits the creation of presentations within the tool that lets the data 'tell a story'. It ensures data governance and security, everyone is using the same data sources and formulas, while simultaneously having the option to define which employees are allowed to see which data.

3.6. GITHUB

GitHub is a Git repository hosting service. In the context of SDG, GitHub was used to manage the different versions of the files the team was working on. When working with a big team it's easy for files to be overwritten or mistakes to happen. Someone can accidentally delete a file, paste over the wrong one, make a change that isn't correct and want to go back, and many other examples. GitHub helps with exactly that.

To prevent the situations described, when working on a file there are some steps to consider. First, we make sure we are the only ones working on that file/dashboard and start on the version in the drive. The files on the drive are the ones connected to Qlik. Once we are confident of our changes, we can paste the file we changed to the same folder but stored in GitHub. And then commit it and push, so that version exists in everyone else's GitHub repository as well. That way if someone overwrites our work, we have it on GitHub.
Before doing any commits and pushes, it's very important to always pull, this grabs changes from other repositories into yours. And it guarantees the most recent version is used.

4. METHODOLOGY

The presented report describes the learning and experiences gathered during the period of the internship. As mentioned, this report is focused on the work done for Daiichi-Sankyo. Because the SDG Group follows a SCRUM Methodology, each project is composed of different sprints, all with a relatively short duration. All sprints follow the same structure, and although with different objectives, the overall process is very similar.

For this reason, and to avoid repetition, the details below will focus on one of the sprints I participated in. However, with each new sprint, new challenges arose, which allowed for the development of new skills and improvement of already obtained skills.

4.1. SCRUM

4.1.1. SCRUM CONCEPT AND FRAMEWORK

SCRUM is an agile software development framework for managing product development. It is defined as a "lightweight framework that helps people, teams, and organizations generate value through adaptive solutions for complex problems" [4]. Because of its flexibility, teams work more as a unit and are incentivized to collaborate.

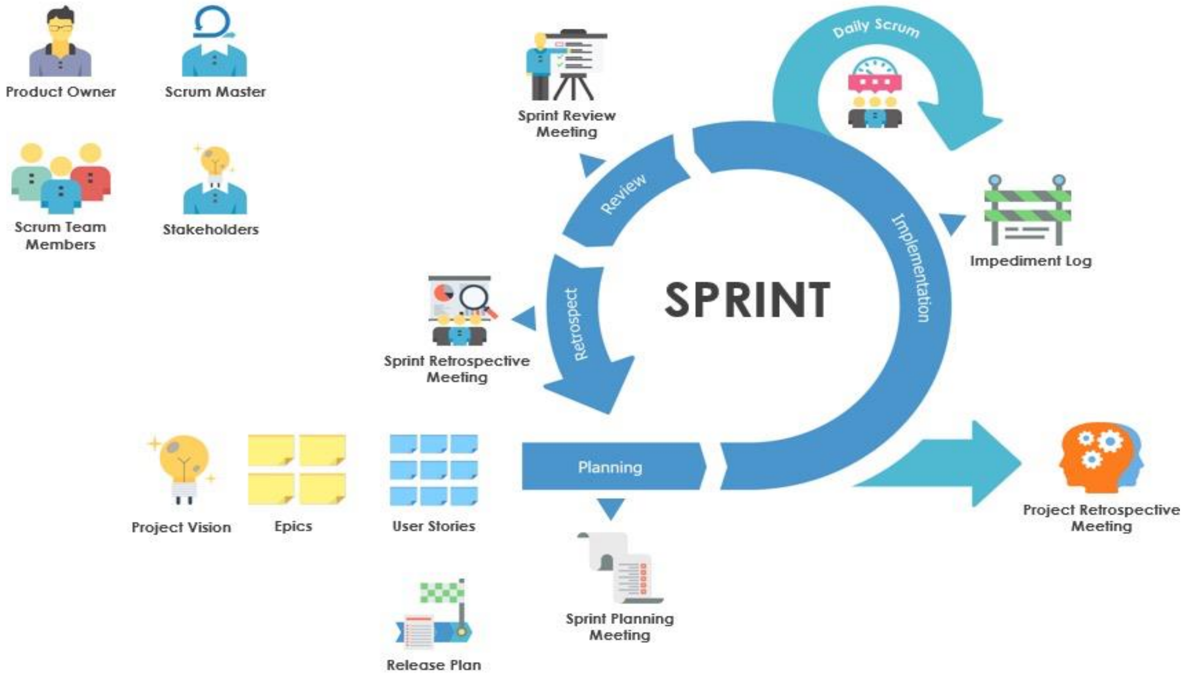


Figure 4.1: Scrum Process (Source: Visual Paradigm)

In Scrum, teams work in short timeframes called sprints, normally around two to four weeks. Each sprint begins with a project vision and the plan to see that vision through. It ends with a review of the work completed. By the end of each sprint, all the requirements should be fully completed and tested.

[4] https://www.scrum.org/resources/what-is-scrum

Occasionally, if the initial plans seem unfeasible considering the time constraints, the less important requirements will be put in the backlog in case there is extra time or to be done in the next sprint.

To understand the SCRUM Methodology, it is first needed to understand some of the concepts used. **User Stories** are where the requirements requested by the business team are described. For the projects detailed in this report, Jira was used to create the US's. Although at the beginning of each sprint some tasks are already defined, every time the Project Owner informs of a change that needs to be done, a new User Story should be created. Each User Story has a **Story Point** assigned to them, which is the estimated effort (number of hours) that the US will take.

As it was mentioned, some US's will be put in the Product Backlog, which is a list of the USs that are ready to be assigned to a Sprint. The sprint the Dev team is currently working on is called the **Active Sprint.** Finally, any blocking point that prevents the Dev team to continue their assigned tasks is called an **Impediment.**

4.1.2. SCRUM TEAM

As displayed the figure 4.1, this process involves four different types of intervenient. The **Stakeholders** are external to the Scrum Team. They are represented by the Product Owner, who should manifest their opinions to the rest of the team.

The **Product Owner** is responsible for the vision of the project and to convey that vision to the rest of the Scrum Team. They are responsible for leading the sprint planning. As well as defining and communicating the project's goals.

The **Scrum Master** ensures the team completes what was agreed on during the sprint planning. They should help with and manage possible impediments, blockers, or dependencies. They lead the daily scrum and sprint retrospectives meetings.

The **Development Team (Scrum Team)** completes the User Stories assigned to each sprint and provides an accurate estimation for the time spent on each task. They lead the Sprint Demo meeting.

4.1.3. SCRUM EVENTS

Scrum events are occurrences that happen every sprint. Initially, the sprint begins with **Sprint Planning**. This is where the Product Owner determines the goals and tasks to work on during the sprint. There should also be a discussion of the initial plan for completing the product backlog items, such as the necessary resources and expected time for release.

Once the sprint has started there should be a **Daily Standup**. As the name implies, is a quick meeting that happens every day. Each member shares the progress they made the day prior and what they have planned for that day. As well as any impediment felt. Since the entire team is working remotely these meetings are very important to maintain an agile development framework.

At the end of the sprint, a **Sprint Review** is held. In the meeting, the completed work and its outcome are shown. Depending on the reviews further adaptations may be necessary and progress towards the goal is discussed.

The last event is the **Sprint Retrospective,** where the work done is discussed. The goal is to improve future sprints by discussing the positives and less positives of the sprint. For the problems encountered the solutions should be shared.

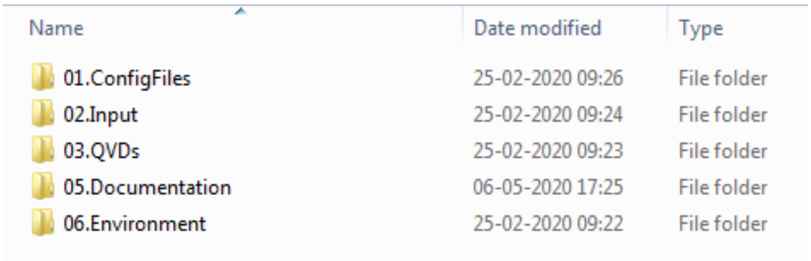
5. PROJECT STRUCTURE

In this chapter, the organizational framework continuously followed in all projects is described.

5.1. The Application Layer Container

To maximize Qlik Sense's potential, it is important to have a structure that allows the best performance, maintenance, traceability, and scalability. As a result, all data extractions and transformations for the Europe Daiichi data are performed in a stream folder entitled '71_DSE_ARE_QVD_Layer'. It's in this folder that the application's QVDs are stored.

Each dashboard has its own folder (e.g. 88_DSE_Productxyz_Performance) where minor changes, data specific to those dashboards, and the text files to load are stored. Below is an explanation of how these folders are organized. Although the content of the apps varies, all of Daiichi Sankyo Europe Apps follow a very similar structure. The Application Layer Container includes the following folders:

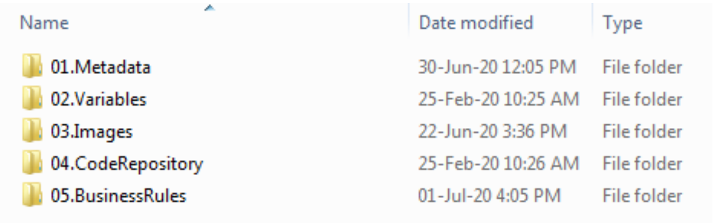


Name	Date modified	Type
01.ConfigFiles	25-02-2020 09:26	File folder
02.Input	25-02-2020 09:24	File folder
03.QVDs	25-02-2020 09:23	File folder
05.Documentation	06-05-2020 17:25	File folder
06.Environment	25-02-2020 09:22	File folder

Figure 5.1: Application Layer Container

5.1.1. Configuration Files

Inside the Config. File's folder the following subfolders can be found:



Name	Date modified	Type
01.Metadata	30-Jun-20 12:05 PM	File folder
02.Variables	25-Feb-20 10:25 AM	File folder
03.Images	22-Jun-20 3:36 PM	File folder
04.CodeRepository	25-Feb-20 10:26 AM	File folder
05.BusinessRules	01-Jul-20 4:05 PM	File folder

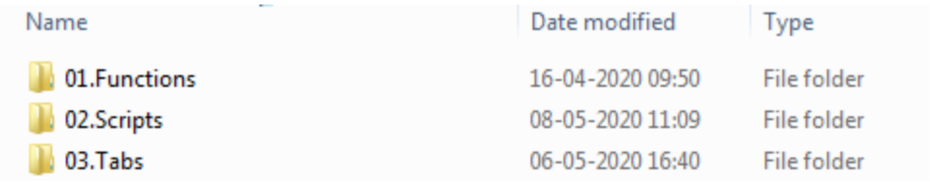
Figure 5.2: Cong Files Folder

a) **Metadata**: Inside this folder, there is an excel called 'DSE.App.xyzproduct.Metadata'. It loads the necessary tables and fields into the Qlik Sense Data Model.

b) **Variables:** This folder has four subfolders, however the most important is the Application one. This is where the KPI's variables are stored to be loaded into the dashboard. It falls in this category expressions for the applications calculations, labels, customized error messages (e.g "No data available", "Please select a country"), and any other variables that affect the dashboards objects. The benefit of creating variables, as opposed to adding the measure directly in the dashboard, is that they can be used as a reference from every sheet object or even other dashboards. This is very important for scalability. For example, if a calculation is no longer multiplied by 2 but now it’s by 3, instead of having to go to all the visualizations where it's used, it's possible to make that change directly in the excel.

c) **Images:** This folder contains all the images that the system is going to use.

d) **Code Repository:** This folder is divided into three subfolders:



Name	Date modified	Type
01.Functions	16-04-2020 09:50	File folder
02.Scripts	08-05-2020 11:09	File folder
03.Tabs	06-05-2020 16:40	File folder

Figure 5.3: Code Repository Folder

1 - **Functions:** Keeps the text files with functions that are used for specific tasks. Such as loading files or variables or deleting them.
2 - **Scripts:** This folder contains the general script for both data extraction (for L1 and L2), building the data model, creating the path variables (using the app’s name), and deleting the created variables. The files in this folder are the same in all dashboards, only changing the names of the files themselves.
3 - **Tabs:** Text files where the main transformations are done to be loaded in Layer 2.

5.1.2. INPUT

The input folder is where the business or the development team can insert the data that is not contained in the main source systems (SQL Cardio Database). The files are in an excel format, organized by tables. These files are included in the Sources Metadata excel so they can be loaded into the data model. They are easy to manage and transform.

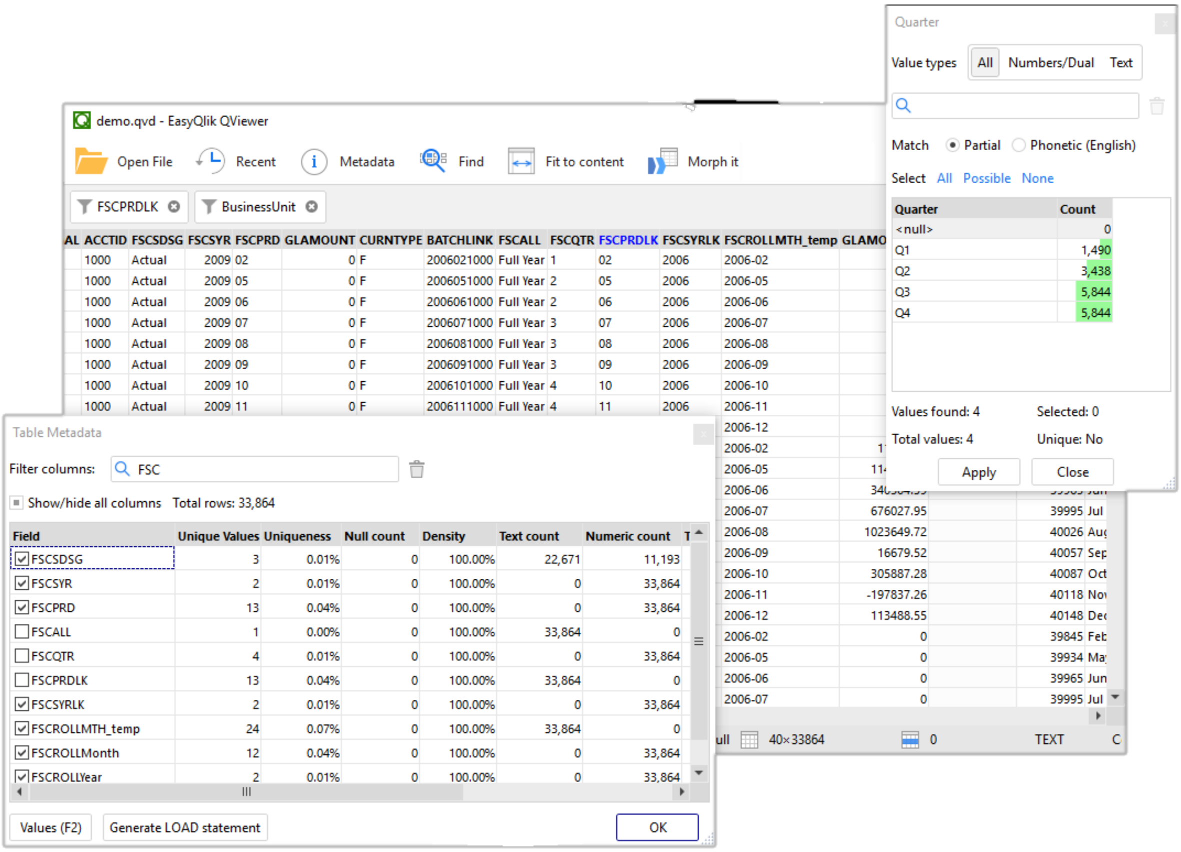
This folder includes two subfolders: Fact.Data and Master.Data. In the Master Folder, there are three other subfolders. The conversion rates, the mappings (that contain several auxiliary files needed in the data transformation process, and to add additional detail in the data model,) and the section access (where the user’s login information is stored).

5.1.3. QVD'S

A QVD is a file format that contains a data table (each QVD is one table) exported from QlikView. Because of its format, it is much faster (10-100 times) than other data files.

To read those QVD’s, a program called *‘EasyQlik QViewer’* is used. It allows for a row-level view. Additionally, by clicking on a field it is possible to see the count of each distinct value. It also provides the code to load the data into the Qlik app.

However, it has some limitations. The version used only displays a maximum of 10,000 rows. In case it has more rows (which almost all QVDs have), the automatic measures will not be correct. Furthermore, due to the high volume of data, it can take a while (sometimes minutes) just to open a file.



demo.qvd - EasyQlik QViewer

AL	ACCTID	FSCSDSG	FSCSYR	FSCPRD	GLAMOUNT	CURNTYPE	BATCHLINK	FSCALL	FSCQTR	FSCPRDLK	FSCSYRLK	FSCROLLMTH_temp	GLAMO
	1000	Actual	2009	02	0	F	2006021000	Full Year	1	02	2006	2006-02	
	1000	Actual	2009	05	0	F	2006051000	Full Year	2	05	2006	2006-05	
	1000	Actual	2009	06	0	F	2006061000	Full Year	2	06	2006	2006-06	
	1000	Actual	2009	07	0	F	2006071000	Full Year	3	07	2006	2006-07	
	1000	Actual	2009	08	0	F	2006081000	Full Year	3	08	2006	2006-08	
	1000	Actual	2009	09	0	F	2006091000	Full Year	3	09	2006	2006-09	
	1000	Actual	2009	10	0	F	2006101000	Full Year	4	10	2006	2006-10	
	1000	Actual	2009	11	0	F	2006111000	Full Year	4	11	2006	2006-11	

2006-12			
2006-02	1		
2006-05	11		
2006-06	34		
2006-07	676027.95	39995	Jul
2006-08	1023649.72	40026	Au
2006-09	16679.52	40057	Sep
2006-10	305887.28	40087	Oct
2006-11	-197837.26	40118	No
2006-12	113488.55	40148	De
2006-02	0	39845	Feb
2006-05	0	39934	Ma
2006-06	0	39965	Jun
2006-07	0	39995	Jul

ull 40×33864 0 TEXT C

Table Metadata

Filter columns: FSC

Show/hide all columns Total rows: 33,864

Field	Unique Values	Uniqueness	Null count	Density	Text count	Numeric count	T
☑ FSCSDSG	3	0.01%	0	100.00%	22,671	11,193	
☑ FSCSYR	2	0.01%	0	100.00%	0	33,864	
☑ FSCPRD	13	0.04%	0	100.00%	0	33,864	
☐ FSCALL	1	0.00%	0	100.00%	33,864	0	
☐ FSCQTR	4	0.01%	0	100.00%	0	33,864	
☐ FSCPRDLK	13	0.04%	0	100.00%	33,864	0	
☑ FSCSYRLK	2	0.01%	0	100.00%	0	33,864	
☑ FSCROLLMTH_temp	24	0.07%	0	100.00%	33,864	0	
☑ FSCROLLMonth	12	0.04%	0	100.00%	0	33,864	
☑ FSCROLLYear	2	0.01%	0	100.00%	0	33,864	

Quarter

Value types: All Numbers/Dual Text

Match ● Partial ○ Phonetic (English)

Select All Possible None

Quarter	Count
<null>	0
Q1	1,490
Q2	3,438
Q3	5,844
Q4	5,844

Values found: 4 Selected: 0

Total values: 4 Unique: No

Figure 5.4: QVD File (Source: *https://easyqlik.com*)

Inside the QVD's folder is where all the created QVD's are stored. The subfolders are organized according to the different extraction layers, L0, L1, and L2. Each containing two subfolders, Master and Facts data.

In Layer 0 is where the QVD's created by the QlikView app are stored. In Layer 1, very minor transformations (e.g., renaming) are made, and the interfaces extract the data from the data sources. Finally, in Layer 2, the extractors apply the metadata parameters stored in the configuration files. In this phase, most transformations are made to apply business rules and prepare the data for the data model in the correct format.

5.2. System & Software Architecture

Due to Qlik Sense's security, it is not possible to connect the database directly with Qlik Sense. So, the first step is to extract the data from the SQL Cardio Database using QlikView. The database includes the Daiichi-Sankyo UK CRM data (Visits and Target Customers).

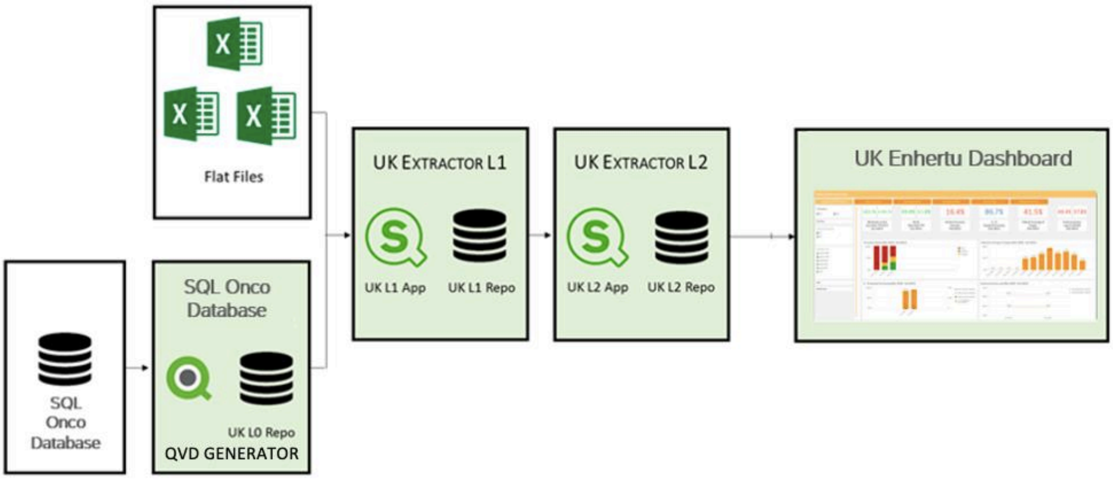


Figure 5.5: System & Software Architecture

As shown in figure 4.4, an ETL process is followed. During this process, the data is sequentially extracted from the SQL Onco Database, loaded into the extractors, and then transformed.

First, the 'QVD Generator' feature allows delivery of specific QVD files to developers and end-users and schedules tasks to create QVD files. The data from the different flat files are loaded with Qlik View and then, the generated QVDs will be automatically stored. In this case, in the L0 QVDs folder.

Then, the data from the L0 Layer and the flat files in the Input folder (specified in the Metadata) are extracted into Layer 1. In this phase, no transformations are done (except for some field renaming).

The Layer 2 extractor loads the data from the L1 layer and transforms it to be ready for the presentation layer. This is where more complicated transformations, such as business rules and calculations occur.

With the final QVDs stored in Layer 2, the data is loaded into the Qlik Sense app. The architecture provides an easily maintainable and scalable solution. This is very important, considering how often there are new requirements that may arise.

In the dashboards, Qlik objects are created to obtain relevant and easy to interpret visualizations.

6. PROJECT DESCRIPTION

6.1. SPRINT PLANNING

As mentioned before, the Sprints have a small duration and follow, for the most part, the same steps. Although, during the scope of this project various sprints were completed, to avoid repetition, the report will detail ‘Sprint 1’.

Before any development is done, the ‘Sprint Planning Meeting’ occurs. In this meeting, SDG is briefly introduced to the stakeholders, as well as the teams from both SDG’s side and the client’s side. Every member's role must be clearly defined.

Following, the scope of the project is presented, which includes the data sources necessary, and a mock-up of the dashboard. These requirements are agreed to previously and are estimated to fit into the sprints plan. The project is divided into two sprints, in the first one the most priority visualizations are included. It is sort of a beta version of the final product.

After it goes live, some time passes after the second sprint can start again. This is so the user has enough time to properly test the dashboard and to find any mistakes, or any changes they would like to see. As well as new visualizations and metrics they think should be included.

In the second sprint, new requirements are added, considering the user feedback.
Finally, after the second sprint goes live again, the Hypercare period starts. In it, the users are trained to use the dashboard seamlessly.

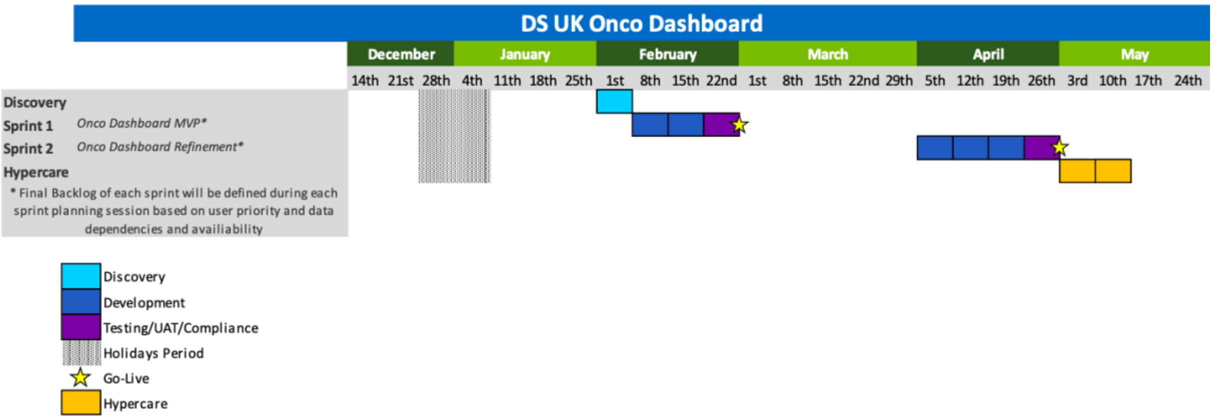


Figure 6.1: Sprint Planning

Lastly, the SCRUM agile approach is presented. This will allow users to provide and incorporate feedback along with the project, resulting in an overall better business engagement and satisfaction.

Typically, there should be a Daily Stand up to discuss the project status, outstanding issues, and questions, lasting about fifteen minutes. However, due to the business owners busy scheduled, it was replaced with a Weekly Meeting of one hour. The rest of the scrum ceremonies are also planned.

6.2. SPRINT DESCRIPTION

The sprint focuses on the sales and performance of a new product created from a partnership between two of the most successful pharmaceutical companies in the world. Given the drug has only been recently distributed to the public, it is very important to see how the UK market and patients are reacting to it.

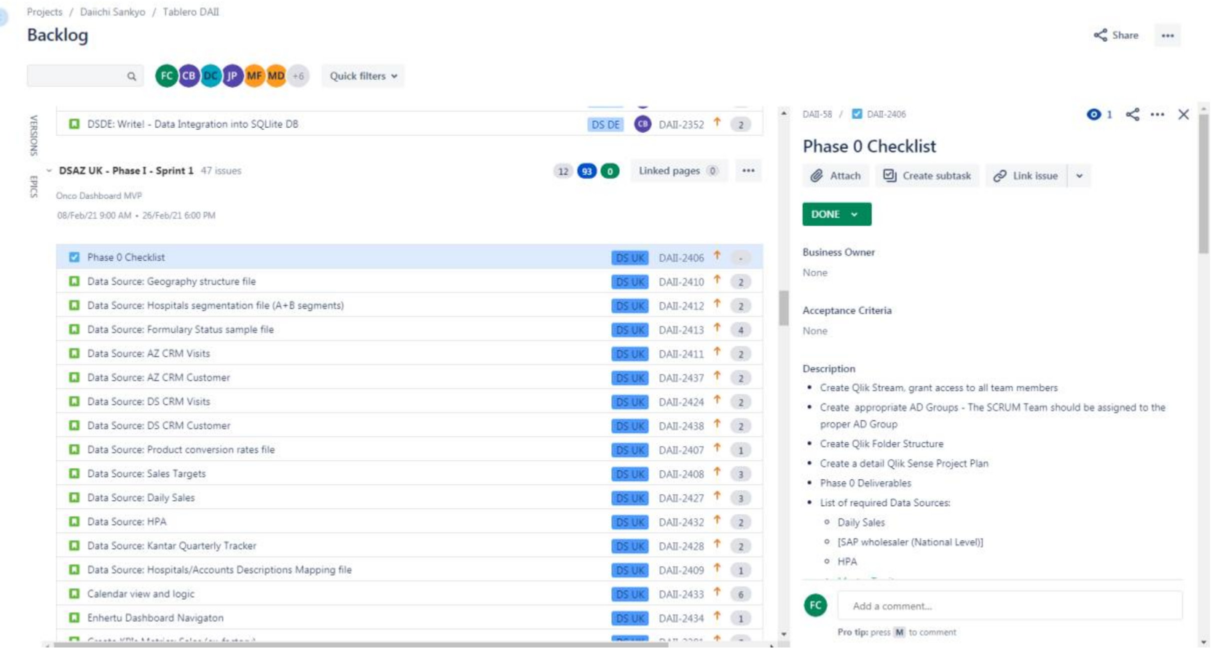
In this way, the dashboard solves the business need for detailed analysis of the product behavior and customer acceptance. The dashboard should aid both companies to grow their sales. With the information provided, it will be easy to understand what actions need to be taken and how some market segments (e.g. Hospitals) are being neglected.

The main objectives are to:

- identify the growth product over time,
- compare the product's performance against territory data and between companies,
- visualize the current and forecast sales,
- evaluate the percentage of sales by category (face to face, calls, meetings, …).

6.3. DISCOVERY

The first week includes discovery. This is where the team gets familiar with the data and the needed requirements. All the tasks that are necessary for the creation of the dashboard are included in the backlog of Jira as User Stories. Below is a screenshot of said backlog, highlighting the 'Phase 0 Checklist'.



Projects / Daiichi Sankyo / Tablero DAII

Backlog

DSDE: Write! - Data Integration into SQLite DB	DS DE	DAII-2352	2

DSAZ UK - Phase I - Sprint 1 47 issues 12 93 0 Linked pages 0

Onco Dashboard MVP

08/Feb/21 9:00 AM • 26/Feb/21 6:00 PM

☑ Phase 0 Checklist	DS UK	DAII-2406	-
Data Source: Geography structure file	DS UK	DAII-2410	2
Data Source: Hospitals segmentation file (A+B segments)	DS UK	DAII-2412	2
Data Source: Formulary Status sample file	DS UK	DAII-2413	4
Data Source: AZ CRM Visits	DS UK	DAII-2411	2
Data Source: AZ CRM Customer	DS UK	DAII-2437	2
Data Source: DS CRM Visits	DS UK	DAII-2424	2
Data Source: DS CRM Customer	DS UK	DAII-2438	2
Data Source: Product conversion rates file	DS UK	DAII-2407	1
Data Source: Sales Targets	DS UK	DAII-2408	3
Data Source: Daily Sales	DS UK	DAII-2427	3
Data Source: HPA	DS UK	DAII-2432	2
Data Source: Kantar Quarterly Tracker	DS UK	DAII-2428	2
Data Source: Hospitals/Accounts Descriptions Mapping file	DS UK	DAII-2409	1
Calendar view and logic	DS UK	DAII-2433	6
Enhertu Dashboard Navigaton	DS UK	DAII-2434	1

DAII-58 / DAII-2406

Phase 0 Checklist

Business Owner

None

Acceptance Criteria

None

Description

- Create Qlik Stream, grant access to all team members
- Create appropriate AD Groups - The SCRUM Team should be assigned to the proper AD Group
- Create Qlik Folder Structure
- Create a detail Qlik Sense Project Plan
- Phase 0 Deliverables
- List of required Data Sources:
  - Daily Sales
  - [SAP wholesaler (National Level)]
  - HPA

Figure 6.2: Sprint 1 Backlog

Before creating the application there are some steps required so it can properly function. As described in the checklist, before any development is done a Qlik Stream needs to be created. A stream is where the Dev Team can publish the app so the end-users can access it.

The streams have security rules so only those users who are meant to see the app can view it, hence why it is the first step.

Simultaneously, the folder where all the necessary files are stored, is also being requested. These folders have a particular structure to them, which is also created in this phase. In the chapter, 'The Application Layer Container', more details are presented.

6.4. SECURITY

Above it is mentioned that the dashboards are restricted. For the user to see it, first, they need to request access to the stream through 8man, a software that provides integrated access right management solutions.

Then, it is necessary to request access to the app itself. To provide this access, the usernames, and respective information is included in the DSAZ User List.

In this specific dashboard, one of the visualizations is not available for all the app users. The access is again, determined in the DSAZ User List file by having 'Full Access' = Yes. For the users without access, in place of the visualization, there is simply a white box.

6.5. DEVELOPMENT

Once all the business owners have provided data from the required sources, the development can start. This is where the first requirements are fulfilled.

6.5.1. CREATING THE DATA MODEL

Before any visualizations can be created, the back end needs to be constructed. The first step is to include all the flat files in the Input section of the application folder. And then, for organizational purposes, divided into master and fact data. For this, it is useful to already have an idea of what the data model should look like.

Afterward, the inputs and any other databases needed are included in the **Sources Metadata** file. As explained, the excel should include all the sources to be converted into a QVD, stored in layer 1. The excel is always composed of these three sheets:

- **Config (sys):** It stores the information about the sources, such as their type (DB, XLS, XLS CROSS, …), historical depth (whether it is full reload, for master data or incremental for fact data), and the QVD Type (Fact or Master)

- **Sources (sys):** The connection string and stream of where the sources are stored.

- **Tables (sys):** Each row has the information of a QVD. In this sheet, the most important columns are the:
  - **Sheet Table:** For each QVD, a separate sheet is created with more detail on what fields to load. The 'Sheet Table' corresponds to the name (number) of where that information is stored. This is true for all rows except for cross table excels since the fields cannot be specified.

  - **QVD Type:** Specified if the QVD will be stored in the Master or the Fact folder of the Layer 1 QVD and if it should be loaded and overwritten when the Layer one extractor is running again. Additionally, for the fact data from flat files, it is also specified it the data is incremental. Meaning only new or modified records are uploaded to the QVDs.

- **QVD's Fields:** As mentioned, for each QVD a sheet is created, with the corresponding 'Sheet Table' number. Inside each sheet, there is a table with four fields: 'Source Field' - the original name, 'Field Alias' - the name to appear in the model, 'Field Load SQL' (if it's to be loaded to SQL), and 'Field Load QVD' (if it's to be loaded to a QVD).

After the excel is completed, Layer One is loaded, and the inputs/data sources are converted into QVDs.

Now the major transformations can take place. For this specific project, there is data from two companies, so it is important to unify them and make sure the related fields are properly connected. Mainly, concatenations and outer joins were used for this purpose.

The data model counts with only six tables, selectors excluded.

1. Territory Table

Starting with the Territory table (M_TERRITORY), it includes the location of all the activities and sales. An OUTER JOIN is done to combine all the territory related QVDs into one.

Initially, the Master Territory Structure provided was the only territory data to be included. However, this data only provided Hospital locations. For this reason, another file was also included, to map the sales that did not belong to hospitals. The file was at Brick and Postal Code Level, and it was concatenated to the Territory QVD of the same level.

For context, "a brick is a collection of accounts and contacts, normally in a common geographic area"[5] (Siebel Life Sciences Guide, 2007), its main purpose is to allow a company to publish Retail Sales Analysis data while protecting the privacy of the accounts, in a way where the sales cannot be tracked.

For the Master Territory Structure, the granularity went from Hospital Name to Postal Code. However, the hospital field is also needed for the M_HOSPITAL table. Having it in both tables would create a synthetic key. And, as explained in '2.4 Best Practices in the Data Model and Dashboard Creation', that would be problematic.

Additionally, the data comes at different granularity levels. One example is the forecasting field, which represents the two types of forecasts included. One at National level (UK) and the other at Territory level (North, South, East, West). By visualizing the data against territory level some of the forecast values would be null, and by comparing it against national level, important information would not be shown.

[5] https://docs.oracle.com/cd/E05553_01/books/LSSIA/LSSIAaccounts4.html

To solve this, a field common for all levels is also included, the ‘Territory Key’. It starts with a number representing the level of detail followed by the actual field value. For example, ‘1_CF31’ represents the Brick CF31. The Territory Key is a staple in SDG best practices and is used in most data models.

2. Hospital table

For the M_HOSPITAL, the Hospital Segments QVD was used, which contained the Hospital’s name list and the segment they belong to. It was also added an ‘Hospital_Counter’. The Count function is avoided because of its slow performance, so instead this solution is used in a lot of master tables. These fields only have the value ‘1’, and as the name suggests, are used to count the number of rows.

3. Customer table

In the M_CUSTOMER table, there is data from both companies, with a different level of granularity. For the first company, the customers are at an Account level and for the second they are at a Postal Code level. The two are concatenated and a mapping is applied. This way, the territory data matches the Territory table. A Customer Counter is also included.

4. Calendar tables

The M_CALENDAR table is a predefined table used for most dashboards. It has all the related time fields (Date, Month, Year, …), with date calculations, such as Year to Date, Calendar Difference, Flag Temp (a binary field that indicates if it is past or future data).

5. Fact table

The facts table data is incremental, meaning only new or updated records are loaded each time. The first time the model is created, all rows are added to a QVD in layer 2, after that only the most recent data from layer one is loaded and concatenated to the rest. Because it corresponds to daily data, only 'today' is loaded.

For the table not to accumulate an excessive number of rows, files from over 2 years ago are deleted. Additionally, the mapping and calculations are done, until the data no longer needs any further transformation.

Afterward, the facts table is concatenated with all the master tables, using the date field. For each master table, a metric is loaded, in most cases being a counter.

Succeeding the transformations, the layer 2 extractor is loaded in the Extractor L2 Application, and with it, the respected QVDs are created. Finally, those QVDs are stored in the Apps Metadata Excel. Similar to the Sources Metadata file, the excel has a sheet titled ‘Tables (sys)’ with information about the QVDs name, type, layer, and path. And, if applicable, a where clause. Then for each of the data model’s tables, there is a sheet with the tables to load.

The data model is now complete and can be used in the dashboard application. It is loaded through a text file titled App and the name of the dashboard. This file contains the formatting and coloring of the application, variables, the master and fact tables, the section access (used to control the security of an application and define who gets to see what), and the selectors.

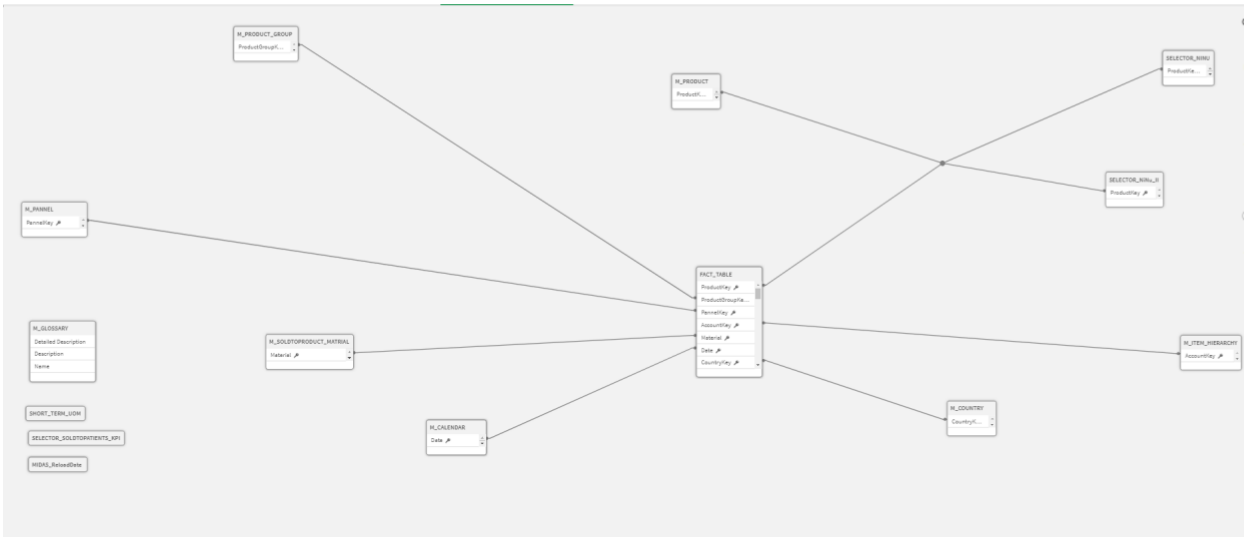


Figure 6.3: Data Model

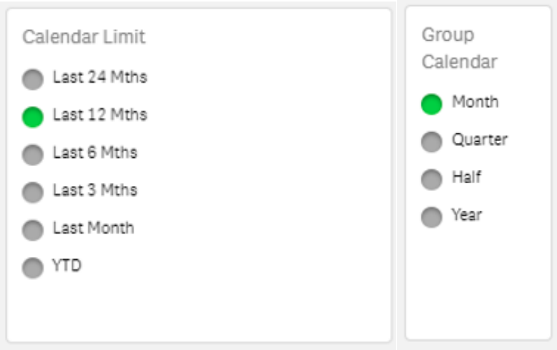
6.5.2. CREATING THE CALENDAR LOGIC

Before starting with the application there is some back-end logic that needs to be created, the calendar logic. This logic is used in all applications, and with some slight differences, the calculations remain the same. This is an essential step since it affects all the visualizations.

In the dashboard, the calculations are used to:

1. Limit the visualizations that show temporal data (only showing the last 3, 6, 12, 24 months, or the year to date). The selector allows users to only see data the user considers relevant.
2. Group the data into Month, Quarter, Half, or Year. Having this option makes it possible for the user to compare the data in the most suitable time slot, eliminating the need to have the same visualization repeated but with a different X-axis.

Both reduce the unnecessary noise and repetition and make the user's experience better and more complete.



Calendar Limit

○ Last 24 Mths
● Last 12 Mths
○ Last 6 Mths
○ Last 3 Mths
○ Last Month
○ YTD

Group Calendar

● Month
○ Quarter
○ Half
○ Year

Figure 6.4: Calendar Limit and Group calendar in the application

To create this logic the following calculations were done:

- **Time-period for the Calendar Limit:** The maximum date is calculated and to that the Calendar Limit value is subtracted.

- **Group Calendar to display:** A variable is created for each option (Month, Quarter, Half, and Year) where the data is grouped considering the number of months each combination has. Then they are placed in the inline, which is explained in the chapter below.

6.5.3. CREATING THE INLINE

The line is the text file where the selectors are created. For this app, the Units, Calendar Limit, and Calendar Group are included.

Qlik provides the option to add selectors directly in the app without any further calculations by inserting the dimension. For example, for a binary variable (Target: Yes or No) the selector would display those two options. However, in some cases, additional calculations are required.

For this reason, we create a table by loading it on the inline. The table is required to have at least two columns: the names that will appear in the selection (this field is added as the dimension in the selector) and the value to use in the calculation.

For the Units selector the there are two columns and rows. Both columns have the same values, Cash and Volume. The first column is used in the selector, and the second is used for the variables that are affected by the selector. The measure that is being aggregated is no longer Cash or Volume, but the inline field (Units), and it changes considering the value selected.

For the Calendar Limit, there are three columns:

- **Period:** The time displayed in the selector (e.g., Last 24 Months)
- **Flag Temp:** The number of months to subtract to the maximum month (e.g., 24)
- **DisplaySecondLabel:** The label displays the time-period shown (e.g. [Mar 2019 – Mar 2021]). However, when last month is selected it only makes sense to display one month (e.g. [Mar 2021]). In the label, a pick match is created so the second label is only shown if the condition is applied.

For the Group Calendar, the most important columns are the:

- **PeriodType:** The time type displayed in the selector (e.g Month)
- **CalendarDimension:** Variables previously created to display the x-axis. When Month is selected, the variable is just the field ‘MonthYear’, however for Year, for example, a calculation needs to be done to extract the year and we add ‘Y’ before it.
- **DisplayLabel:** Same logic as the ‘DisplaySecondLabel’ above.
- **MonthsDifference:** The number of months to subtract to the maximum month (e.g 24)

Once the inline is completed, the calculations can be done in the Variable’s excel. For the Calendar Limit, the *Flag Temp* (e.g 24) is subtracted to the maximum date month (just a simple *Max(YearMonth)* with *[CalendarDifference]={0}* in the set analysis). For the calculations affected by the selector, the date is restricted between the maximum date and this new variable.

As for the Group Calendar, new variables need to also be created. First, a calculation to group the data considering the selected values is done. Then, a variable for each option is created with the correct formatting (for example for Quarters ‘Q’ is added before the value). The variables are added to the *CalendarDimension* field in the inline. Which is the one used as the dimension for the temporal charts.

This means there are two ways to calculate the dates that will be displayed. As an implication, two different variables need to be created for most KPI’s. One to represent the metrics that are used in visualizations that don’t have a time field, and one for the visualizations that do.

In figure 5.5 we can see both examples. The first only displays the most recent month, whereas in the second figure we have a time evolution chart. The calculations, in this case, are a bit more complicated because they depend on two factors, what is selected on both the ‘Calendar Limit’ and the ‘Group Calendar’.

The field in the X-axis can’t just be the Month Year, because when, for example, the Quarter is selected we expect the axis to display ‘Qx’. So, as mentioned, the variable created for Calendar Dimension is used.

As for the time period to be shown (e.g Last 24 months), the restrictions occur in the set analysis. Both these time restrictions are incorporated in the KPI’s by default.

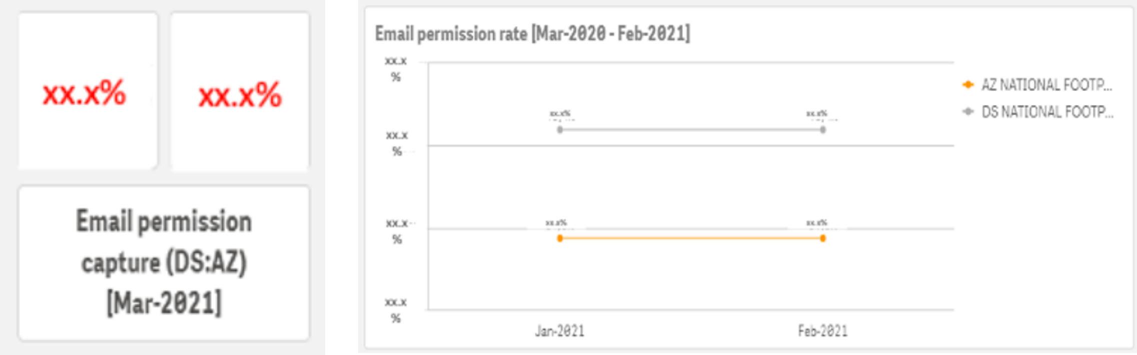


Figure 6.5: Measures with and without temporal data

For one of the sheets, the Sub National View Sheet, there is the need to create another variable for the same KPIs. The measures are displayed in a bar chart, so they don’t have a time-period associated. However, the sheet has the ‘Group Calendar’ selection in it. So, unlike the first example where only the most recent month is shown, in this case, it’s also possible to choose the most recent quarter or the YTD.

The logic implemented is a bit different from the one above for ‘Group Calendar’ because we are not using it as the dimension field, but rather as the time-period (1 month, 2 months, number of months for that year). So, the selection works more like the Calendar Limit. As a matter of fact, the logic implemented is the same as that one. The difference is the field used to subtract the months is corresponding to the ‘Group Calendar’ selection (*MonthsDifference*.)

6.5.4. CREATING THE KPI'S

Once the data is loaded into the app, the visualizations, pre-defined in the mock-up, can finally be created.

Before diving into how the formulas are created, there are some important facts to mention. A concept that will be brought up a lot throughout this chapter is 'Set Analysis'. It makes it possible to have selections in the variable that won't be affected by the filters in the dashboard. For example, if in the Set Analysis it's added *[Year]={'2021'}*, it doesn't matter if the user selects the year 2020, the measure will only show data for 2021. However, if we have *[Year]=max(Year)*, the measure will only have data for the maximum year, so if the user doesn't make any selection the measure will only show 2021, if the user selects 2020 (or 2020,2019 and 2018), the measure will only show 2020 data because it's the maximum value selected.

One Set Analysis that is present by default is the *[CalendarDifference]={0}.* The concept can be hard to understand at first, but ultimately the goal is to make variable creation easier. The field calendar difference makes it possible to create variables that cover more than a month. For example, if the user required a variable with a 4-month coverage, by adding *[CalendarDifference]={"<=3"}* no further calculations would need to be done. Adding the Month Year in the set analysis wouldn't be an option because the user doesn't want the calculation done only for the four most recent months. Rather, for each month the calculation to include a four-month period (June should include June, May, April, and March).

Also in the set analysis, the Source of the data is added by default (e.g. *Source={4}*). This is done because the final tables are often results of concatenation, and the source guarantees only the data we want will be considered.

Finally, as mentioned previously, the *Sum()* is used, as opposed to *Count()* to improve performance. Counters are created for this purpose.

Below the definition of each KPI and how they were created is explained. In the annexes, the actual formula used is attached.

- **% Face to Face Meetings:** Percentage of Face-to-Face Meetings compared to the other non-virtual meetings.
  The measure is created by dividing the sum of all the non-virtual face-to-face meetings, using in the set analysis two flags for this purpose (*FlagF2F={1}*, *FlagVirtual={0}*). Then dividing it by all the types of non-virtual meetings (so only with *FlagVirtual={0}*). An additional set analysis is added, *[Flag Temp]={1}*], this field is '1' when it's a past date. This way it is guaranteed future meetings are not being accounted for.

- **4 months coverage of targets:** Percentage of customers that have had a visit (non-virtual) between a 4-month period. The measure is calculated by diving the sum of customers *(Customer_Counter)* that have been visited (*FlagVirtual={0}, Event_Counter={1}*) by all customers. In order to guaranty the 4-month coverage the *[CalendarDifference]={"<=3"}* is added to both expressions.

In the denominator's set analysis we have *Source={991}* (the M_CALENDAR source with all the dates), this was added after it was noticed customers who didn't have a date associated (hadn't made a purchase or had any events scheduled) were not being counted.

- **A+B Hospital Purchasing:** Percentage of A+B Hospitals that have purchased stock within 3 months. These segments are given to the biggest hospitals, with the most purchasing potential.
  Again, the done with the *Hospital Counter*. The measure only accounts for hospitals belonging to segment A or B and where the total purchasing amount is higher than zero (*Segment={'A', 'B'}, Cash={">0"}*). An additional requirement was added so the measure accounts for a 3-month period because some hospitals don't make purchases every month, and this would be a more reliable result. The same logic as above is applied and *[CalendarDifference]={"<=2"}* was added to the set analysis.

- **Average, Maximum, and Minimum Hospital Sales:** The names of the measures pretty much define them. The calculations however are made per hospital, as opposed to the total sales. This is done by using an aggregate function. For example, instead of the total average (summing the sales and dividing it by the number of orders), the function works by first calculating the average sales per hospital and then doing the average of that number for all hospitals. The measures are used to display the growth over time. The maximum and minimum were added later to consider the variations of the average, due to the difference in the size of the hospitals being compared.
  Inside the aggregation function set analysis we also have *[CalendarDifference]={"<=2"}* because the measures are meant to be used in the same chart as the 'A+B Hospital Purchasing'.

- **Email permission capture:** Percentage of customers who allow emails to be sent to them. The calculation is the division between the sum of customers that allow emails (*PermissionStatus={'Y'}*) divided by the sum of all customers. The field used is *Customer_Counter*. In the denominator's set analysis the *Source={991}* is added for the same reasons as mentioned above in the '4 months coverage of targets' measure.

- **Positive Formulary inclusion:** Each Hospital is given a formulary status (Green, Yellow, Red) to classify their purchasing behavior. The metric measures the positive formulary inclusion percentage. Ideally, all of them being Green.
  It is the division between the sum of hospitals classified as A or B *(Segment={'A', 'B'})* and with green status *(Formulary_Status={'GREEN'})* divided by the sum of all hospitals classified as A or B *(Segment={'A', 'B'}),* independent of their status.

- **Sales (Wholesaler to End User Sales YTD):** Number of sales, measured in volume within the year. Can be shown as a trendline to make the sales' development over time visible, with the possibility to compare it to the targets and convert it to cash.
  The metric is calculated using the sum function on the field 'Volume'. Because it's a YTD calculation, in the set analysis we limit the Year Month to only account for the maximum year selected. This is done by restricting the data to be between the most recent completed month (calculated as normal but with the month completed flag) and the first

month of that year. This was created by subtracting the necessary months with the *AddMonths()* function.

- **Sales versus Target:** Proportion of Sales compared to the target. Ideally, the sales should surpass the target. Allows to very easily determine if the expected sales are being achieved.
  As mentioned, there are two types of targets in the data. However, the measure is calculated the same for both. It is done dynamically so the field that is being summed is dependent on the user's selection in the 'Select Forecast Type' and in the 'Select Unit of Measures' (either Volume or Cash) boxes. This is done with the inline.
  For the metric, the numerator is the sum of the field with the set analysis limiting the *Target_Type* to the selected target, same as for the field (Cash or Volume). As for the numerator, is simply the sum of the selected field.

6.6. Presentation layer: Qlik Sense Dashboards

The final dashboard is presented below, composed of six sheets. Aforesaid, the application gathers insights regarding a new drug. The end-user is composed of the sales team, whose intention is to easily draw conclusions about the product's performance. Along with new information to facilitate decision making. Ultimately, resulting in more sales and higher customer satisfaction.

Although each sheet has a specific purpose, they share some functionalities. The data can be filtered by Company, Territory, Year, and Month Year for all of them, except Account View and Sub National View. Other filters are also included, for specific sheets.

There are also two Date related filters included in most of the sheets. The Calendar Limit where the default is set to '12 months', showing the data for the most recent year. And the Group Calendar, where the default is set to Months.

Because there are two types of calendar selectors an error message was implemented, to avoid incompatible combinations. For example, if the user selects 'Last Month' but also tries to select 'Year', the charts affected by both selectors will display a text informing that for 'Last Month' it is only possible to select 'Month' in the Group Calendar. The message is personalized for each Calendar Limit selection, so the user knows exactly what selections are possible. These calculations are done in the inline.

The data presented in the dashboard is confidential, so below the data of the visualization is covered.

6.6.1. KPI SHEET

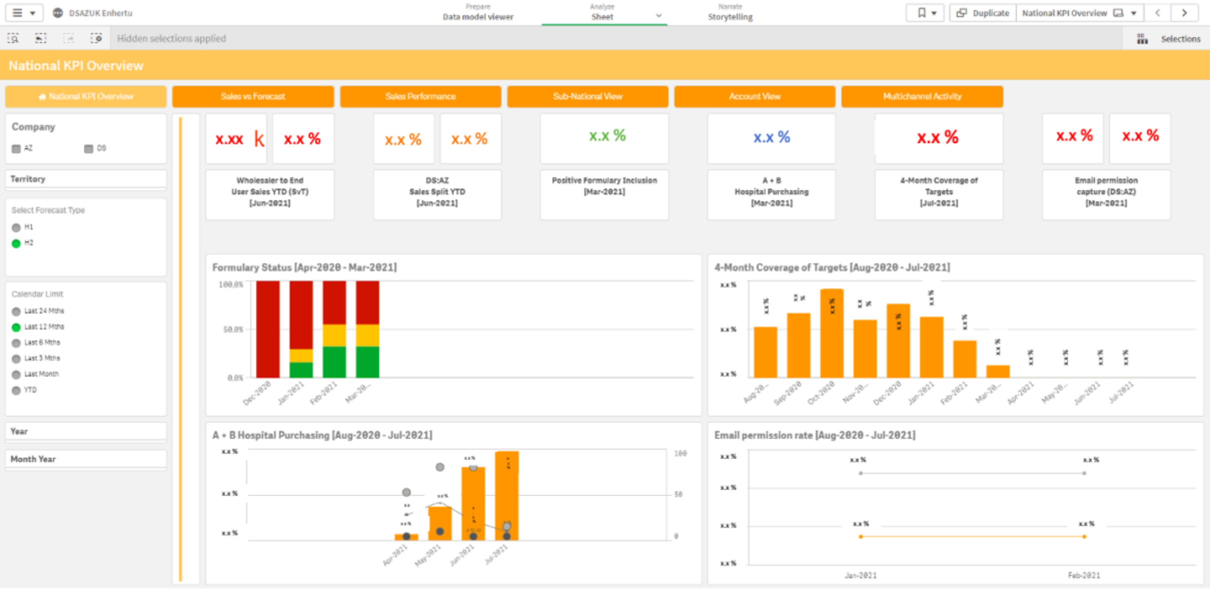


Figure 6.6: KPI Dashboard

The first sheet is called 'National KPI Overview'. As the name suggests, the main KPIs are presented in the sheet. They are laid out in an eye-catching way for simple information gathering. It allows to track progress towards strategic objectives and goals, as well as to measure the product's success.

For the KPIs shown in a square text, the most recent month is shown. They are colored either in green, orange, or red, depending on the expected result. Another way for the user to quickly understand the product’s performance. For the fourth KPI, no color system was assigned because there was no target defined.

Nevertheless, some metrics require an overtime analysis, available below the KPIs. In brief, the sheet serves as an overview, allowing to display the ‘bigger picture’.

Dashboards can sometimes provide too much information and even become confusing and off-putting, so it is helpful to have a sheet that summarizes the most important metrics. Keeping things simple so that decision-makers can focus on what’s important.

6.6.2. SALES VS FORECAST

The second sheet is called ‘Sales vs Forecast’. In it, two types of sales are presented. The second graph has a particularity, the forecast data is only available at a national level. To avoid confusing the user, when a territory is selected a subtitle appears explaining the data shown is not being filtered because it is not provided at that level of detail.

Moreover, the graph also has added security to it. Not all users are allowed to see it. If the permission isn’t granted then both the filter and the visualization are hidden.

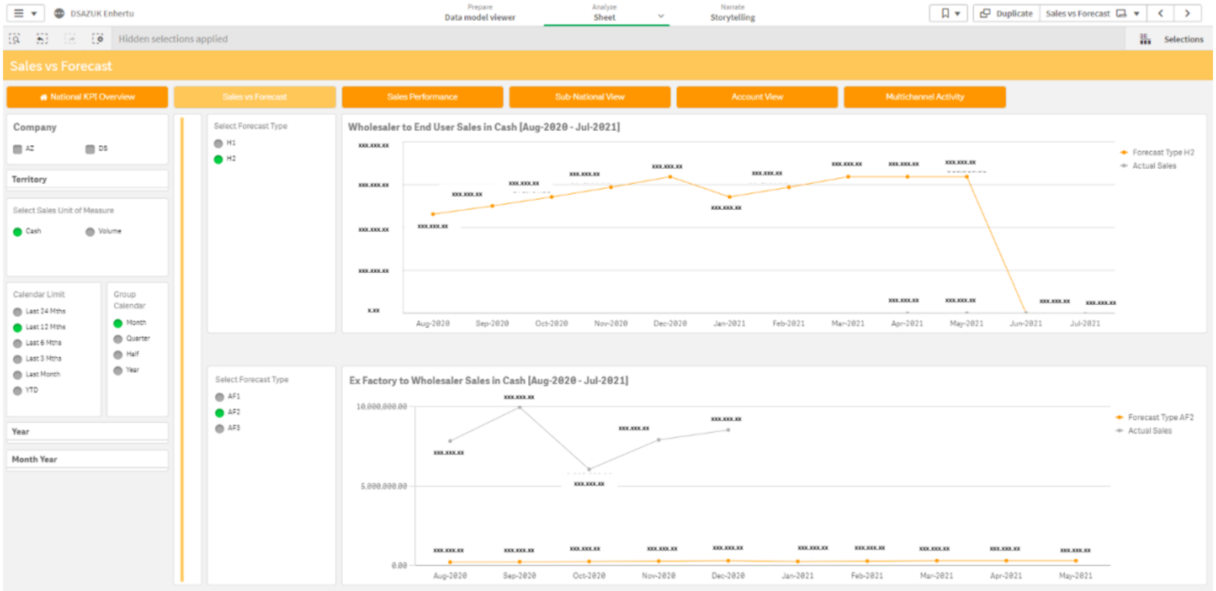


Figure 6.7: Sales vs Forecast sheet

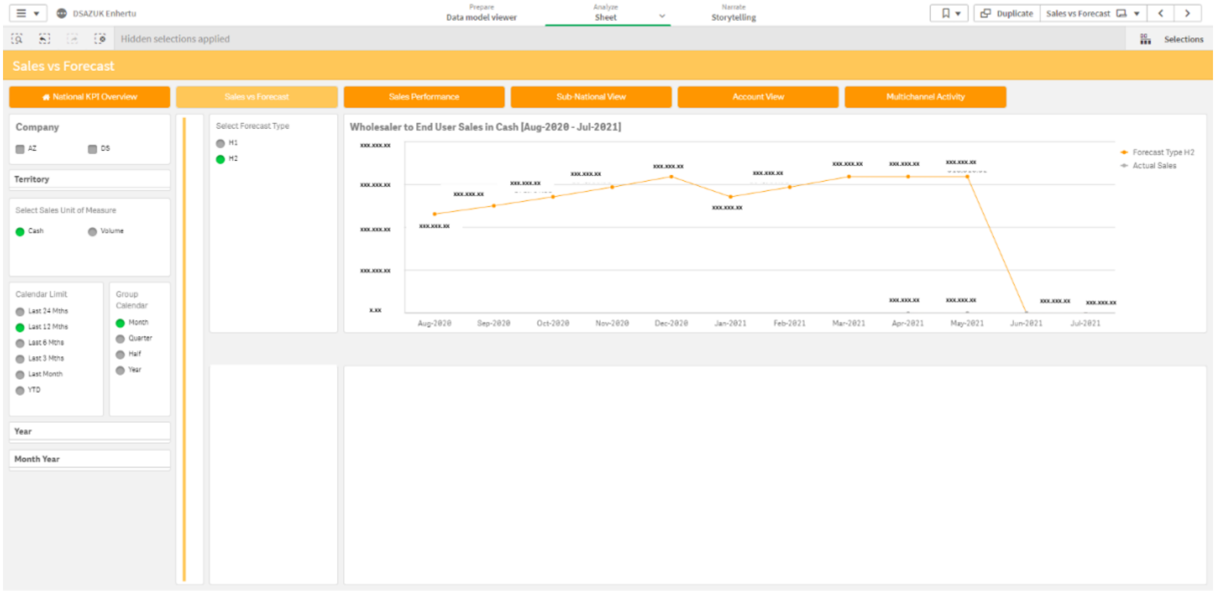


Figure 6.8: Sales vs Forecast sheet for users without permission for the second graph

Forecasting creates very insightful visualizations. With line charts, the users can understand how the sales are naturally evolving compared to what was expected from them. Especially talking about a new product, a Sales versus Target analysis can determine its continuity. It can, among other things, help to determine production volumes, budget, and policy decisions.

6.6.3. Sales Performance

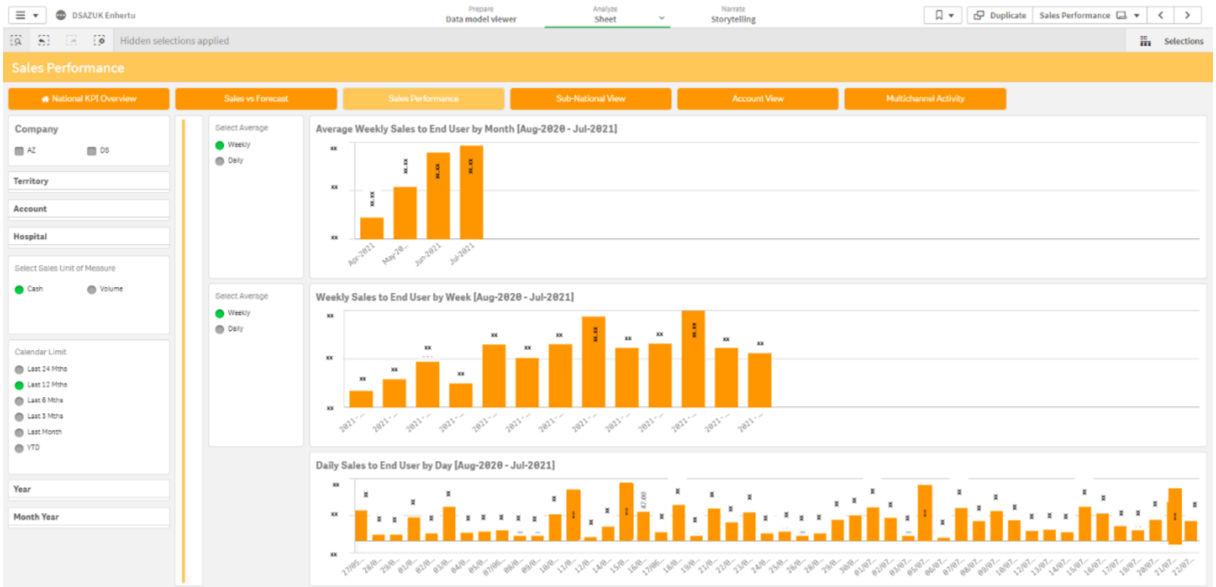


Figure 6.9: Sales Performance Sheet

The purpose of this sheet is to show the average sales over time, instead of their total. For the two top graphs, it is possible to show both the weekly and daily average by week and month. At the bottom, the total daily is shown.

Although this can seem repetitive, displaying the averages in the different time formats can help with understanding sudden changes and provided a more detailed analysis.

6.6.4. Sub National View

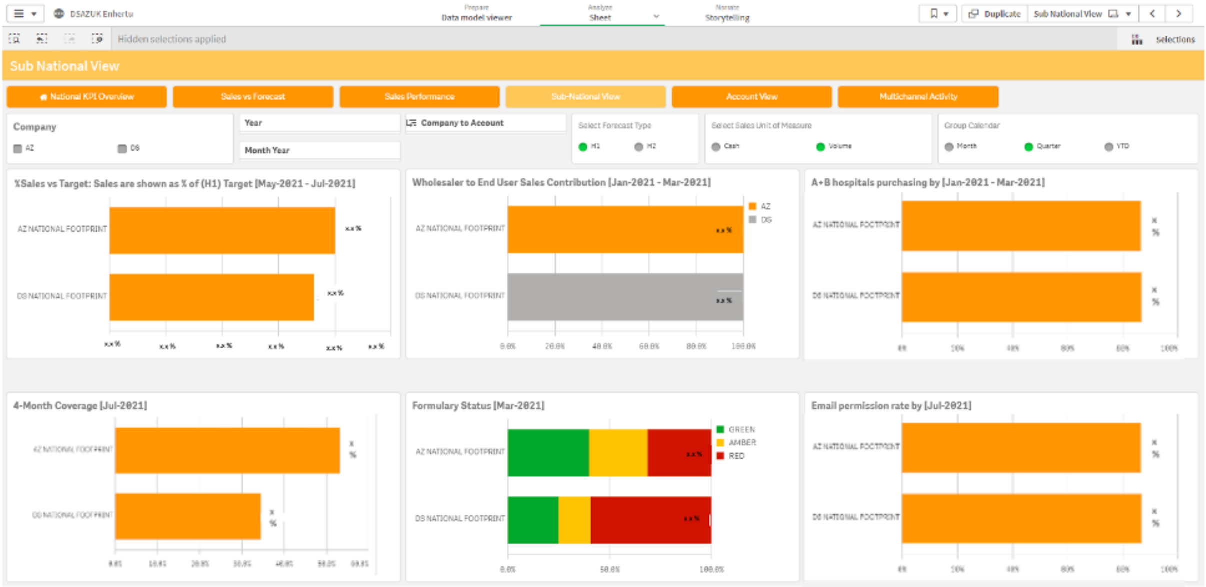


Figure 6.10: Sub National View Sheet

In this sheet, the same metrics displayed as KPIs in the Overview are shown again. However, now they are shown in a horizontal bar chart with changing dimensions.

This way, the user has the option to look at the data from a more detailed point of view. In previous sheets, the metrics are only shown as totals. With the drill-down selector ‘Company to Account’, the user has the option to choose between the dimensions and compare them with each other.

Above the Company’s dimension is being shown. When clicking on the selector the user can click on one of the companies. The displayed dimensions will now be regions (North, West, Midlands & East, and South East) for the selected company since it is the next granularity. The filtering continues until Account level.

This way it’s very simple to compare not only how both companies are performing, but also what regions and accounts need more attention.

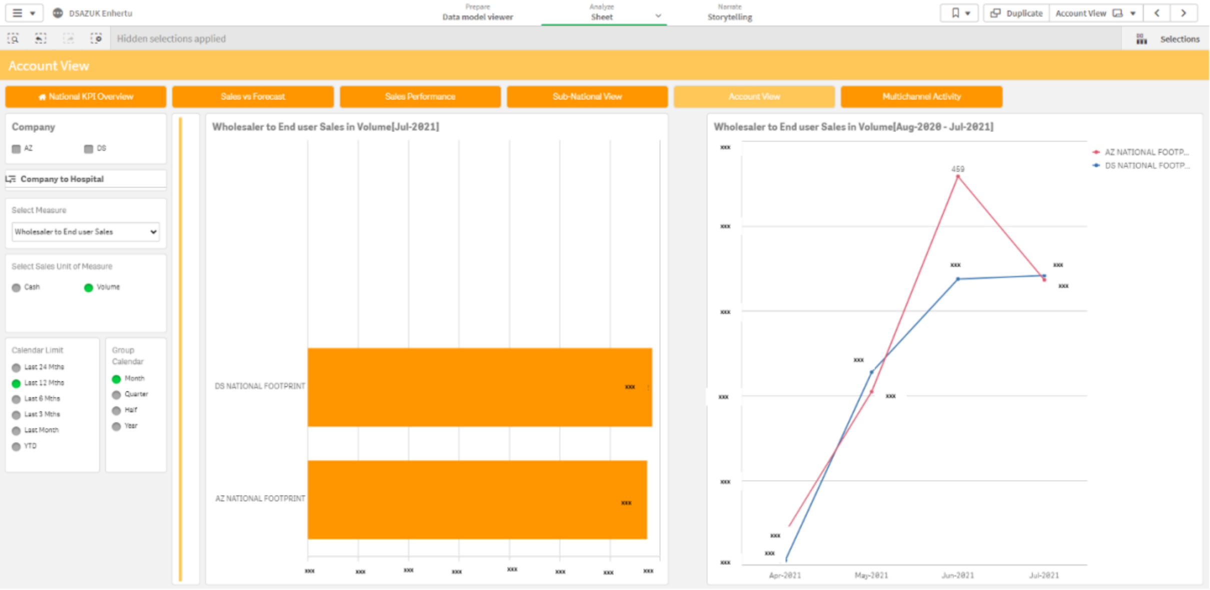
6.6.5. Account View

Although both visualizations have appeared in previous sheets, with Account View it is possible to have both analyses side by side, but now from Company to Hospital level.

Only two measures can be selected (in ‘Select Measure’) because they are the only ones with Hospital Granularity.

The first graph is similar to the Account View bar charts, the most recent data (considering Group Calendar) is shown. This way the user can quickly understand how the selected dimension is currently behaving.

Looking at the second graph, the user can see how the dimensions have been evolving over time.


Account View

Company
□ AZ □ DS

Company to Hospital

Select Measure
Wholesaler to End user Sales

Select Sales Unit of Measure
○ Cash ● Volume

Calendar Limit
○ Last 24 Mths
● Last 12 Mths
○ Last 6 Mths
○ Last 3 Mths
○ Last Month
○ YTD

Group Calendar
● Month
○ Quarter
○ Half
○ Year

Figure 6.11: Account View Sheet

6.6.6. Multichannel Activity

The sheet is divided into two visualizations. Both represent the distribution between the different channels over time. For the first one, all call and meetings type channels are displayed. An additional measure is also included, the percentage of face-to-face meetings. It was created so it could be easily seen if the sales team was fulfilling the expectations in this regard. For the second visualization, only digital channels are shown.

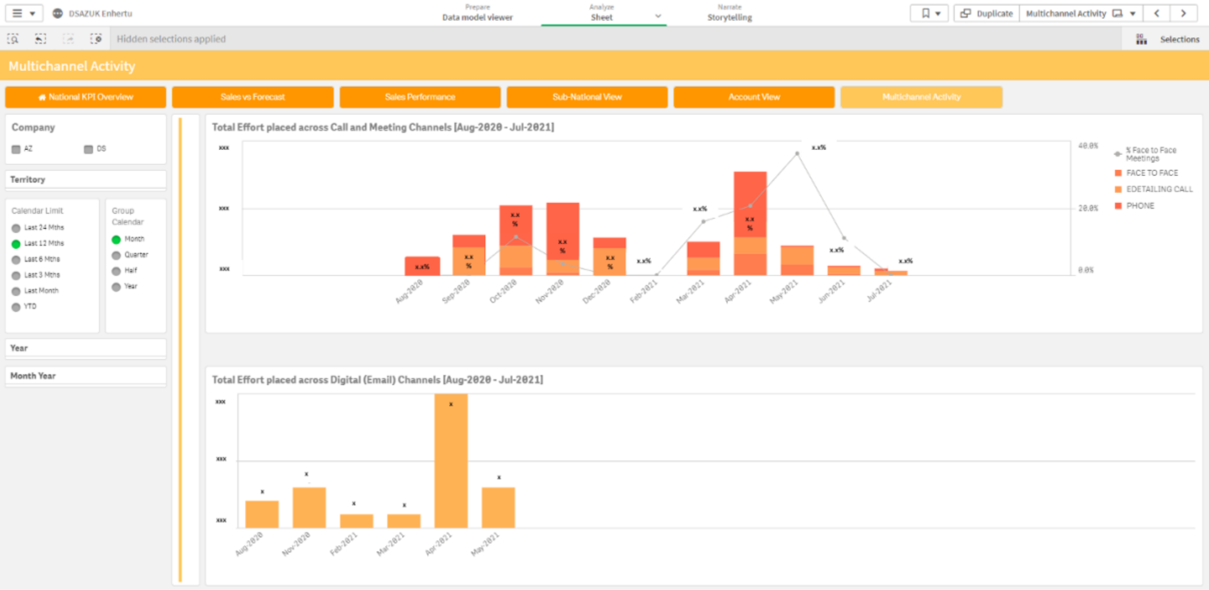
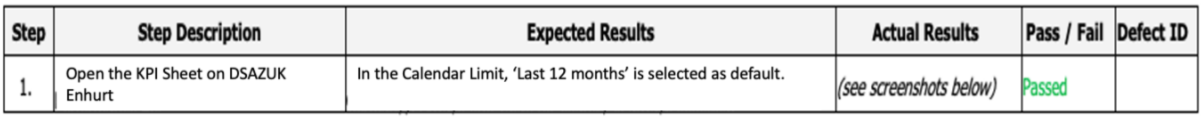


Figure 6.12: Multichannel Activity Sheet

6.7. Testing

In the testing phase, the developers have the responsibility to ensure all dashboard functionalities are as expected.

A member of the development team is responsible for the Test Scripts. In this document, all the User Stories are tested, and, in case of any failure, it is required to go back to the Development Phase for rework.



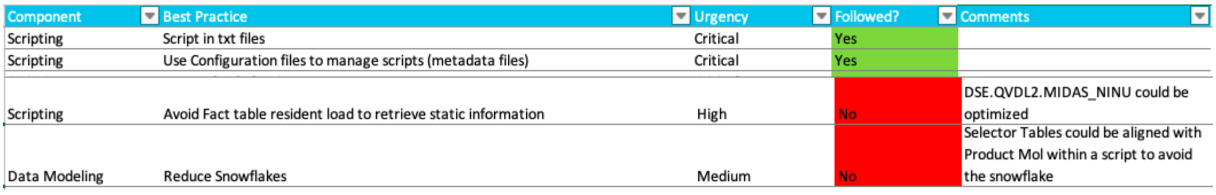
Step	Step Description	Expected Results	Actual Results	Pass / Fail	Defect ID
1.	Open the KPI Sheet on DSAZUK Enhurt	In the Calendar Limit, 'Last 12 months' is selected as default.	(see screenshots below)	Passed	

Figure 6.13: Example of a step in the Test Script

The visualizations, requirements, features, and measures are tested. This is done by comparing the results against the original source, normally the excel input files. The same calculations are re-done with the original data and the values should match. Also, the logic is reviewed to make sure the metrics are showing what is intended.

This process should be exhaustive and all-encompassing. In this document, the team is saying to the Product Owner that all the steps are performing as planned and the dashboard is ready for production. If the Test Scripts are not correctly done, something that is failing will only be detected after the end-users already have access to it. Which, as imagined, is far from ideal or even acceptable.

Afterward, an excel titled Code Review needs to be concluded. This document contains all the best practices that need to be followed. If a practice isn't followed a reason should be given. The document helps to ensure the app is as efficient as possible and that there aren't any solutions that can potentially cause problems in the future.



Component	Best Practice	Urgency	Followed?	Comments
Scripting	Script in txt files	Critical	Yes	
Scripting	Use Configuration files to manage scripts (metadata files)	Critical	Yes	
Scripting	Avoid Fact table resident load to retrieve static information	High	No	DSE.QVDL2.MIDAS_NINU could be optimized
Data Modeling	Reduce Snowflakes	Medium	No	Selector Tables could be aligned with Product Mol within a script to avoid the snowflake

Figure 6.14: Example of a followed and not followed best practice in Code Review

Finally, the Technical Design Specification is reviewed and updated if necessary. This internal document serves as a guide for the development team and new team members to know more about the dashboard, the files it contains, and how it is organized. Succinctly, it details the scope and purpose of the dashboard, its structure, and configuration files.

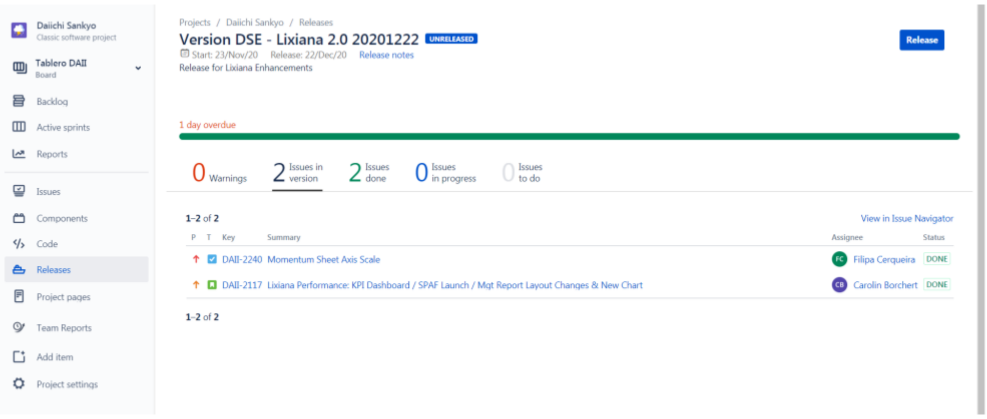
Finally, the documentation is finalized, and once all User Stories pass, the dashboard is ready for production.

6.8. MIGRATION

After testing is completed, the migration process starts. This process consists of moving the dashboard(s) from development to production. For this purpose, it is necessary to follow some steps.

Firstly, all User Stories need to be moved in JIRA to the Done status. As aforesaid, it is only possible to move to production once all the requirements are tested and proved to be functioning. Meaning they are 'done'. So, JIRA should be updated, in case any of the User Stories are not still in this state.

Next, it is necessary to create a release in Jira. This is done in the release section. The first step is to fill in the project information, such as name, small description, and release day. And then, all User Stories related to this project need to be linked to this release. This is done by filling the section 'Fix Version' in each User Story.



Projects / Daiichi Sankyo / Releases

Version DSE - Lixiana 2.0 20201222 UNRELEASED

Start: 23/Nov/20 Release: 22/Dec/20 Release notes

Release for Lixiana Enhancements

1 day overdue

0 Warnings 2 Issues in version 2 Issues done 0 Issues in progress 0 Issues to do

1-2 of 2 View in Issue Navigator

P	T	Key	Summary	Assignee	Status
↑	☑	DAII-2240	Momentum Sheet Axis Scale	FC Filipa Cerqueira	DONE
↑	🟩	DAII-2117	Lixiana Performance: KPI Dashboard / SPAF Launch / Mgt Report Layout Changes & New Chart	CB Carolin Borchert	DONE

1-2 of 2

Figure 6.15: Project Release in JIRA

Once this is completed, the project can be released. Since the User Stories are ‘done’ and will suffer no further changes, they are no longer in the active sprint dashboard with all the required tasks. By doing this the active dashboard will continue ‘clean’, only with User Stories that are related to projects still in production.

Lastly, the dashboard can be move to production. For migration purposes, Qlik QMC is used to export applications. The exported app contains all the development’s functionalities. However, the actual data needs to be in production as well. For this, it is necessary to copy the Application Layer Containers where the data and data model are created.

This is done by first, copying the current folders into a subfolder, the ‘07.Relase’, with the sprint's name and release date. This is where the previous sprints are stored.
Afterward, the new files can be pasted into production. Now the application will have the most recent changes in the data model.

Again, in Qlik QMC, but this time in PROD, the application is imported and then publish in the correct stream, which should be the same as in DEV.

Lastly, as mentioned, the dashboard needs to have the same data as in development, so it needs to also be loaded every time the data is updated, aka the second QVD Layer is extracted. For this reason, the same tasks are also repeated in production.

7. CONCLUSIONS

The main purpose of this report is to explore the importance of implementing business intelligence to modern businesses and how it can, ultimately, be used to make better decisions.

Particularly, it focuses on two companies, who collaborated on the creation of a drug, and their need to have a dashboard display the performance of the said product. Besides the most obvious metrics, such as sales or number of target customers, it was also needed to see the distribution of those metrics between the two companies.

Despite limitations, the project fulfilled the clients' expectations. Gathering the development team's previous business knowledge on the pharmacy industry and the clients' preferences, the team was able to deliver a dashboard that combined what was originally agreed with new functionalities.

As for the project's next steps, the following sprint has already started. New measures and visualizations that couldn't be added due to time constraints are being included. The user's feedback from the Hyper-care period is also being considered. With these new improvements, it's expected for the dashboard to be even more user-friendly and to provide more information whilst reducing data redundancy.

The report highlights the lessons learned in the internship in conjunction with the master's degree. Firstly, working in a professional environment was a new challenge. With extra responsibilities being added as the project progressed, I learned how to be more independent and resourceful when it came to problem-solving. The internship also allowed me to be in direct contact with the business side, this has made me more able to communicate ideas and manage expectations with clients.

Additionally, I was able to understand how everything that goes behind creating a dashboard is built. With previous knowledge in SQL Server, creating data models wasn't something foreign to me, however, because of the different formats and granularity levels, I was able to deepen my knowledge in cleaning, transforming, and building data models. During the internship, I also learned how to use Qlik, which is a great asset to have.

In conclusion, the internship was a great way to start my professional carrier and to dive more into Business Intelligence. In the future, I hope to continue exploring this path further.

8. LIMITATIONS AND RECOMMENDATIONS FOR FUTURE WORKS

The project took place in a work environment that followed the Scrum Methodology, so sharing knowledge was very important. However, since I had no prior knowledge of Qlik, in the beginning, the tasks given to me didn’t match my levels of expertise. This required great support from the rest of the team.

Again, given the Scrum Methodology, each project lasted about three weeks. Considering the client's expectations and demands, it became a very short amount of time for all the deliverables. This sometimes meant not being able to answer all the clients’ requests.

Along the same lines, managing the clients’ expectations regarding Qlik visualizations was also proven difficult at times. Every tool has its limitations, so it was up to the team to compensate for them and find workarounds or alternatives to fulfill the initial vision.

Lastly, the ‘less is more’ concept. It was challenging to find the balance between extracting all the information possible from the data whilst not overloading the dashboard with irrelevant metrics. The end-users feedback facilitated this. Having regular meetings so they could provide input and describe what they would use, was very helpful.

9. BIBLIOGRAPHY

«About us». *SDG Group*, (URL: https://www.sdggroup.com/pt-pt/about-us/sdg, consulted in 17-10 20).

About Us - Daiichi Sankyo. (URL: https://www.daiichisankyo.com/about_us/. , consulted in 17-10-20).

Business Intelligence in Pharma | A Key Enabler of Industry Transformation. (URL: https://www.pharmafocusasia.com/information-technology/business-intelligence-pharmaceuticals, consulted in 21-06-21).

Kascelan, Ljiljana (2011). Advantages and Limitations in Implementation of Business Intelligence System in Montenegro: Case Study Telenor Montenegro, Economic Review: Journal of Economics and Business, Vol. 09, Iss. 2, pp. 19-30

Kimball R., Ross M. The Data Warehouse ToolKit. The Definite Guide to Dimensional Modeling, Third Edition

Negash S., Gray P. (2008) Business Intelligence. In: Handbook On Decision Support Systems 2. International Handbooks Information System. Springer, Berlin, Heidelberg.

Sabherwal R., Becerra-Fernandez I. (2013) Business Intelligence: Practices, Technologies, And Management. Pg. 9-10

«Data Science vs Business Intelligence | Top 20 Comparisons To Learn». *EDUCBA*, 2018, (URL: https://www.educba.com/data-science-vs-business-intelligence/., consulted in 21-10-20).

«Data Warehouse Environment». *Qlik*, (URL: https://www.qlik.com/us/data-warehouse/data-warehouse-environment. consulted in 17-10-20).

«How Data Analytics Can Improve the Pharma Industry? | GreenM™». *GreenM™ – Global Health Data Innovation, Architecture and Design Company* 2020, (URL: https://greenm.io/how-data-analytics-can-improve-the-pharma-industry/. consulted in 21-06-21).

GreenM^{TM} – Global Health Data Innovation, Architecture and Design Company, 2020, (URL: https://greenm.io/how-data-analytics-can-improve-the-pharma-industry/., consulted in 22-06-21).

Inc, BizXcel. «6 Reasons Why Qlik Sense Makes Sense». *BizXcel*, (URL: https://www.bizxcel.com/blog-post/6-reasons-why-qlik-sense-makes-sense. , consulted in 20-10-20).

Karim, Akram Jalal. *The value of Competitive Business Intelligence System (CBIS) to Stimulate Competitiveness in Global Market*.

ltd, Research and Markets. *Healthcare Business Intelligence (BI) - Global Market Trajectory & Analytics*. (URL: https://www.researchandmarkets.com/reports/5302573/healthcare-business-intelligence-bi-global. , consulted in 22-06-21).

«MarketScope for Business Intelligence and Information Management Services in Western Europe». *Gartner*, (URL: https://www.gartner.com/en/documents/2487221/marketscope-for-business-intelligence-and-information-ma. , consulted in 20-11-20).

Maynard-Smith, Julian. *Ultimate Guide to Business Writing: All the Secrets of Creating and Managing Business Documents*. 1.ª ed., Routledge, 2021.

Moss , Larissa T. , e Shaku Atre. *Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications*.

«Qlik Sense and QlikView Earn 2020 Top Rated Awards From TrustRadius». *Qlik Blog*, (URL: https://www.qlik.com/blog/qlik-sense-and-qlikview-earn-2020-top-rated-awards-from-trustradius. consulted in 22-10-20).

What Is ETL (Extract, Transform, Load)? (URL: https://www.ibm.com/cloud/learn/etl. consulted in 17-10-20).

W. Eckerson, *Smart Companies in the 21st Century: The Secrets of Creating Successful Business Intelligent Solutions.* The Data Warehousing Institute, Seattle, WA, 2003, p. 121

Larson B., *Delivering Business Intelligence with Microsoft SQL Server* 2016, p.35

Ljiljana K.,2011, *Advantages and Limitations in Implementation of Business Intelligence System in Montenegro: Case Study Telenor Montenegro*, pp. 19-30

Ziama and J. Kasher, 2004, *Data Mining Primer for the Data Warehousing Professional*. Dayton, OH: Teradata.

Yeoh, William, Koronios, Andy and Gao, Jing 2008, Managing the implementation of business intelligence systems: a critical success factors framework, *International Journal of enterprise information systems*, vol. 4, no. 3, pp. 79-94

Kimball, R., & Ross, M., 2016, *The data warehouse toolkit, p.20*.

Negash, Solomon and Paul Gray., 2008, "Business Intelligence." Handbook on Decision Support Systems 2: Variations, p. 175-193.

Garani, Georgia and Sven Helmer., 2012, *Integrating Star and Snowflake Schemas in Data Warehouses, p.27*.

TechValidate, 2015, Case Study: Mission Health System (URL: https://www.techvalidate.com/product-research/qlik/case-studies/1BF-84A-E1A, consulted in 25-07-21).

Eckerson, W., 2013, *Performance dashboards*. Hoboken, N.J.: Wiley, p. 3-22

10.APPENDIX

Qlik Sense Components

- **Sheets:** Can contain graphics, tables and filters with specific information. An App consists of one or several sheets.
- **Fields:** Hold the used data, loaded from the script. They are the data model columns.
- **Dimensions:** Fields create from tables of data models whose values contain descriptive information for each record. And they determine how the data is grouped in a visualization.
- **Measures:** Expressions composed of aggregation functions, such as SUM or MIN, with one or several fields. For example, the sum of sales for this month.
- **Variables:** Contains a unique value or expression and it's especially useful when it is repeated several times. To facilitate readability the variable's name always has the same name format. They start with a 'v.' followed by the variable type (KPI, Auxiliar variable, Label, …), the name of the report, and the actual name of the variable. For example, v.KPI.Dashboard.MarketShare. As mentioned in the document the variables are stored in excel and then uploaded into the app, to centralize KPIs and avoid extra calculations that can slow down the application.
- **Master Items:** Master Items have a similar logic to why we use variables. We can save visualizations, measures, and dimensions and reuse them throughout the application. Besides the value, the formatting type and appearance are also saved. So, changing something in the master item will affect all the sheets it is inserted in. This is very useful for metrics, when they are used several times, this way if there is a need to change the expression, instead of needing to change it every time it was used, it is simply needed to be changed in the master item. And to have drill-down dimensions.
- **Drill down dimensions:** Several dimensions can be included, connected hierarchically, selecting a record of Level1 and going down to Level n. For example, in a Calendar filter, with the first dimension being year and the second month year.

Flat Files: Flat files are plain text format databases, where each line represents a record. They are mentioned throughout the report, mainly referring to Excel files.

Metadata: Can be referred to as data about other data, such as administrative or structural data.

11.ANNEXES

Most important measures used to create the dashboard:

v.KPI.Overview.PositiveFormularyStatus	Sum({< Segment={'A','B'}, Formulary_Status={'GREEN'}, Source={2},YearMonth={"<=$(=$(v.Aux.General.LatestPeriodInitialBySource(2))) >=$(=$(v.Aux.General.LatestPeriodFinalBySource(2)))"}>}Hospital_Counter) /Sum({< Segment={'A','B'}, Source={2},YearMonth={"<=$(=$(v.Aux.General.LatestPeriodInitialBySource(2))) >=$(=$(v.Aux.General.LatestPeriodFinalBySource(2)))"}>}Hospital_Counter)
v.Aux.General.Latest4Months	Date(Addmonths($(v.Aux.General.CurrentMonthMax),-4), 'YYYYMM')
v.Aux.General.CurrentMonthMax	date(Today(), 'YYYYMM')
v.KPI.Overview.4monthtargetcoverage	sum({<DeliveryChannel={'FACE TO FACE', 'REMOTE DETAILING'}, Event_Counter={1}, [Calendar Difference]>={3}>}Customer_Counter)/ sum({<Event_Counter={1}, [Calendar Difference]={4}>}Customer_Counter)
v.KPI.Overview.NationalSalesByProductAndPeriod	sum({<CalendarDifference={0}, Source={$1}, ProductID={$2}, Product=, Region=, Territory=, [CCG-HB-HSCP]=, GuidanceStatus=, Account=, Brick=,MonthYear=, YearMonth={"<=$(=$(v.Aux.General.LatestPeriodInitialBySource($1))) >=$(=$(v.Aux.General.LatestPeriodFinalBySource($1)))"}>}[$3])
v.Aux.General.LatestPeriodInitialBySource	max({<$(v.Aux.General.CalendarExpression), Source={$1}>}YearMonth)
v.Aux.General.LatestPeriodFinalBySource	date(Addmonths(Date#($(v.Aux.General.LatestPeriodInitialBySource($1)), 'YYYYMM'),-$(=%MonthsDifference)),'YYYYMM')
v.Aux.General.CalendarExpression	CalendarDifference={0}
v.Dimension.Sales.TemporalCharts	ceil(((Year(date#(MonthYear, 'MMM-YYYY'))*12 + Num(Month(date#(MonthYear, 'MMM-YYYY')))) - (Year(date#($1, 'YYYYMM'))*12 + Num(Month(date#($1, 'YYYYMM'))))) /$2)
v.Dimension.Sales.QuartersSheet1	'Q'&$(v.Dimension.Sales.TemporalCharts($(=$(v.Aux.Overview.MaxYearConditionRXA_DRX)), 3))
v.Dimension.Sales.HalfsSheet1	'H'&$(v.Dimension.Sales.TemporalCharts($(=$(v.Aux.Overview.MaxYearConditionRXA_DRX)), 6))
v.Dimension.Sales.YearsSheet1	'Y'&$(v.Dimension.Sales.TemporalCharts($(=$(v.Aux.Overview.MaxYearConditionRXA_DRX)), 12))
v.Dimension.Sales.QuartersSVT	'Q'&$(v.Dimension.Sales.TemporalCharts($(=$(v.Aux.LixianaPerformance.MaxMonthYearSVT)), 3))
v.Dimension.Sales.HalfsSVT	'H'&$(v.Dimension.Sales.TemporalCharts($(=$(v.Aux.LixianaPerformance.MaxMonthYearSVT)), 6))
v.Dimension.Sales.YearsSVT	'Y'&$(v.Dimension.Sales.TemporalCharts($(=$(v.Aux.LixianaPerformance.MaxMonthYearSVT)), 12))