INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Analytical Enrichment: A Target To Source Approach for Missing Requirements in Decision Support Systems

João Maria Sousa Gomes Neto

Master in Computer Science

Supervisor:
PhD José Eduardo de Mendonça Tomás Barateiro ,
Invited Assistant Professor,
LNEC, ISCTE-IUL

Co-supervisor:
PhD Elsa Alexandra Cabral da Rocha Cardoso,
Assistant Professor,
ISCTE-IUL

November, 2022

TECNOLOGIAS
E ARQUITETURA

Department of Information Science and Technology

Analytical Enrichment: A Target To Source Approach for Missing Requirements in Decision Support Systems

João Maria Sousa Gomes Neto

Master in Computer Science

Supervisor:
PhD José Eduardo de Mendonça Tomás Barateiro,
Invited Assistant Professor,
LNEC, ISCTE-IUL

Co-supervisor:
PhD Elsa Alexandra Cabral da Rocha Cardoso,
Assistant Professor,
ISCTE-IUL

November, 2022

Acknowledgements

I would like to first and foremost thank both my supervisors, Professor José Barateiro and Elsa Cardoso, for their amazing effort in helping me accomplish this dissertation as well as for keeping on believing in this project even though it took a long time to be finished. I will be forever thankful for allowing me to take my time while trying to invest in both my education and professional life.

I also want to give a special thank you to my good friend António Lorvão for giving me as much motivation to keep on working as well as to also do the very important task of beer drinking afterwards. All jokes aside, I couldn't have done it without him, and his help throughout the process was unbelievable.

Last but not least, I want to thank my Family, Girlfriend and Friends for all the motivation and for making my Academic path better than I could have hoped for. These Years at ISCTE have been amazing, and I wouldn't have wanted it any other way.

Resumo

Os sistemas operacionais recolhem dados transacionais e apoiam a execução de processos de negócio numa organização. Estes sistemas são frequentemente a fonte de dados para os Sistemas de Apoio à Decisão (SAD), ou seja, sistemas analíticos concebidos para auxiliar os utilizadores empresariais no processo de tomada de decisão. Por esta razão, vários problemas nos Sistemas Operacionais, tais como requisitos de dados em falta ou questões de qualidade de dados, podem levar a necessidades analíticas não satisfeitas do SAD e, consequentemente, ter um efeito negativo no Processo de Tomada de Decisão, uma vez que as questões de negócio relevantes podem não ser respondidas.

O objetivo do presente estudo é compreender o impacto da integração dos requisitos do SAD na conceção dos sistemas operacionais. Para atingir este objetivo, esta dissertação utiliza um caso de estudo real de um SAD para identificar os requisitos em falta e desenvolver um SAD protótipo para demonstrar o impacto positivo no processo de Tomada de Decisão quando estes requisitos são cumpridos. Ao longo deste desenvolvimento, as formas de lidar com os vários tipos de requisitos em falta serão abordadas. É também proposto um método de avaliação para compreender e categorizar os requisitos em falta e a forma como podem ser tratados. Além disso, o método de avaliação é aplicado, e o protótipo desenvolvido é comparado com o sistema de base, no sentido de medir o impacto.

Finalmente, são mostrados os benefícios desta integração, bem como outros fatores que também podem limitar os requisitos do SAD.

Palavras-Chave: Data Warehouse, Business Intelligence, Requisitos em falta, Sistemas de Apoio à Decisão, Sistemas Operacionais

Abstract

Operational Systems collect transactional data and support the execution of business processes in an organization. These systems are often the data source for Decision Support Systems (DSS), i.e. analytical systems designed to aid business users in the decision-making process. For this reason, several problems in Operational Systems, such as missing data requirements or data quality issues, can lead to unfulfilled analytical needs of the DSS and, consequently, have a negative effect on the Decision Making Process since relevant business queries may not be answered.

The objective of this study is to understand the impact of the integration of DSS requirements in the design of operational systems. To achieve this objective, this dissertation uses a real use case DSS to identify the missing requirements and develop a DSS prototype to demonstrate the positive impact on the Decision-Making process when these requirements are fulfilled. Throughout this development, ways of dealing with the various types of missing requirements are going to be addressed. Additionally, a methodology to evaluate the missing requirements is suggested, along with a proposal to classify and understand the missing requirements and how they can be dealt with. Also, the evaluation method is applied, and the developed prototype is compared to the baseline system in order to measure the impact.

Finally, the benefits of this integration are shown, as well as other factors that can also constrain the DSS requirements.

Keywords: Data Warehouse, Business Intelligence, Missing Requirements, Decision Support System, Operational System

Contents

Table of Figures

Acronyms

ETL - Extract, Transform and Load processes

BEAM - Background, Exhibit, Argument, Method

BI – Business Intelligence

DSS - Decision Support Systems

DSRM - Design Science Research Methodology

DW – Data Warehouse

ERP – Enterprise Resource Planning

HR – Human Resources

SMART - Specific, Measurable, Attainable, Realistic and Time-sensitive

SQL - Structured Query Language

XP - Extreme Programming

1. Introduction

1.1. Motivation and Context

Operational Systems collect transactional data and support the execution of business processes within an organization, the so called day-to-day data, is therefore stored in this type of system. These systems are often the source of data to the Decision Support Systems (DSS), which are purely analytical systems designed to be able to answer business queries in a performant way. These systems aid the business users in the decision making process as they provide an interactive and intuitive presentation layer, displaying the data commonly through graphic visualizations.

Decision support systems depend on both the existence and quality of data produced by the operational systems, as some of the challenges of the development of these systems are to identify the appropriate data sources and assure proper data quality [1].

Most Decision Support Systems development projects are only thought of or implemented once the operational systems have already been developed [2] therefore becoming constrained by them. “In a similar fashion, the DSS is constrained by the organization’s available technology (…). This includes data as well as computer power, and the base of reliable operational systems and technical expertise on which a DSS is built” [3, pp.3]. This becomes an issue since all the metrics and context that could be of interest to the decision makers to have available to them to visualize and support their decisions are already reduced to those that can be retrieved from the source system. Therefore, the decision-making is being made based on systems that could very well be missing key business information.

Addressing this problem covers several areas, ranging from software engineering, requirements engineering as well as business intelligence and its project development, including requirements collection to understand better the processes behind the development of both the operational and the DSS.

Typical operational systems were not designed to fulfil the data requirements of the DSS [2]. Thus, to support the analytical needs of decision-makers, the operational systems must be adapted to integrate those data requirements. Otherwise, the DSS will be developed using only the analytical capabilities that can be achieved using limited data from operational systems, bringing less insight to the decision-makers and being overall a less useful tool.

There is a need for a solution to mitigate these data requirement issues between both systems (operational and analytical) so that they can relate in a symbiotic way rather than having a “one-sided relationship.”

1.2. Research Questions and Objectives

The operational systems are not able to fulfil some of the data requirements of the DSS, as these last systems are mostly implemented and even taken into consideration by the companies or businesses long after the operational systems have been developed.

This dissertation intends to cover the dependencies between operational and analytical systems, addressing the following research questions:

- [RQ1] How does the integration of DSS requirements in the operational systems design impact the decision-making process?
- [RQ2] How does the integration of the DSS requirements in operational systems design affect the design of the latter?
- [RQ3] Why does the integration of DSS requirements in the operational system's design impact the decision-making process?

These are the main questions that this dissertation is going to assess with the objective of analysing how assessing DSS requirements in earlier stages can positively impact the fulfilment of the business analytic needs, proposing a requirements integration approach to improve the quality of the DSS.

1.3. Research Methodology

The Design Science Research Methodology (DSRM) will be used as a guideline for the development of the present dissertation. This methodology is composed of six steps to solving the perceived problem, as shown in Figure 1.1.

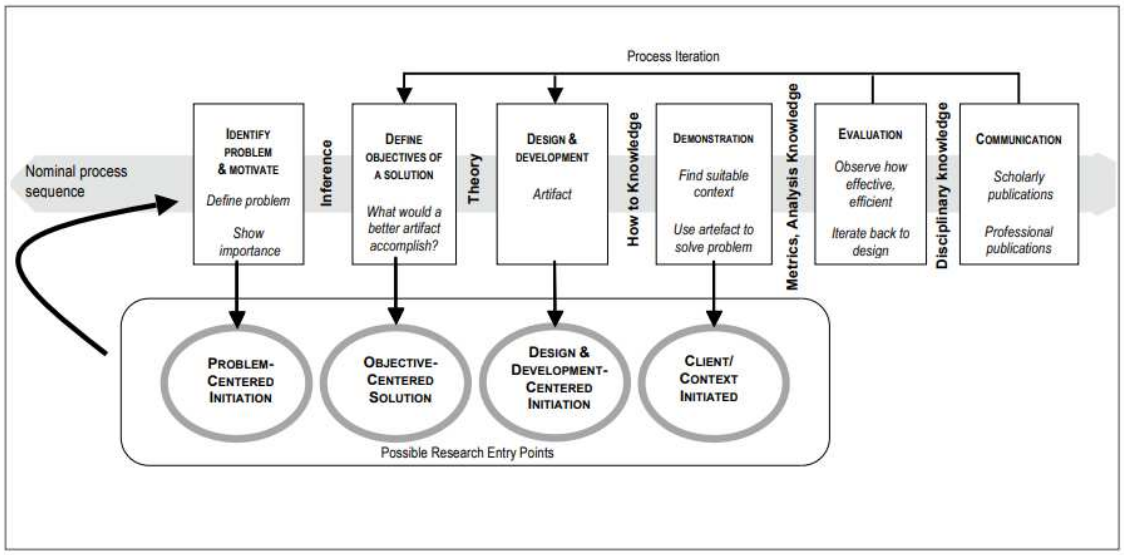


Figure 1.1. Design Science Research Methodology (DSRM) process model [4, Fig. 1]

Making the link between this methodology and the specific case, approaches to counter the lack of DSS Data Requirements in operational systems will be studied, by following the defined steps, with the purpose of understanding the positive impact of the DSS requirements on the design of the operational systems and also by providing a way to measure the impact of each requirement.

In order to achieve those, the following artifact will be developed: (1) The *Instantiation* of a data mart to highlight the importance of the defined arguments in what concerns the requirement assessment; (2) A *Method* to evaluate the impact of a missing data requirement in the decision process.

A demonstration and evaluation will be made after this development in order to later finish this research with its communication.

The following figure (Figure 1.2) demonstrates the adaptation of the DSRM to the present study.

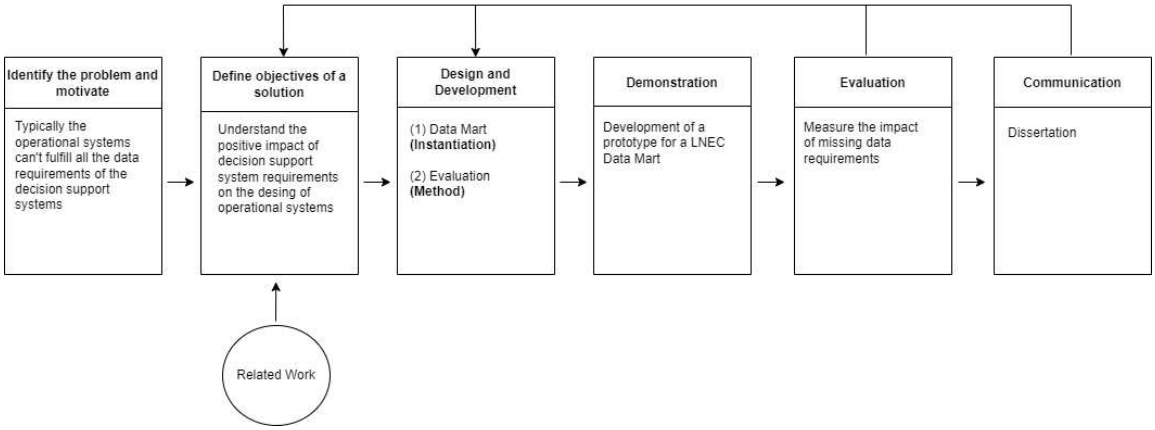


Figure 1.2. Adaptation of the DSRM process model for this dissertation

The following topics will be an in-depth description of each stage of the Methodology (Figure 1.2).

1.4. Design and Development

When developing a DSS, there are two possible stages of the operational systems that are relevant and need to be looked at in different ways: the already-developed operational systems and the ones yet to be developed. Logically, if changes are to be made in an already developed operational system to cope with the requirements of the BI system, these changes will have an associated cost. On the other hand, when assessing a yet-to-be-developed operational system,

these changes have a significantly lower cost as they are only changes to the list of requirements of the system.

For both these stages, the objective is to realize how to integrate the DSS requirements into the operational systems and how to measure and understand which requirements make sense to be integrated in the case of already developed operational systems. To achieve this, the artefacts used for this dissertation are the following:

(1) Development of a Data Mart (*Instantiation*): A DSS Prototype will be developed with gathered missing requirements from another already developed DSS.

(2) *Evaluation Method*: Proposition of a method to measure the impact of a missing data requirement in the decision process.

1.5. Demonstration

A DSS prototype will be developed for a data mart in the real context of a Portuguese Public Institution. This prototype will be developed with missing requirements from the already existing data mart creating an ideal prototype. This will be done by creating the ideal dimensional models and showing how to integrate analytical requirements in the design of the operational system, displaying the gap due to the integration of these requirements on the operational system.

1.6. Evaluation

The impact of the proposed approach on the analytical capabilities of the DSS will be evaluated by comparing the queries, hierarchies, dimensions, and facts that can be obtained by the developed approach against the results that are obtained by a conventional development where no DSS requirement was considered on the design of the operational system.

2. Literature Review

The current chapter will review the related work needed to bring context to this dissertation as well as to better understand the problematic and what scientific research work has already been done that can be related to the subject we are assessing.

Therefore, we are going to be focussing on Software Engineering once it covers the full software lifecycle, from its conception to its creation and maintenance. Since this dissertation covers dependencies between operational and analytical systems that can affect any part of the software lifecycle, this chapter is organized as follows:

2.1. Software Engineering

According to Parnas [5], Software Engineering is a “multi-person construction of multiversion software” [5, ch.1, pp.1]. In other words, software engineering is the development of a system by various software engineers in which each of the engineers develops components that can be modified by others and combined to deliver the system [6].

These Systems play a huge role in society, and the modern world would not properly work without them. From industry to health to entertainment, most of the world has daily contact with devices and, consequently, software [7].

Making the bridge from the general software systems to the information systems is crucial since the definition of the second is close to the first, especially in recent years. Software Systems are defined as a mix of people, software, hardware, and various other resources that will store information and make it available to the organization [8]. Although there is a tendency to associate information systems with computers and software in general, these have existed since the dawn of civilization, being an example of this the card catalogues in a library that are used to store data about books [8].

The referred definitions explain that software and specifically information systems are, in fact, imperative for business success in today's global business dynamic [8] Agreeing with Sommerville's view, Pressman [9] states that in the '50s, no one would have guessed that "software would become an indispensable technology for business, science, and engineering" [8, Ch.1, pp.2]. Therefore, when it comes to the evolution of the role of software, the author [9] considers that there is a dual role to it: being a product and a vehicle.

This statement is backed by the argument that as a product, software delivers computing potential to produce, manage, acquire, modify, display, and transmit *information*. On the other hand, it is the vehicle for delivering the product as operating systems, networks, software tools, and environments are all based on software themselves [9].

2.1.1. The Attributes of Software

As there are very different kinds of software systems for the most various purposes, there are also different methodologies and techniques to deal with them. Nevertheless, there are still reports of software failures, even with all these different methods.

According to Sommerville [7], these failures can be either due to the increasing system complexity or the failure of the use of software engineering methods. This author also raises a few crucial questions about software and its attributes and development. One of which is very relevant to bring context to the present work, which is: "What are the attributes of good Software?" [6, Ch.1, pp.20].

The properties of quality in use are categorized into eight characteristics: Functional Suitability, Performance efficiency, Compatibility, Usability, Security, Maintainability, and Portability, according to the ISO 25010:2011 norm [10], as shown in the figure below (Figure 2.1).

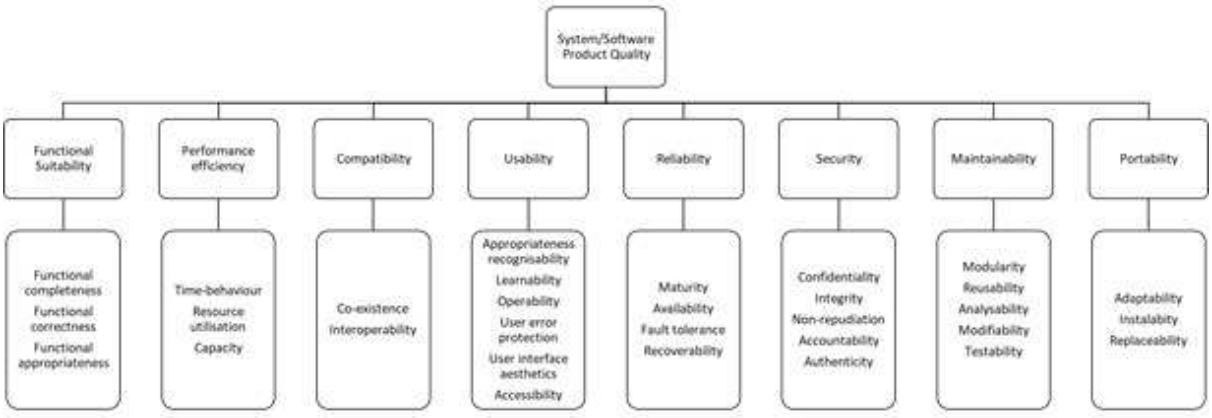


Figure 2.1. ISO 25010:2011 Product Quality Model [10]

Pressman [9] also believes that Software components should be developed in a way that can be *reused,* as he states that a library of reusable components is the way to go when developing software. The example of a today's user interface and how they all use components such as pull-down menus and others is used to back this argument as a library of reusable components will make the process of developing this software easier. *Reliability*, *Robustness*, *Verifiability* and *Timeliness* amongst others are also some qualities of Software that are considered key for its success [6].

With all these quality rules to success, it becomes challenging to keep up with the attributes and to develop software systems that can fulfil all of them. Therefore, the need to adapt is imperative.

2.1.2. The Challenges of the Software Lifecycle

As it becomes easier to operate on a global scale as a business, the number of opportunities, markets, and competing products and services has grown as well. For this reason and because most business operations depend, on some level, on software, the production of software needs to be quick to match the fast-moving markets and the pressure made by the competition. Speed in the development of software systems becomes one of the most critical factors for businesses, as some will easily trade off speed over quality.

With this changing environment surrounding business operations, the task of collecting user requirements becomes harder. On many occasions, it is only after the software is developed and the user has gained experience that the needed requirements can be defined. For this reason, an old-school approach to developing these systems may not be ideal as the change is constant, and there would be a lot of rework [7].

To better understand this process, it is necessary to recognize the four fundamental activities in which the software production is decomposed: Software Specification; Software Development;

Software Validation; and Software Evolution. *Software specification*, the first step of the software production process, is going to be on focus throughout this dissertation, as one of the main objectives is to understand how a different approach to the specification of operational systems can positively impact the insight retrieved by a DSS**.** A deeper reflection on this topic is going to be assessed further in this review.

On the other hand, when it comes to the activities, it is also important to bring attention to the Four Universal Challenges for Software [7] which are: Increasing Heterogeneity; Business and Social Changes; Security and trust; Scaling.

Taking the above statements into account, it is necessary to adapt the development process to overcome these challenges.

2.1.3. Software Development Methods

As stated before, old-school approaches are not capable of dealing with the fast-paced changes of the business, and for that reason, new methodologies need to be adopted to better cope with the new reality of software engineering [7], [11] and with the speed of the constantly changing requirements. This need for fast pace software development has been pointed out long ago [7], [12].

After this topic, the software development methodologies that originated from the need to adapt to the software challenges must be addressed, reflecting on three different key methodologies: Waterfall, Prototyping, and Agile. The primary purpose of this reflection is to suggest a methodology that better adjusts to the objectives of this dissertation.

2.1.3.1. Waterfall

The waterfall model is a sequential development model composed of six different stages: Analysis, Design, Development, Testing, Implementation, and Maintenance. Before each stage, the requirements are checked, and there cannot be overlapping stages. At the end of each stage, documentation and tests are made. Once they are accepted by the customer, that stage is frozen, and the next one begins.

This model tolerates no changes to the initial requirements as the next stage is implemented, taking only into account the previous stages [13].

2.1.3.2. Prototyping

When using more conventional approaches to software development, end users came across the issue of being only capable of coming up with some of the requirements after the development process.

This methodology helps counter this effect as, as its' name indicates, prototypes or minor versions of the product are developed throughout the development process and given to the end user to test. Feedback from these tests and new requirements are welcome during the development stage [14].

2.1.3.3. Agile

Agile is one of the methodologies developed to deal with a rapidly changing business environment where requirements change faster than software would be traditionally developed. Nevertheless, a plan-driven approach, where a complete system analysis is made before development, is best when it comes to a "safety-critical control system" [7].

Therefore, Agile is an incremental development method where small components of the product are added to the main system every few weeks, composing a new release that is made available to the customer. The priority is to satisfy the customer with continuous delivery, and the term continuous delivery is critical as requirements are welcome to change in any stage of development.

It is also important to note that the communication between the developers and the businesspeople must be daily, and between the development team members, a casual face-to-face conversation is the preferred communication method. The development team must also regularly reflect on how they can improve from the previous development iterations [15].

Scrum and Extreme Programming (XP) are two prevalent approaches to the agile method, which are, to this day, being used by development teams as standard development methodologies [16].

2.1.3.4. Comparison of Methods

As shown by the definition of the mentioned software development methodologies, all of them have pros and cons and different utilities in different situations. As the Waterfall model may be ideal for projects where the goals are well defined and easy to implement as the stages are very well defined themselves, it is not flexible [17], and change amid development is not tolerated.

To counter the cons of the Waterfall model, the Prototype model is best used when the goal is not well understood, and it does tolerate change, more so it indulges it, as a prototype is given to the customer to test and understand what needs to be a difference in the next delivery. However, it is more time-consuming and requires more money [17].

Nevertheless, besides taking into account the ups and downs of the methodologies themselves, it is essential also to understand the current environment surrounding the businesses and its impact on software development. Moreover, the agile model, as referred earlier, does work best in this environment as it has a tolerance for change as well as promotes fast software development [7].

2.1.4. Software Specification

Taking into account the capabilities to tolerate change from the reviewed methodologies, the software specification process is still a key piece in the software lifecycle. To better understand this process and its impact, it is necessary to take a step back and understand what a requirement is.

"A "requirement" is a necessary attribute in a system, a statement that identifies a capability, characteristic, or quality factor of a system for it to have value and utility to a user" [14, pp.9].

To better understand requirements in a software development environment, it is important to note that the software requirements can be categorized as either *Functional or Non-Functional* requirements. *Functional requirements* are statements of service that must be provided by the system. These should also provide information about the system's behaviour and reactions to inputs, as well as it can state what the system should not do [7].

On the other hand, *Non-functional requirements* are constraints to the system's services as well as to the development and time. Non-functional requirements mostly apply to the whole system [7].

These requirements must be gathered through a gathering process, typically with meetings between the development team and the customer stakeholders [7].

2.1.4.1. The Gathering Process

Inevitably the requirements a customer wants will be modified over time, and especially when using more conventional software development methodologies, it is very important, when gathering requirements, that help is provided to the customer so that they can better communicate their needs. Those needs must be in line with the cost and the schedule of the development process [19].

In the steps to better retrieve the requirements, it is necessary to understand the organization's business requirements. With this information, the vision and scope of the project can be defined. The scope must be well-defined and agreed upon by both users and developers.

In the requirement elicitation process, creative and detailed thoughts are generated, affecting the scope and, therefore, the system by defining how the problem will be solved. These requirements shall be gathered from customers and users.

As the retrieved requirements are gathered and documented, customers and developers are to meet and jointly evaluate the requirements and verify their actual needs [18]. Furthermore, requirement elicitation is a critical part of the software development process, as quality is extremely dependent on the quality of the requirements [20].

2.2. Decision Support Systems

2.2.1. Introduction to Decision Support Systems

As it became essential to deal with the systems explained throughout this literature review (operational/source systems), strategies to retrieve the best possible knowledge from the systems in use by businesses also became essential, and the need for DSS began. Therefore, Data warehousing is defined as a group of decision-support technologies that will enable the decision-maker to perform better [21].

Nevertheless, to explain what a DSS is, there is a need to take a step back and understand a few other concepts.

The following Figure (Figure 2.2) illustrates the Data Warehouse/ Business Intelligence architecture and each of the key components for its' development.

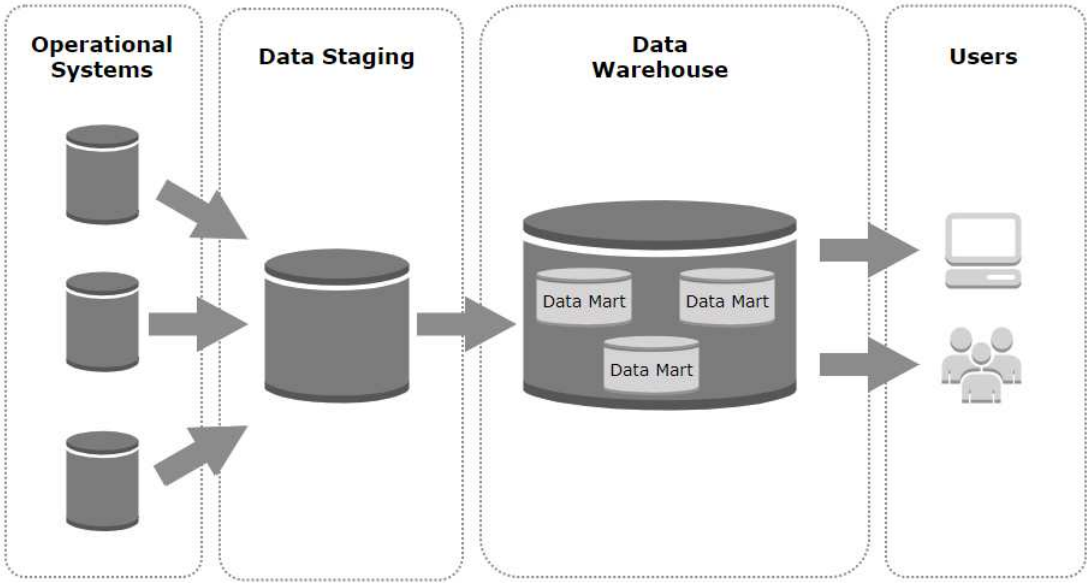


Figure 2.2. DW/BI system architecture.

As shown by the image, the data is retrieved from the Operational Systems into the Data Staging, where the data is extracted, transformed, and loaded into the Data Warehouse [22]. The Data Warehouse is combined of various Data Marts, all corresponding to a single Business Area. This warehouse is queryable and is the source for dashboards and reports that the end user will have access to.

The first concept that is key to the understanding of the DW/BI System is the *Source System* and how it differs from the first. According to Kimball and colleagues [22], the operational/source systems' role is to capture the transaction of the business, and their main attributes are uptime and availability.

Operational Systems are, therefore, systems that track business events and transactions [23] as well as maintain minimum historical data.

It is also important to note that none of the keys used in the operational systems are used as keys in the data warehouse, as these keys are treated the same way as any other attribute [22].

As said before, *Data Staging* is the area where extract, transform and load (ETL) processes are executed [22]**.** Kimball defines the Data Staging as the storage area where the data is transformed, cleaned, and overall prepared to be loaded into the Data warehouse.

The *Business Processes* are several business activities retrieved from the business users. For each Business Process, one or more data marts must be developed. Being a *Data Mart* "a logical subset" of the data warehouse, meaning that it is a subset of the data warehouse that assesses a single business process or various business processes from the same business group [22].

Finally, and as defined before, the *Data Warehouse* **"**is a queryable source of data in the enterprise (…) and nothing more than the union of all the constituent data marts"[18, Ch. 1, pp. 1.4].

Still, about DSS, it is important to note that these are also software systems that must follow the software quality norms as any other software system should.

2.2.2. Decision Support Systems Lifecycle

As the Operational Systems must be optimized to deal with the everyday business transactions once they were designed based on the business rules and developed to efficiently register business activity, on the other hand, BI Systems have a whole different intent. These systems must efficiently answer queries about the different Business Metrics and context around them. Therefore, the objectives of each system are different, and consequently, the design will also be different. Although different, the BI system is highly dependable on the Operational Systems, which reverts to issues that have already been pointed out throughout this dissertation. These issues are already being countered with Agile BI strategies where a more symbiotic relationship between both systems is being used as they are being developed in parallel. [2]Taking into account the previous statements, as the development of a DSS is still a software development process, it also has a lifecycle defined to better organize the tasks in need for this development process, as shown in Figure 2.3.

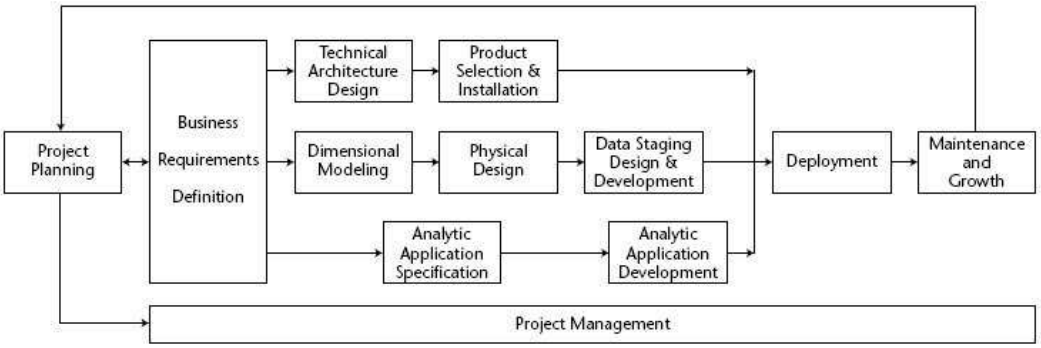


Figure 2.3. The Business Dimensional Lifecycle diagram [22]

Much like the other Lifecycles, this one begins with the Project Planning activity. This activity's intent is to define the cope, the staffing and the task assigning and sequencing of the project [22]. These tasks are dependent on the business requirement definition, being that the reason for the double-sided arrow in Figure 2.3.

The Business Requirement Definition, the second step of the lifecycle, is very important for the process as the better the development team understand the business requirements and the way those will impact the users, the higher the chances of the success of the DSS [22].

The business requirements will also lead to the data needed to fulfil the analytic requirements [22]. As stated previously, an operational system is designed for its specific role in the business, and the data modelling on this kind of system is going to differ from the modelling used for a DSS. In the second case, the dimensional modelling is the way to go, according to the main authors on this subject [22], [24]. The process behind the development of the dimensional model and the retrieval of requirements for that same model will be assessed further in the review.

As well as there are a few more Specification activities, such as the Technical Architecture Design and The End User Application Specification; logically, there will be matching development activities for each of the specifications throughout the lifecycle.

After the development process, the final product must be deployed to be used by the end-user [22]. The project then needs to be maintained, and support must be given to the end users when in need. The success of the previous iteration of development may also make room for growth [22].

2.2.3 BEAM and Dimensional Modelling

As referred before, there is a need to answer queries about the business in an efficient way, and the dimensional model accomplishes that.

Dimensional modelling is a model built upon two main concepts: Dimensions and Facts. Facts are measurements/business metrics, and dimensions are the description used to bring context to the facts by filtering, grouping and aggregating the measurements [24].

The BEAM approach [24], a business process-oriented method, makes it so that each business event is matched to a table; therefore, the challenge of the requirement gathering will be to identify the attributes [25].

When modelling a star schema, the 7W's method was developed. This method simplifies the modelling process by narrowing it down to answering the following questions: When, how, who, how many, where, why and what. The answer to the "how many" question will lead to the measures that shall be included in the fact table. By answering the rest of the questions, we will obtain the attributes for each dimension table which will bring context to the facts [24].

This approach is later described as using "the notion of data stories that are told by stakeholders to capture data about business events that comprise business processes"[22, pp.3]. The BEAM Canvas is a visual "Board" for this methodology, as shown in the following figure (Figure 2.4).

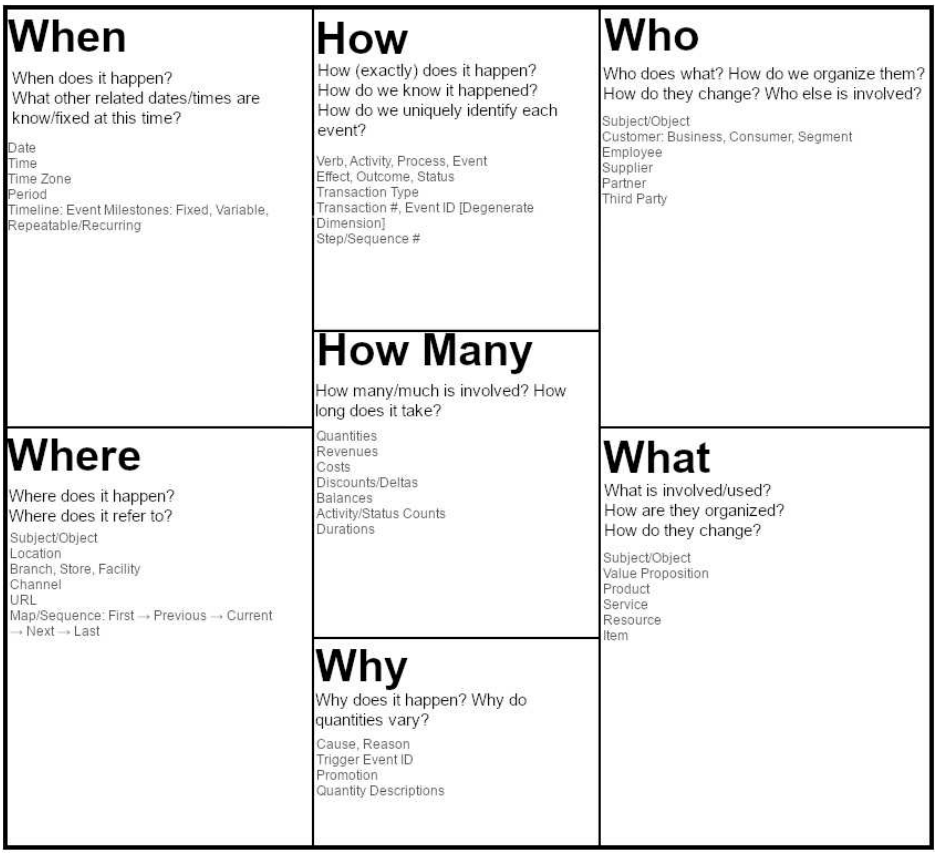


Figure 2.4. BEAM Model Canvas

The business metrics and the context brought by the dimensions must be captured from the source systems. Therefore, these are only attainable if the data gathered by the source systems is sufficient. Consequently, some relevant metrics to analyze, business-wise, may be impossible to calculate with the data gathered, which may affect the decision-making process and the business.

Being *limited access to necessary information* reportedly one of the issues that has a strong relation to poor business management [27].

Although it is possible to change an Operational System to start gathering the data needed to calculate the required business metrics, changes to software generally represent a big cost [7].

2.2.4. Extract-Transform-Load (ETL) and Data Quality

Following the DSS introduction, there's a need to dig deeper into a very important process that was lightly mentioned above - The Extract-transform-Load (ETL) process.

Kimball [28] defines ETL as the base for the data warehouse. Also, pointing out that this is an activity that is not very visible to the users, although it typically takes up most of the development time [28].

As referred before, the activities for this process are the following: (1) Extract, (2) Transform and (3) Load.

Starting with the (1) Extract, this first phase of the process is very dependable on the source system, which is one of the main focuses of this dissertation. To begin this process, it is necessary to develop the logical data mapping or *source-to-target mapping*, which is usually documented by identifying the source of the data, the target data warehouse data model, and the transformation rules that need to be implemented [28].

There are two phases for the extraction process, the initial extraction and the changed data extraction [29].

The initial extraction [22] is the stage where the data is loaded from the source systems into the data warehouse for the first time.

This is when all the historical data is loaded up until the current loading period. After this first extraction, the Data warehouse will only be incrementally updated with the changes made to the already loaded data or the data added since the last extraction. This stage is referred to as the changed data extraction or changed data capture [29].

Following the loading stage is the (2) Transformation stage, where the data is to be cleaned and conformed, and the data warehouse is essentially developed from the fact tables and its grain definition to the dimensions [29].

When cleaning and conforming the data, the data quality must be assured, and therefore the following are to be used as the main attributes for the data to be considered accurate: Correct, Unambiguous, Consistent and Complete [28],[29].

This process may be the most challenging in the data warehouse development process as it is not always the case that these attributes can easily be fulfilled.

Once the data is extracted and transformed, the final stage of the ETL process, (3) Load, can begin. This process is defined by writing the data into both dimension and fact tables [29].

2.2.4.1 Data Quality Challenges

The data quality is, in fact, very important for the success and usability of a DSS [30] as it is also generally an essential component for the costumer's perception of a product's quality [31].

As stated before, the ETL process, specifically the cleaning tasks of the transformation process, is also used to improve the quality of data that is going to be loaded into the Data Marts. Therefore, we are going to define the data quality problems that are categorized in the literature and reflect on the ways that each issue must be dealt with.

In order to access the data quality problems, these must be split between schema and instance-level data problems [30].

The schema-level problems are those that can be dealt with by applying constraints or adjusting the schema in a way that the data problems will be no more [30].

Examples of this type of data problem are missing data, wrong data types or values, duplicates, and wrong categorical data, among others. [30] Some of which can be dealt with using methods such as defining a not-null constraint to solve the missing data problem or a unique constraint for the duplicate values.

On the other hand, there are the instance level problems [30] which are the problems that cannot be dealt with, with schema level constraints, nor can they be detected from a schema level point of view. as perhaps a dummy value replacing a null on a not null constrained field represents a missing data problem at an instance level.

Among the issue above, there are many others [30], such as misfielded values, duplicate records due to typing differences for example and many others.

These problems must be assessed with the right kind of techniques of cleaning and transformation, and in some cases, there may be the need to resort to data enrichment techniques to reach the best possible data quality to develop a DSS.

2.2.5. Agile Methodology Uses in the Development of a Data Warehouse

It has been pointed out throughout this review that the development of a Business Intelligence (BI) system is still a kind of software development. With that in mind, it is understandable that the best methodologies still apply to this process.

It is stated by Corr [24] that the Agile Method is the way to go, with its reputation for generating business value through incremental development and delivery. The implementation of the Agile Method, along with techniques like Scrum, will generate a potential improvement in the development of Business Intelligence applications.

Nevertheless, in order to have an actual impact, agile should be adapted to meet the needs of the development of a DW/BI application while taking into account its principles [24], and its application should also address the design of the Data Warehouse itself.

As referred previously, the Agile Method is also based on a continuous delivery where changes to the software specification can be made throughout the development process [15], and this stage is the most impactful on this dissertation's objective.

2.3. Cost of Change

With the basis of good software production and the need for change in some cases in mind, it's important to dig deeper into the software Process to better understand the impact of change on software. The ever-changing software systems quickly become "Complex, difficult to understand and *expensive to change*"[5, Ch. 1, pp.18]**.** Sommerville [7] states that change is a constant in any large software project. There are many factors that can pressure the change in the software systems, such as business needs, competition, and others. It is also valuable to note that the later the software development phase is, the bigger the cost of rectifying errors [20].

For this reason, there is a need for software systems (operational systems) to be able to accommodate change. Sommerville [7] also points out that change raises the cost of software development, which means that approaches to reduce the cost of rework had to be developed.

One of the proposed approaches is Change Anticipation which means that the possible changes are studied, and sometimes prototypes are made before the production of the software to avoid rework costs. The other proposed approach is Change Tolerance, where the design and, therefore, the production of the software is made, taking into account the fact that the system needs to be easily changed according to new requirements. Besides these approaches, the author [7] also refers to two ways of coping with change which are *Prototyping* and *Incremental delivery* (assessed in the earlier chapter of this literature review).

2.4. Prioritization

As the cost plays a big role in the decisions made by companies in what concerns software, there is a need to understand how to efficiently prioritize the software features.

For software systems in general, the opinions about the difficulty of requirement prioritization differs between the various authors and ranges from easy to extremely difficult [32].

The most important characteristics of a requirement when defining a priority between requirements are Importance, Penalty, Cost, Time and Risk [32].

Specifically, regarding Key performance indicators, the most referenced criteria to evaluate and prioritize is SMART - Specific, Measurable, Attainable, Realistic and Time-sensitive [33].

Most of the related work about prioritization of software requirements refers to the development of a software system which usually will be the typical operational system.

In these cases, the intervening parties are the stakeholders and the development team, and when implementing the techniques, the developers can give a rough estimate of the cost of an added requirement to the system [34].

When it comes to a DSS, the development team is an added layer to this already complex topic that is defining requirement priorities.

Therefore, all three parties will need to dialogue and weigh their perspectives to achieve the prioritization of the requirements.

3. Design and Development

The following chapter will address the path to developing the artifacts defined earlier in this dissertation.

3.1 Evaluation Method

To evaluate the impact of each missing data requirement, it is crucial to take into consideration that both the context and measures around a business are always changing (as stated in the related work chapter); therefore, the most impactful requirement today can very well be irrelevant tomorrow. Nevertheless, there is a need for a method that can help the decision makers decide which requirements may make more sense to fulfil, being the impact on the business and the effort to retrieve the requirement, the metrics for this evaluation method. As the first step of this process, it makes sense to break these missing data requirements into three major categories: (1) Missing Context (Dimensions), (2) Missing measures (Facts), (3) Missing Values (could affect both Dimensions and Facts)

1- Missing Context:

This topic will be divided into three subcategories: (1.1) The missing W's, (1.2) Hierarchies and (1.3) Attributes

1.1. The missing W's:

As stated in the literature review, the various dimensions retrieved on the specification phase of the dimensional model all answer to a specific W from the BEAM methodology.

When it comes to the different W's, it is very probable that some of them are more impactful than others, business-wise. Using an example to illustrate this point: we can consider a supermarket as the business in which DSS is going to be implemented. Let's also consider the number of sales as the metric for the desired business queries. If the intended query is the amount sales by brand of products, some conclusions from that query could be taken, but not many as it does not take into consideration a time period. Whereas if we have the number of sales by year, this query clearly shows very relevant information about the state of the business and its evolution.

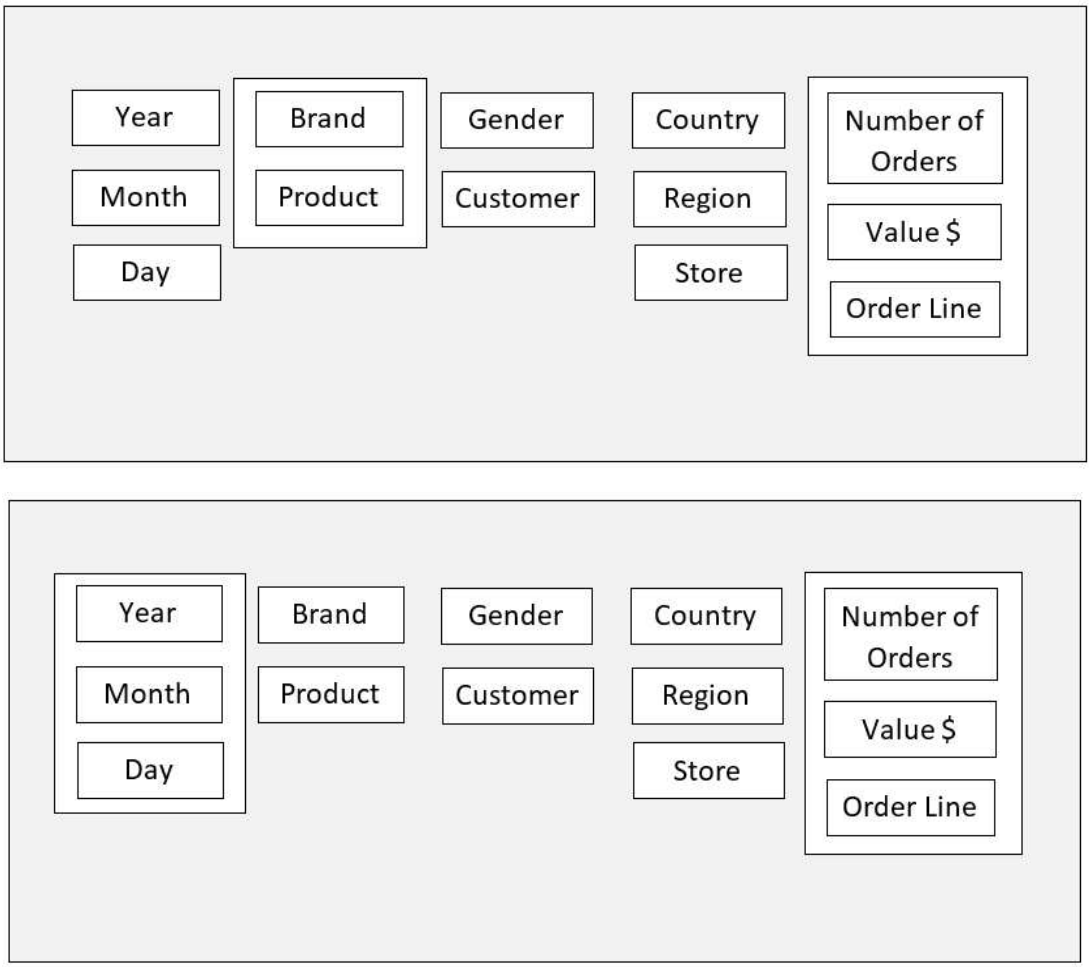


Figure 3.1. Example Data Mart with Highlighted When, What's and How Many's

With this example, we intended to show that a When is typically more impactful than a What; therefore, if the effort needed to retrieve each missing data attribute was the same, it would be reasonable to state that retrieving the temporal attribute would be a better choice.

1.2 Attributes:

When addressing missing requirements in what concerns dimension attributes, adding any attribute to the model will always enrichen it. However, the impact of these attributes on the system is very subjective and highly dependable on the decision makers needs and opinions. Although, if the attribute in question is the one that describes the elementary grain of the model (e.g., transaction identifier, order line) as is in figure 3.4, it then becomes impossible to guarantee the integrity of the derived models and analysis.

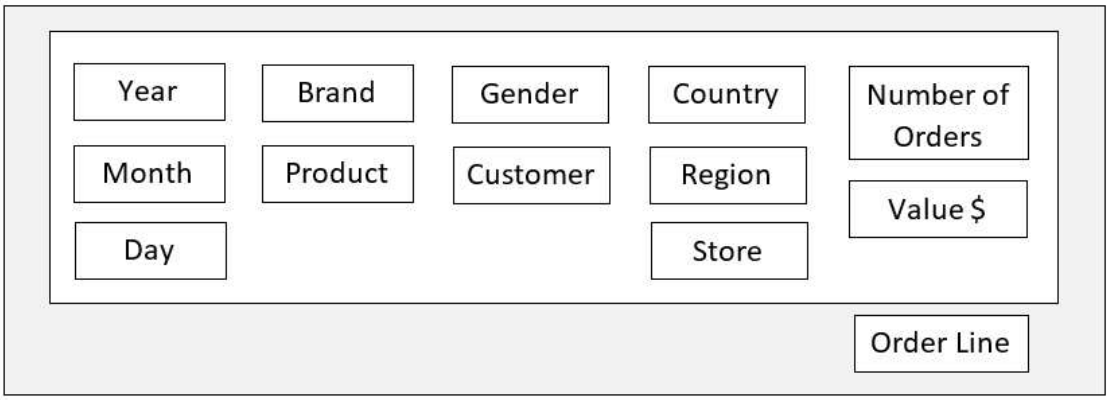


Figure 3.2. Example Data Mart with Highlighted When, What's and How Many's

1.3 Hierarchies:

When arguing about dimension attributes and addressing the possibility of adding an attribute to increase the detail levels of a dimension, it will also increase the analytic capacity of the system, whereas when adding an attribute on the same hierarchical level as another will only allow to display the information that was already available, in another way. Considering the same hypothetical example as previously, the capacity to add the detail level of the product being sold in the supermarket rather than just its brand will bring a lot more value than adding an attribute like the customer number, as the name of the customer would suffice for queries on the customer detail level.

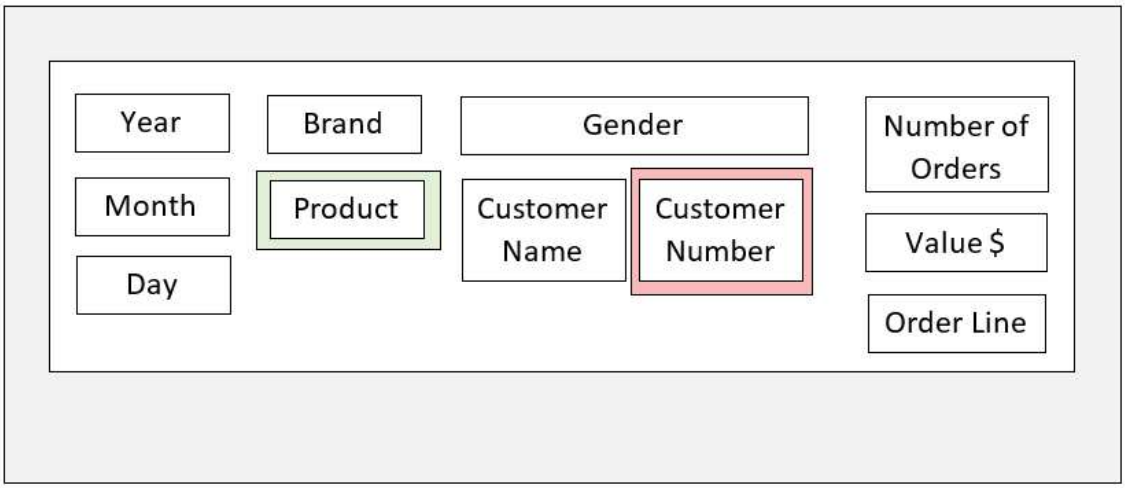


Figure 3.3. Example Data Mart attribute relevance

2-Missing Measures :

In what concerns the fact table, it is once again necessary to take into consideration that the impact of each requirement is very subjective and different for each specific case. Logically, it could seem that a new measure on a fact table would always be more insightful than adding an attribute, but this is not always the case.

Therefore, it makes once again sense to evaluate the importance of measures and attributes, taking into account the 7 W's mentioned before, adding that the missing How Many's and How's are some of the most impactful.

Another important factor is whether a business event is or not present on the fact tables. It is our understanding that missing a metric for a business event that already has some metrics represented on the fact table is different from a missing metric for a business event that is totally absent from the fact table, as not having a business event taken into consideration by the DSS could represent a blind side and therefore a negative impact for the decision-making process.

3- Data Quality Issues

Once again, referencing what has already been stated in the related work section, historical data is extremely important for a DSS. The whole reasoning behind these systems' existence is to take advantage of the information about the past to improve the present and the future of the business. Nevertheless, operational systems do not always have historical data management capacities, and therefore it is not unusual to come across this kind of issue when developing a DSS. When assessing this kind of issue, the level of missingness of the data is the metric that should be used to understand and decide whether this data can become available to the DSS.

3.1- Data Quality Issues – Dimensions

Just like in the missing attributes, also in the missing values, the weight of what is missing may differ a lot; for this reason, the same logic will be applied. Therefore, typically, missing values on an attribute which is on the same detail level as other attributes without missing values is less problematic than missing values on one or multiple attributes, which makes it, so there is close to none or no information available on a specific detail level. Also, the theory about the importance of some of the w's over others can be applied.

3.2- Data Quality Issues - Facts

In what concerns the missing Values on a fact table, the thought process is identic to the dimension. The factors that are more concerning on the missing attributes are also the most concerning on the Missing values side of things. Therefore, missing values of a metric that describes a business event by itself is potentially more negative than metrics that do not.

Method:

With all the subjectivity of the categories above taken into consideration, it was desirable to come up with a simple, intuitive, and fast way of evaluating the impact of the missing data requirements on the DSS.

At first glance, it would seem that counting the number of missing data requirements by category (W's, attributes, hierarchies, etc.) would suffice and therefore be able to compare the amount of each missing data requirement on both approaches of the DSS systems and come out with conclusion on this. Nevertheless, business requirements are just not that simple and straightforward. It may very well be the case that there is a specific business query that is of extreme importance to the decision-making process and that there are a few other queries which will be barely looked at and taken into consideration. Therefore, the weight of this first requirement would be much higher. With this, the approached can only be compared in the following way: Taking DSS 1 and DSS 2 as hypothetical Decision Support Systems, we can only state that the DSS 1 is more beneficial than DSS 2 if the DSS 2 consists of a subset of the requirements fulfilled by the DSS 1.

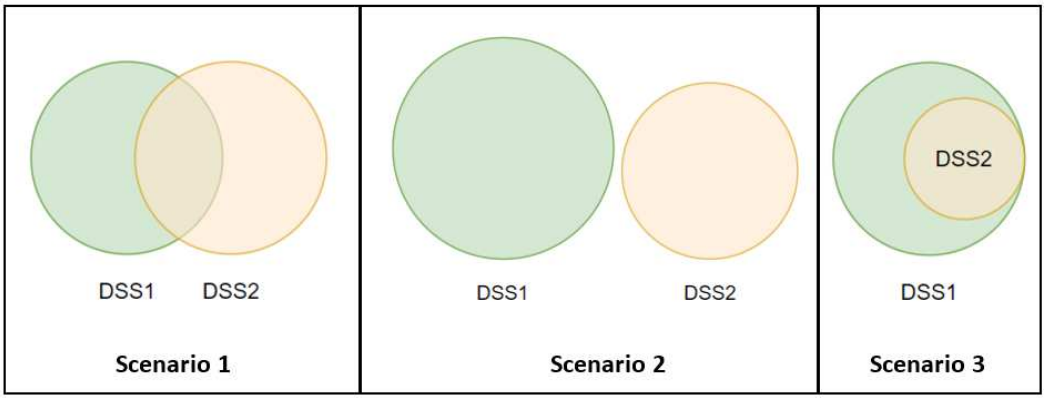


Figure 3.4. Scenarios for comparison of Decision Support Systems

Looking into the three scenarios in figure 3.4, it is not possible to state which one is more beneficial in scenario one, and only a deep assessment of the requirements in both approaches with stakeholders could potentially tell which one of them would be more appropriate for the current business needs. Also, in scenario number two, even though DSS1 may have more requirements fullfiled than DSS2, again, it is not possible to state which approach would be more beneficial as DSS1 could have more irrelevant business queries answered than DSS2. Therefore, only in scenario three is it possible to clearly reach the conclusion that the DSS1 approach will be beneficial when compared with DSS2.

Nevertheless, as stated previously, the stakeholders for the DSS may very well be capable of evaluating the queries answered by each of the systems in scenarios one and two and coming to the conclusion of which covers the most relevant business queries and therefore is more beneficial.

Back up for the Evaluation method:

To test and justify the chosen evaluation method, the query capabilities of two data marts will be compared, and the method will be applied.

A factor that will also be impactful is the starting point of the source system. If the case is that the source system has not been developed yet, the development of this system should take into consideration the DSS requirements. Whereas if the operational system is already built and functional, it will be necessary to consider the cost-benefit of these changes as well as understand if it is possible to change the system to meet the needed requirements

Method to fulfil data requirements:

For all the explained categories, it is now important to take into consideration the following factors: *the level of lack of data* and the *system's capacity to collect the data (or lack of it)*.

In what concerns the existence of data, there are different levels to this issue. If the data exists physically but is not in the operational systems, or if it is in the operational system but only partially, there are ways to deal with it, such as inputting the data manually into the source system or using files as the source of data for the DSS. On the other hand, if the data simply does not exist or if the effort put into making the data available to the DSS is very significant, it becomes much harder to fulfil the data requirements. From this surfaces the belief that the impact of these changes can be better understood if the software development lifecycle is taken into consideration and it is analyzed which stages of the software development are affected.

As the weight of each missing data requirement was very much subjective, also the cost associated with changes to the operational system is very hard to standardize and quantify as each system is unique and what is a table with many dependencies in one system may have close to none in another. There could even be the case that the ERP cannot be changed.

Therefore, although being able to suggest ways to better understand the impact of requirements on the business, it's not possible to associate a cost to the fulfilment of a data requirement in a simple way.

In order to reach this cost/benefit value, the decision makers would be needed to evaluate the impact of each requirement, and also, the developers and the users of the operational system would have to estimate the feasibility of adding these requirements to the source system.

3.2 Case Study

To better describe the design and development section of this thesis, first, the case study on which we are testing and developing our artifacts, will be presented. This Case Study is based on a Business Intelligence project developed for a public institution containing a few data marts

focused on the Financial, HR and Projects processes. The focus of the case study will rely on the Projects data mart. Therefore, to better explain the design of the data mart that was made in this project, first, the steps and key information about the Projects business process will be explained.

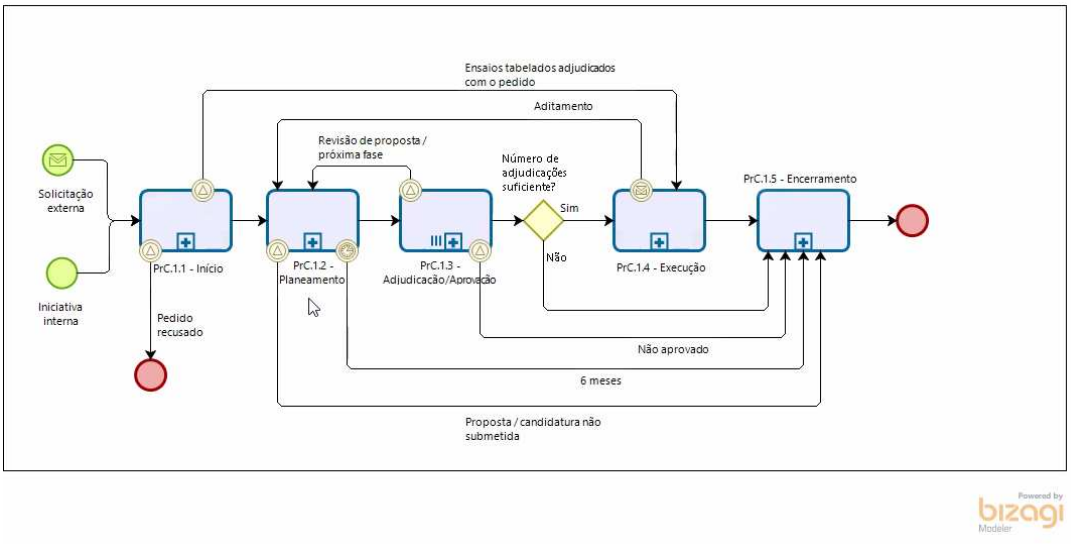


Figure 3.5. Project activities diagram

As is shown in the figure above, every project has within it a few phases marked by dates.

PrC.1.1 – Start of the project

PrC.1.2 – Planning

PrC.1.3 – Approval

PrC.1.4 – Execution

PrC.1.5 – Closing

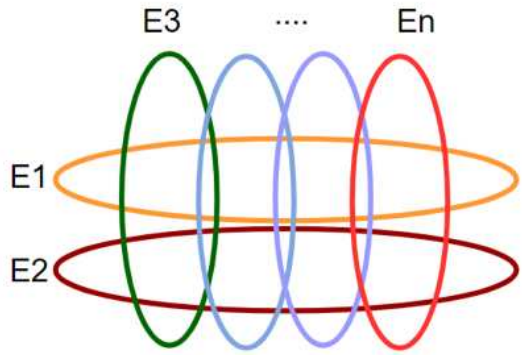
These projects can be either an intern initiative or external contract. In the latter, the institute would be providing a service to an external entity, as the name suggests.

Logically, not every project goes through the whole stages and there maybe various reasons for a project not to be accepted in the early stages or to not be approved after planning, etc.

Also, as some of the markings on the figure suggest, there may be various reasons for the project to go back to an earlier phase of the process, such as a push back on the delivery date, which will require a replanning.

All the data about these various phases of the project is manually inserted into a specific platform by the project managers. This data is written in a SQL database that is the data source for the DSS.

Related to the projects, this public institution deals with complex challenges through innovative and integrated responses. Therefore, every project is classified according to a set of axis and themes in order to track the project's alignment with its strategy.



				THEMES			
				T1	T2	...	Tn
				New technologies	Information Technologies		
AXIS	TRANSVERSAL AXIS	E1	RISK AND SAFETY				
			Risk evaluation				
			Risk reduction measures and technologies				
		E2	INNOVATION TOOLS				
			Experiment tools				
			Forecast and analysis tools				
		...					
	STRUCTURAL AXIS	...					
		En	NATURAL RESOURCES				
			Integrated management of resources				
			Resource usage				

Figure 3.6. Axis and Themes Diagram

The axis are directly connected to the public institution's areas of action. These main axis are then broken down into two to three lines about their program.

Transversal Axis refer to: E4 – risk and safety; E5 – innovation tools

The Structuring Axis refer to projects that aim at dealing with society's necessities: En – natural resources.

On the other hand, the priority themes are a set of policies that are relevant to the institution, such as T1-New Technologies and T2 – Information Technologies.

These are just a few examples of a broader range of axis and themes.

3.2.1 In Production DSS

As the projects are a key component of the business of the use case institution, it was decided that this business process would be addressed, and metrics and dimensions would be defined to bring the stakeholders insights on this side of the business.

As a result of that, the following model was a proposed and developed Project's data mart. There were also HR and Financial components to this Data warehouse, although the focus was exclusively on the facts and dimensions regarding the Projects.

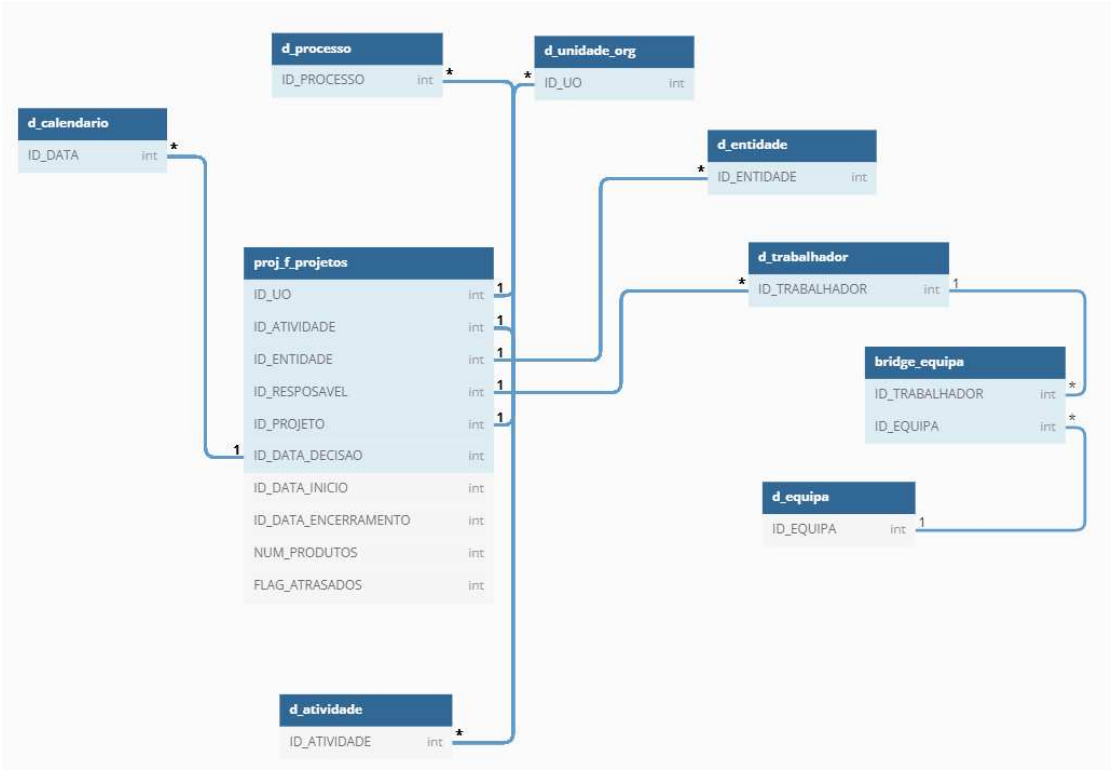


Figure 3.7. Projects Data Mart

Figure 3.8. Axis and Themes Data Mart

As seen in the figures above, two fact tables were modelled and developed to meet the requirements for the Projects.

The projects fact table will contain the following metrics:

- Number of Projects
- Number of Delayed Projects

And it's grain will be: A line per project

And the Dimensions are the following:

- Activity – Area of Activity
- Process – Project in this specific context
- Entity – Entity that finances the project
- Employee – Project responsible
- Team – Project's team
- Calendar – Calendar dimension

The axis and themes fact table will contain the following metrics:

- Axis percentage
- Theme percentage

And it's grain will be: A line per project, per axis, per theme, per date.

And the Dimensions are the following:

- Process – Project in this specific context
- Axis – Project Axis (explained previously)
- Theme – Project Themes (explained previously)
- Calendar – Calendar dimension (project decision date, project start date, project end date)

Taking this model as a basis and also the project documentation, the goal is to evaluate the requirement-gathering process and make a full assessment of the missing requirements. This assessment will consist in checking whether the best practices of BI modelling have been used as well as spotting inconsistencies with the model. Nevertheless, without some business context, it is close to impossible to retrieve all the missing requirements, as one would be biased by the documentation developed.

While looking into the inconsistencies, it is relevant to also understand the limitations that come with a project of this nature developed in a professional setting. Due to budget, time and scope limitations, it is not always possible to use the best practices of the development process for a DSS, and it is very usual for development teams to use strategies that will make for a quicker process, but on the other hand, will produce a poorer overall model. An example of this is the requirement gathering process which, according to literature, should be agnostic of any already developed analytics system that a company might have, but developers will often look into those to have a faster sense of what the client may be looking for. This type of approach may also

disregard relevant business queries if the data that would answer that query is not available within the source systems.

On the instantiation of the Ideal Data Mart Prototype, a very practical way to gather a categorize the missing requirements based on literature and also professional personal experiences will be proposed.

3.3 Instantiation of Data Mart

In this chapter we will be instantiating our second artifact, in this case a data mart based on our case study.

3.3.1 Ideal Data Mart Prototype

To instantiate the Ideal Data Mart Prototype, first, it's crucial to gather the flaws of the baseline data mart so that it's possible to understand what the necessary changes are - this process will be called the Missing Requirements Gathering.

3.3.2 Missing Requirements Gathering

When the DSS requirement gathering process is made with a scientific basis, in which the requirements are retrieved following the BI model canvas, only taking into consideration the business needs, none of the participants is biased by the limitations of the operational system. In this ideal scenario, once the source-to-target mapping is developed, it is trivial to realize the requirements/ business needs that were pointed out by the stakeholders and cannot be fulfilled. This lack of requirement fulfilment can be motivated by a few limitations that were pointed out previously, and the impact can range from big to small according to the type of requirement that can't be met.

Nevertheless, according to personal and professional experience in the Business Intelligence field, most companies will not comply with these requirement-gathering frameworks in an effort to save time. They will, therefore, look into the operational systems and develop the analytic models taking into consideration only the business needs that can be retrieved from the source systems and completely discard the unmet business needs. Apart from these cases, there may also be budget limitations that will make it so that the team that is developing the DSS doesn't even look at specific requirements that may have been pointed out by the stakeholders.

In the latter cases, a new requirement gathering must be done to understand what requirements haven't been met, which kinds of requirements these are, and what impact they have in the analytic model.

Just like in many other cases, our Case Study project also had unfulfilled requirements. In this specific case, the source-to-target mapping wasn't developed, taking these requirements into consideration; therefore, they had to be retrieved via meeting; as this is only a case study, these requirement-gathering meetings were only made with one of the project's stakeholders.

On the other hand, it is also necessary to do a best practice assessment to check if some of the key rules of modelling a BI system have been followed.

As a result of these meetings and assessments, it was possible to achieve the following Missing Requirements:

Table 3.1. Missing Requirements table

Issues		Type of issue							Solution				
ID	Description	Missing Context				Missing Measures	Data Quality		Operational System Changes	Business Process Change	Data Enrichment	BI Model Changes	Manual Data Input
		W's	Attributes	Hierarchies	Hierarchy Levels	Missing Measures	DQ issues on Facts	DQ issues on Dimensions					
I1	Lack of data for cost associated with the earlier stages of the projects					X			X	X		X	
I2	Cost of projects that were not approved					X			X		X	X	
I3	Axis and thematic (without the need for pairs)		X						X			X	
I4	Missing Dates for the earlier stages of the projects (info isn't inserted by the users)	X						X				X	X
I5	Semester and Quarter missing on date hierarchy				X						X	X	
I6	Missing Date of Birth on Employee Dimension		X									X	
I7	Missing Gender on Employee Dimension		X		X							X	
I8	Department missing on Employee Dimension		X		X							X	
I9	Day of the Week missing on Date Dimensions		X								X	X	
I10	Holidays missing on Date Dimensions		X								X	X	
I11	Axis Hierarchy missing on Axis Dimension		X	X							X	X	

In order to classify these issues, they will be separated into three different groups. Note that there may be an overlap between the groups:

BI model defects, which are issues that are specifically due to errors in the BI modelling process and can be fixed only by changing the layers of the DSS. And all the data available to solve these issues can be retrieved either by ETL rules, for example, the missing holidays or date hierarchies, adding additional columns to already built dimensions and merging dimensions that shouldn't be separated in the first place.

In the current practical case, these are the issues numbers: I5, I6, I7, I8, I9, I10, I11

The second group of issues that we came across is the *Process flaw.* These issues are mainly relevant data inputs that aren't being done by the employees, as perhaps the project dates are only being input once the project is accepted and is going to start, and many are left empty, resulting in no data or insights about the earlier stages of the projects. Some of these issues can be solved on the historical data side with data enrichment techniques, although it is mostly necessary to enforce that the manual inputs are done according to the process by all the employees. A change to the operational system to make some of these fields mandatory could also be a solution to enforce this policy.

The issues that refer to this group are the following: I1, I2, I3, I11

The third and final group is about the *Transactional Model flaws.* This group refers somewhat to the previous one as the process isn't being registered correctly in the source system, although the reasoning behind the groups is different. In this case, the system is not allowing the process to be correctly input. An example of this is the obligement for axis and thematics to be in pairs, which by design is wrong and should not be mandatory. Therefore, in order to correct this, a change to the source system must be done impacting then, the process and the BI system.

The issues that belong to this group are: I1, I2, I4.

For development purposes, there will be a focus on the requirements that weren't met by this project and that are missing on this data mart, and we will be trying to shine light on ways to solve these issues. Data usage has been allowed under the condition that it is all randomized to protect its confidentiality.

Once the missing requirements are retrieved, and the matrix is filled, it is possible to then know the impacts that obtaining each requirement will bring to the business. Therefore, it should be a stakeholder call of whether a missing requirement is worth the work that is needed to obtain it.

In our specific case, we gathered with a project's stakeholder and came up with four relevant missing requirements we are going to obtain. Together we have chosen issues from the three different groups to face different challenges and walkthrough each of them along the development

of this dissertation. Therefore, the chosen requirements we are going to be developing are those with the following IDs: 2, 3, 4, 5 and 11.

3.3.3 Development process

Once the requirements to be developed in order to fix the issues were picked, the actual development process started.

The first step of this process was to assess the logical design of the solution, where the project's data mart design was analysed, and the relevant changes were made, as shown below.

For this step, each requirement was looked into, one by one, coming to the following conclusion:

For requirements I1, I2 and I4, the business process isn't being followed as it should, and therefore there is no data for the earlier stages of the project resulting in no data for the costs of these stages and no costs associated with projects that were not approved as these sometimes aren't even input into the platform.

Following this issue, the calendar dimension relation to the fact table was developed without taking into consideration the two earliest stages of a project, submission and planning. Therefore, the first change to the model was to add both ID_DATA_SUBMISSION (for the submission date) and ID_DATA_PLANEAMENTO (for the planning date). That will work as logical keys from the Calendar dimension table to the Projects Fact table.

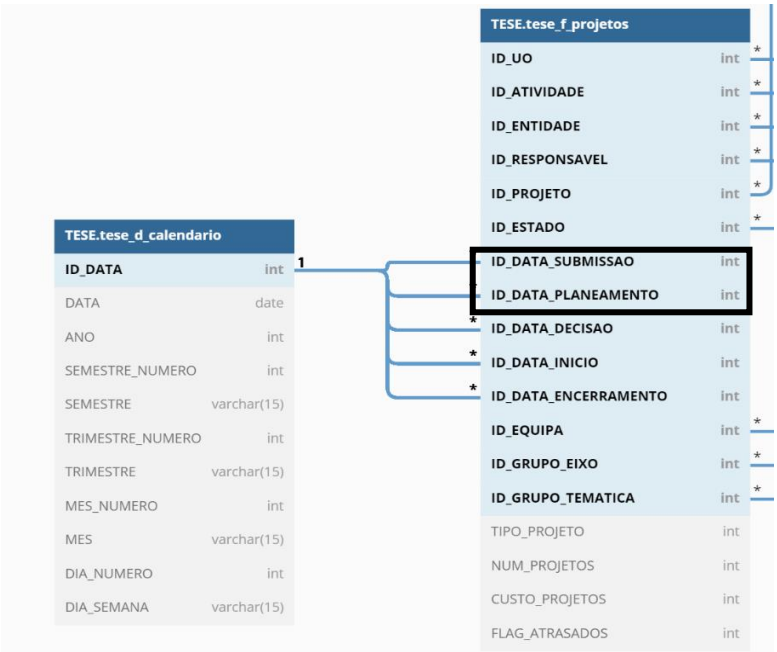


Figure 3.9. Added dates for earlier stages of the project

When it comes to the I3 requirement, it was possible to come to the conclusion that the source system didn't follow the right logic for this process. For that reason, a change to the Operational System would need to be done in which the need to input Axis and Themes in pairs would be removed, and the users would be able to set as many Axis and Themes for a project with whatever percentages. In order to mimic this on the Operational System side, there was a need to produce some mock data to then be able to show the analytics benefits this will add to the system.

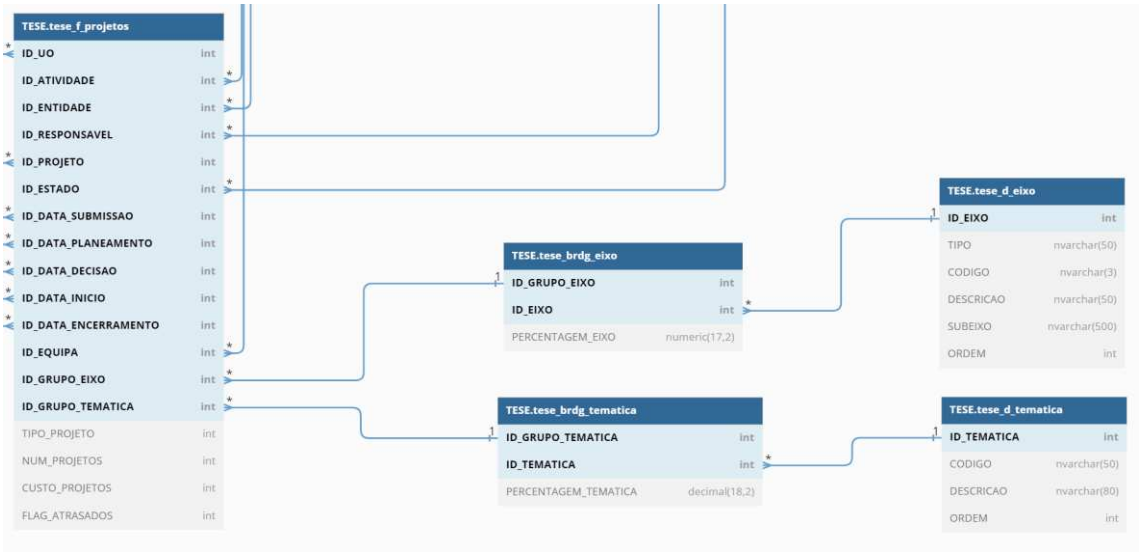


Figure 3.10. Axis and Themes bridge tables and relation to the Projects fact table

Furthermore, looking into these dimensions, another issue that was found was the missing hierarchy on the Axis Table, I11. Therefore, this issue was fixed by adding the missing columns and using that enrichment to populate them, as the data for these were available in the Lab's documentation on Axis and Themes (Figure 3.10)

Figure 3.11. Added axis type and subaxis attributes to the axis dimension design

In what concerns the I5 issue, all the columns and data were already present in the dimension, and it was only not available in the dimension's hierarchy set in Power BI. Therefore, the fields Semester ('Semestre') and Quarter ('Trimestre') were added to the hierarchy as shown below.

Figure 3.12. Added semester and quarter attributes to date hierarchy on Power BI

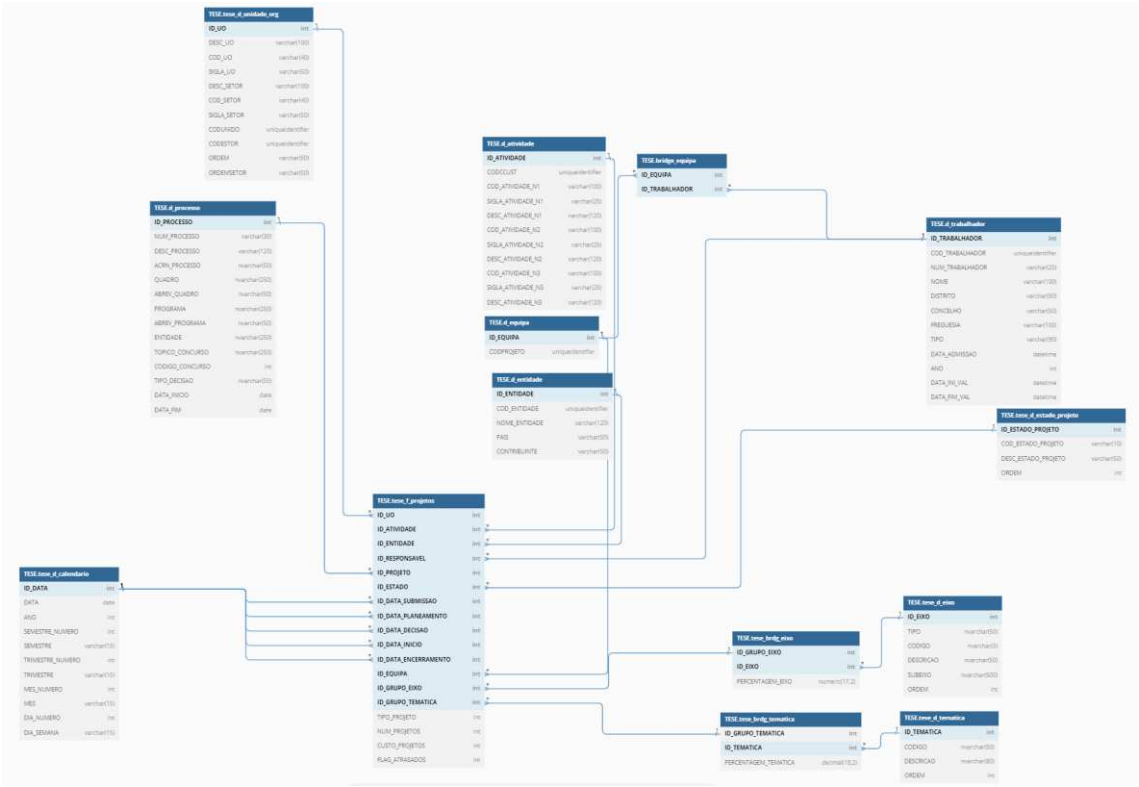


Figure 3.13. Projects ideal data mart prototype

After the model was updated to fulfil the necessary changes, it was also necessary to update the source to target mapping where the new fields and the way they will be loaded will be documented.

For the specific case where the operational system needs to be altered to have the relevant data, there will not be a possible match on the source system to fulfil the data requirement. In this case, it's necessary to simulate the changes to the operational system to accommodate these new requirements and add a mock field to the mapping.

This column will then have mock data in order to be processed and presented to show the benefits of the change that was made.

Upon modelling the changes, the next step will be to alter the tables on the Data warehouse.

For this, a built-in table design tool on MS SQL Server was used, and the missing columns, which by default were 'Na' values added.

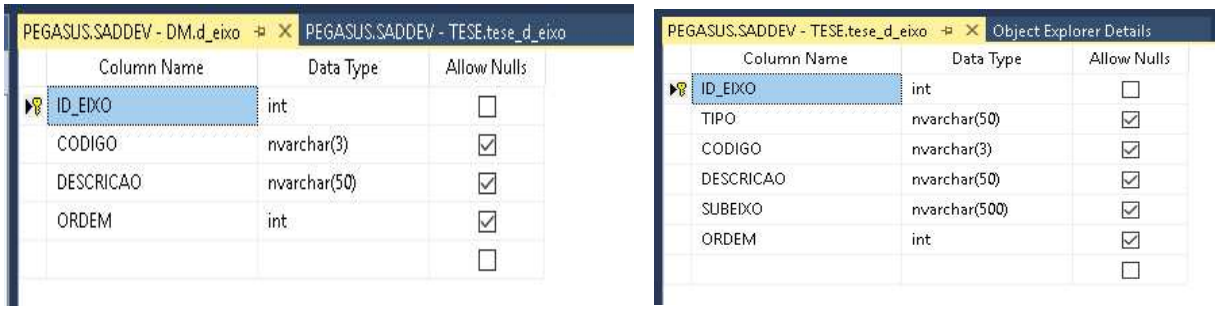


Figure 3.14. Adding the columns to the physical tables on the DBMS

This led to the conclusion of the modelling steps.

Once the modelling was over, both logically and physically, followed the work on the ETL process. The extraction was made with a scheduled procedure that loads the data from the source system into a staging schema with the SQL Server database. As for the remaining of this process, it was developed with SQL Server Integration Services both for orchestration and actual transformation and loading of the data into the data warehouse.

This second part consists of a few steps:

1. *Time frame setup:* The process can either be a full load or an incremental load. For this reason, the pipeline starts with the setup of the time frame that is being loaded.
2. *Transforming (Lookup):* The Lookup phase is a phase in which the data from the staging is read and matched with the dimension data on specific columns. After the match, the corresponding ID from the dimension is added to the row that is going to be inserted in the fact table.
3. *Transforming (Aggregate):* As the name suggests, this is the phase in which metric aggregations are made based on the requirements gathered.
4. *Load:* After steps 2 and 3, the rows must be aggregated and have the right IDs to match the data in the dimensions. These rows will then be inserted into the fact table, concluding the loading process.

Typically, once this process is done, some tests are run to guarantee the logic is being rightfully implemented. These tests are better off being performed by someone that didn't develop the ETL to avoid Bias and systematic errors.

Upon developing and validating the changes to the ETL, the final step is within the presentation layer. In this specific case, this layer consists of a Power BI solution.

In this phase, a couple of steps need to be done:

1. Refresh the sources and make sure they are compliant with the physical model

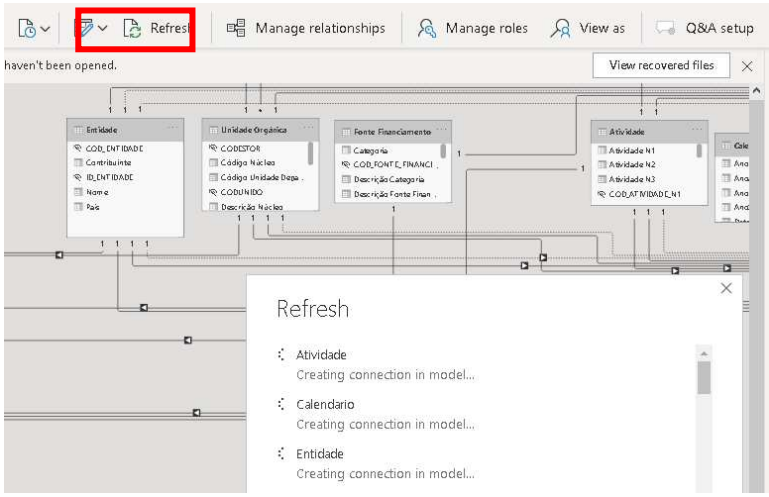


Figure 3.15. Power BI source refresh

2. Alter the hierarchies and metrics according to the requirements set in the requirement-gathering process.

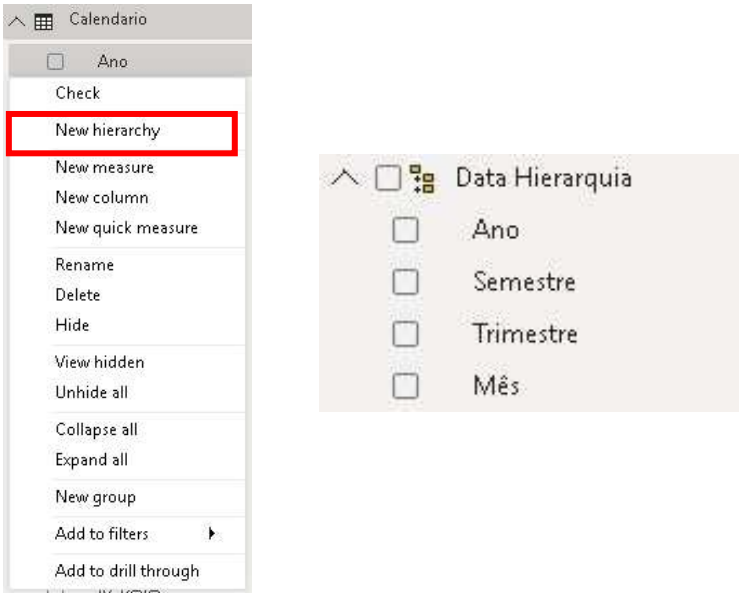


Figure 3.16. Creating hierarchies on Power BI

3. Assess the dashboard mock-up and create/modify the visualizations

4. Demonstration and Evaluation

4.1. Demonstration

In order to demonstrate and evaluate the chosen approach and developments, a comparison of their presentation layer outputs is needed; in this case, Power BI Dashboards and SQL Queries will be used for the following issues: I2, I3, I4, I5 and I11 to demonstrate the approach on each of the BI systems and compare the benefits. In the end, a set of metrics based on our evaluation method will be used to compare the approaches.

I2- Cost of projects that were not approved:

About the I2 requirement, it isn't possible to present a result comparison as in the original Data Mart, the cost of the project isn't a metric, to begin with, and there was no information on whether the projects had been accepted or not. Nevertheless, the Ideal Data Mart Prototype managed to achieve this requirement fulfilment bringing insights into the cost associated with the projects that were not approved.

Ideal Data Mart Prototype Result

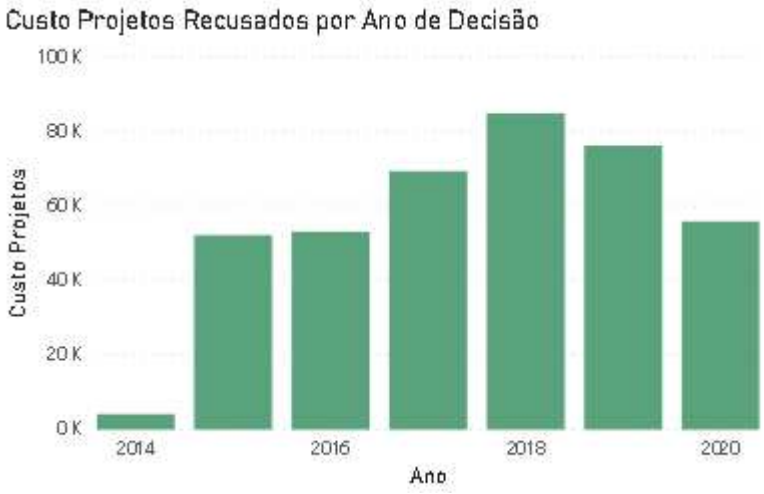


Figure 4.1. Cost of Projects by Decision Year

I3- Axis and thematic (without the need for pairs):

Since Power BI doesn't accommodate bridge tables and won't understand the relationships between such tables, the only way to show the different outputs for the same business query in both data marts is by using a SQL query. There are other visualization tools which would be able to deal with this type of table.

Baseline Data Mart Result

```
select
t.DESCRICAO as TEMATICA,
f.PERCENTAGEM_TEMATICA,
e.DESCRICAO as EIXO,
f.PERCENTAGEM_TEMATICA,
p.DESC_PROCESSO
from
DM.proj_f_eixos_tematicas f
inner join DM.d_eixo e
on f.ID_EIXO=e.ID_EIXO
inner join DM.d_tematicas t
on f.ID_TEMATICA= t.ID_TEMATICAS
inner join DM.d_processo p
on f.ID_PROJETO=p.ID_PROCESSO
where ID_PROJETO in (40,41,1484,1966,2150)
order by ID_PROJETO ASC
```

	TEMATICA	PERCENTAGEM_TEMATICA	EIXO	PERCENTAGEM_TEMATICA	DESC_PROCESSO
1	Nao Definido	NULL	Nao Definido	NULL	BARRAGEM DE CASTELO DE BODE. OBSERVAÇÃO E CONTROL...
2	Nao Definido	NULL	Nao Definido	NULL	COLABORAÇÃO DO NGA NA OBSERVAÇÃO GEODÉSICA DA BAR...
3	Novas tecnologias	100	Património Construído	100	MONITOR - SUSTENTABILIDADE DE ESTRUTURAS DE MADEIR...
4	Novas tecnologias	100	Património Construído	100	MONITOR - SUSTENTABILIDADE DE ESTRUTURAS DE MADEIR...
5	Sustentabilidade e alterações climáticas	100	Património Construído	100	GERIA - ESTUDO GERIÁTICO DOS EFEITOS NA SAÚDE DA QUA...
6	Sustentabilidade e alterações climáticas	100	Património Construído	100	GERIA - ESTUDO GERIÁTICO DOS EFEITOS NA SAÚDE DA QUA...
7	Sustentabilidade e alterações climáticas	100	Recursos Naturais	100	ENQUADRAMENTO INICIATIVO EU-CHINA WATER PLATFORM ...
8	Sustentabilidade e alterações climáticas	100	Recursos Naturais	100	ENQUADRAMENTO INICIATIVO EU-CHINA WATER PLATFORM ...

Figure 4.2. Query result for axis and themes joined with fact table (Baseline Data Mart)

Ideal Data Mart Prototype Result

```
Select
f.ID_PROJETO,
dt.DESCRICAO as TEMATICA,
bt.PERCENTAGEM_TEMATICA as PERCENTAGEM_TEMATICA,
de.DESCRICAO as EIXO,
be.PERCENTAGEM_EIXO as PERCENTAGEM_EIXO,
p.DESC_PROCESSO as PROJETO,
f.CUSTO_PROJETOS as CUSTO

from
[TESE].[F_Projetos] f
inner join TESE.tese_brdg_eixo be  on f.ID
GRUPO_EIXO=be.ID_GRUPO_EIXO
inner join TESE.tese_d_eixo de on de.ID_EIXO=be.ID_EIXO
inner TESE.tese_brdg_tematica bt on f.ID_GRUPO_TEMATICA =
bt.ID_GRUPO_TEMATICA
inner join TESE.tese_brdg_tematica bt on bt.ID_GRUPO_TEMATICA=
f.ID_GRUPO_TEMATICA
inner join TESE.d_processo p on f.ID_PROJETO=p.ID_PROCESSO
where f.ID_PROJETO in (40,41,1484,1966,2150)
```

	ID_PROJETO	TEMATICA	PERCENTAGEM_TEMATICA	EIXO	PERCENTAGEM_EIXO	PROJETO	CUSTO
1	40	Tecnologias da informação	0.50	Risco e Segurança	1.00	BARRAGEM DE CASTELO DE BODE. OBSERVAÇÃO E CONTROL...	25756
2	40	Saúde e bem-estar	0.30	Risco e Segurança	1.00	BARRAGEM DE CASTELO DE BODE. OBSERVAÇÃO E CONTROL...	25756
3	40	Capacitação organizacional e institucional	0.20	Risco e Segurança	1.00	BARRAGEM DE CASTELO DE BODE. OBSERVAÇÃO E CONTROL...	25756
4	41	Tecnologias da informação	0.50	Cidades e Territórios	1.00	COLABORAÇÃO DO NGA NA OBSERVAÇÃO GEODÉSICA DA BAR...	2682
5	41	Saúde e bem-estar	0.30	Cidades e Territórios	1.00	COLABORAÇÃO DO NGA NA OBSERVAÇÃO GEODÉSICA DA BAR...	2682
6	41	Capacitação organizacional e institucional	0.20	Cidades e Territórios	1.00	COLABORAÇÃO DO NGA NA OBSERVAÇÃO GEODÉSICA DA BAR...	2682
7	1484	Desenvolvimento de competências e transferência ...	1.00	Nao Definido	0.25	MONITOR - SUSTENTABILIDADE DE ESTRUTURAS DE MADEIR...	914
8	1484	Desenvolvimento de competências e transferência ...	1.00	Património Construído	0.50	MONITOR - SUSTENTABILIDADE DE ESTRUTURAS DE MADEIR...	914
9	1484	Desenvolvimento de competências e transferência ...	1.00	Risco e Segurança	0.10	MONITOR - SUSTENTABILIDADE DE ESTRUTURAS DE MADEIR...	914
10	1484	Desenvolvimento de competências e transferência ...	1.00	Instrumentos para a Inovação	0.15	MONITOR - SUSTENTABILIDADE DE ESTRUTURAS DE MADEIR...	914
11	1966	Novas tecnologias	0.20	Cidades e Territórios	1.00	GERIA - ESTUDO GERIÁTICO DOS EFEITOS NA SAÚDE DA QUA...	1408
12	1966	Tecnologias da informação	0.20	Cidades e Territórios	1.00	GERIA - ESTUDO GERIÁTICO DOS EFEITOS NA SAÚDE DA QUA...	1408
13	1966	Coesão social e territorial	0.40	Cidades e Territórios	1.00	GERIA - ESTUDO GERIÁTICO DOS EFEITOS NA SAÚDE DA QUA...	1408
14	1966	Saúde e bem-estar	0.10	Cidades e Territórios	1.00	GERIA - ESTUDO GERIÁTICO DOS EFEITOS NA SAÚDE DA QUA...	1408
15	1966	Capacitação organizacional e institucional	0.10	Cidades e Territórios	1.00	GERIA - ESTUDO GERIÁTICO DOS EFEITOS NA SAÚDE DA QUA...	1408
16	2150	Políticas públicas	0.40	Nao Definido	0.25	ENQUADRAMENTO INICIATIVO EU-CHINA WATER PLATFORM ...	24684
17	2150	Indústria para a globalização	0.60	Nao Definido	0.25	ENQUADRAMENTO INICIATIVO EU-CHINA WATER PLATFORM ...	24684
18	2150	Políticas públicas	0.40	Património Construído	0.50	ENQUADRAMENTO INICIATIVO EU-CHINA WATER PLATFORM ...	24684
19	2150	Indústria para a globalização	0.60	Património Construído	0.50	ENQUADRAMENTO INICIATIVO EU-CHINA WATER PLATFORM ...	24684
20	2150	Políticas públicas	0.40	Risco e Segurança	0.10	ENQUADRAMENTO INICIATIVO EU-CHINA WATER PLATFORM ...	24684
21	2150	Indústria para a globalização	0.60	Risco e Segurança	0.10	ENQUADRAMENTO INICIATIVO EU-CHINA WATER PLATFORM ...	24684
22	2150	Políticas públicas	0.40	Instrumentos para a Inovação	0.15	ENQUADRAMENTO INICIATIVO EU-CHINA WATER PLATFORM ...	24684
23	2150	Indústria para a globalização	0.60	Instrumentos para a Inovação	0.15	ENQUADRAMENTO INICIATIVO EU-CHINA WATER PLATFORM ...	24684

Figure 4.3. Query result for axis and themes joined with fact table (Ideal Data Mart Prototype)

While comparing the two outputs, it is clear that the Ideal Data Mart Prototype brings a lot more insight and allows different pairs of axis and themes with different percentages, whereas the production data mart only allowed for a single pair of axis and themes, which does not correctly represent the business logic.

Still looking into the data presented on the Power BI for the original Data Mart, it is possible to see that for a specific activity, there is no data about Axis and themes.

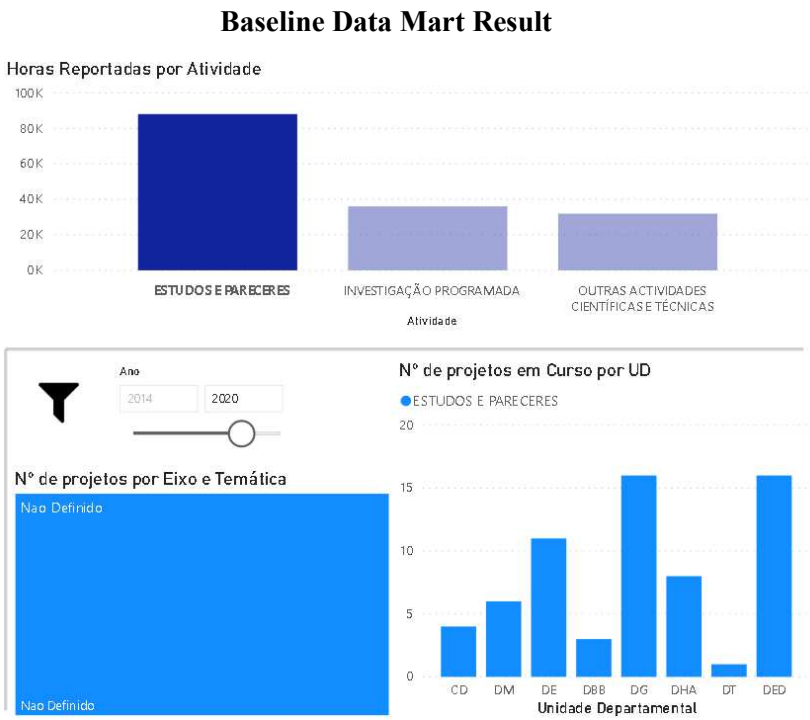


Figure 4.4. Power BI visualizations filtered by a activity (Baseline Data Mart)

When making exactly the same query in SQL format, we can see that for that same activity, there are various projects tied to multiple axis and themes

Ideal Data Mart Prototype Result

```
select
COUNT(NUM_PROJETOS) NUMERO_PROJETOS,
 de.DESCRICAO as EIXO,
 dt.DESCRICAO as TEMATICA
from  [TESE].[F_Projetos] f
left join TESE.d_atividade a on f.ID_ATIVIDADE=a.ID_ATIVIDADE and
a.DESC_ATIVIDADE_N2='ESTUDOS E PARECERES'
inner join TESE.tese_d_grupo_eixo e on
f.ID_GRUPO_EIXO=e.ID_GRUPO_EIXO
inner join TESE.tese_brdg_eixo be on
be.ID_GRUPO_EIXO=e.ID_GRUPO_EIXO
inner join TESE.tese_d_eixo de on de.ID_EIXO=be.ID_EIXO
inner join TESE.tese_d_grupo_tematica t on f.ID_GRUPO_TEMATICA =
t.ID_GRUPO_TEMATICA
inner join TESE.tese_brdg_tematica bt on bt.ID_GRUPO_TEMATICA=
f.ID_GRUPO_TEMATICA
inner join TESE.tese_d_tematica dt on bt.ID_TEMATICA=dt.ID_TEMATICA
inner join TESE.d_processo p on f.ID_PROJETO=p.ID_PROCESSO
group by de.DESCRICAO,dt.DESCRICAO
order by 1 desc
```

	NUMERO_PROJETOS	EIXO	TEMATICA
1	858	Cidades e Territórios	Saúde e bem-estar
2	795	Cidades e Territórios	Tecnologias da informação
3	659	Cidades e Territórios	Novas tecnologias
4	658	Cidades e Territórios	Coesão social e territorial
5	595	Cidades e Territórios	Capacitação organizacional e institucional
6	506	Risco e Segurança	Saúde e bem-estar
7	481	Risco e Segurança	Tecnologias da informação
8	436	Cidades e Territórios	Sustentabilidade e alterações climáticas
9	431	Cidades e Territórios	Indústria para a globalização
10	400	Risco e Segurança	Novas tecnologias
11	399	Risco e Segurança	Coesão social e territorial
12	398	Cidades e Territórios	Políticas públicas
13	398	Instrumentos para a Inovação	Saúde e bem-estar
14	396	Instrumentos para a Inovação	Tecnologias da informação
15	363	Risco e Segurança	Capacitação organizacional e institucional
16	306	Instrumentos para a Inovação	Coesão social e territorial
17	301	Instrumentos para a Inovação	Novas tecnologias
18	286	Instrumentos para a Inovação	Capacitação organizacional e institucional
19	282	Recursos Naturais	Saúde e bem-estar
20	272	Risco e Segurança	Sustentabilidade e alterações climáticas

Figure 4.5. Query result of number of projects with the same activity filter (Ideal Data Mart Prototype)

I4-Missing Dates for the earlier stages of the projects (info isn't inserted by the users):

Addressing the I4 requirement, the dates for the earlier stages of the projects were not being filled within the operational system on the project management platform. Therefore, the only comparable date that is on both Data Marts is the Decision Date.

When comparing the two approaches, the benefits of our developed one are clear, as it was possible to attribute actual project stage dates to all of the projects in the year 1900 (default year for lack of data) on the original Data Mart.

Furthermore, data for two additional dimensions is available, which are the Planning Date and the Submission Date. Both were not considered as dimensions as, once again, these dates were not being input into the project management platform.

Baseline Data Mart Result

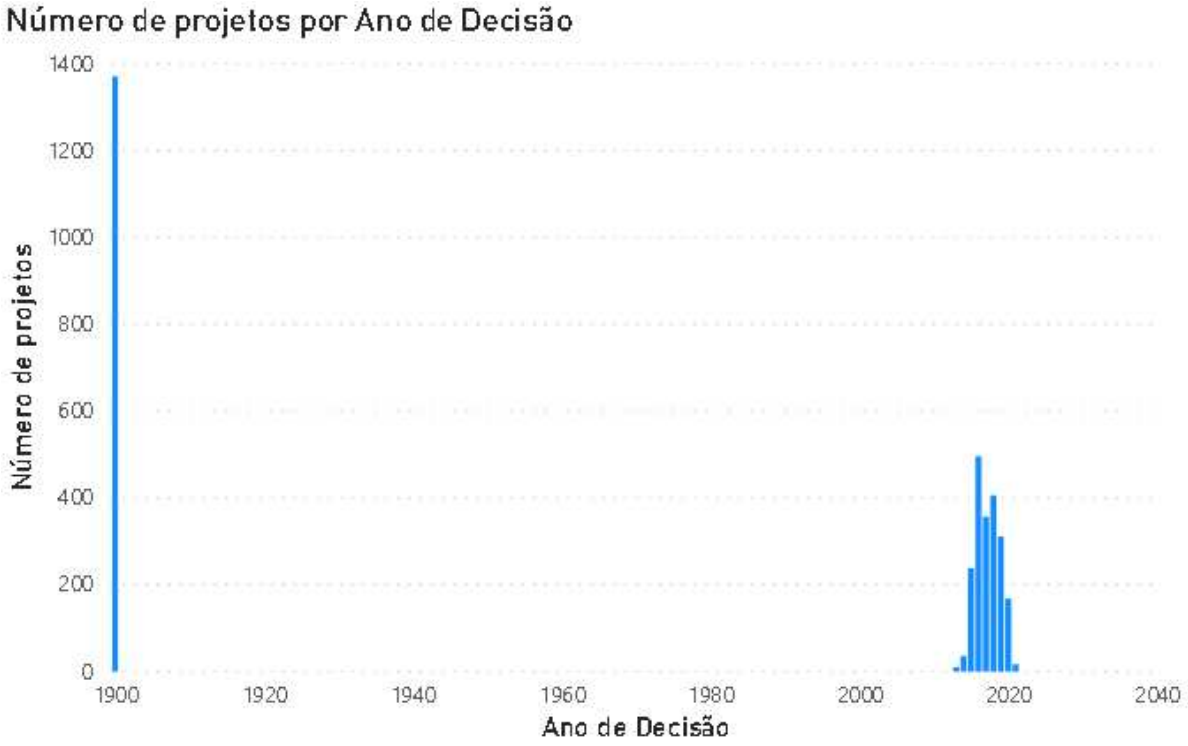


Figure 4.6. Bar chart with the Number of Projects by Decision Year (Baseline Data Mart)

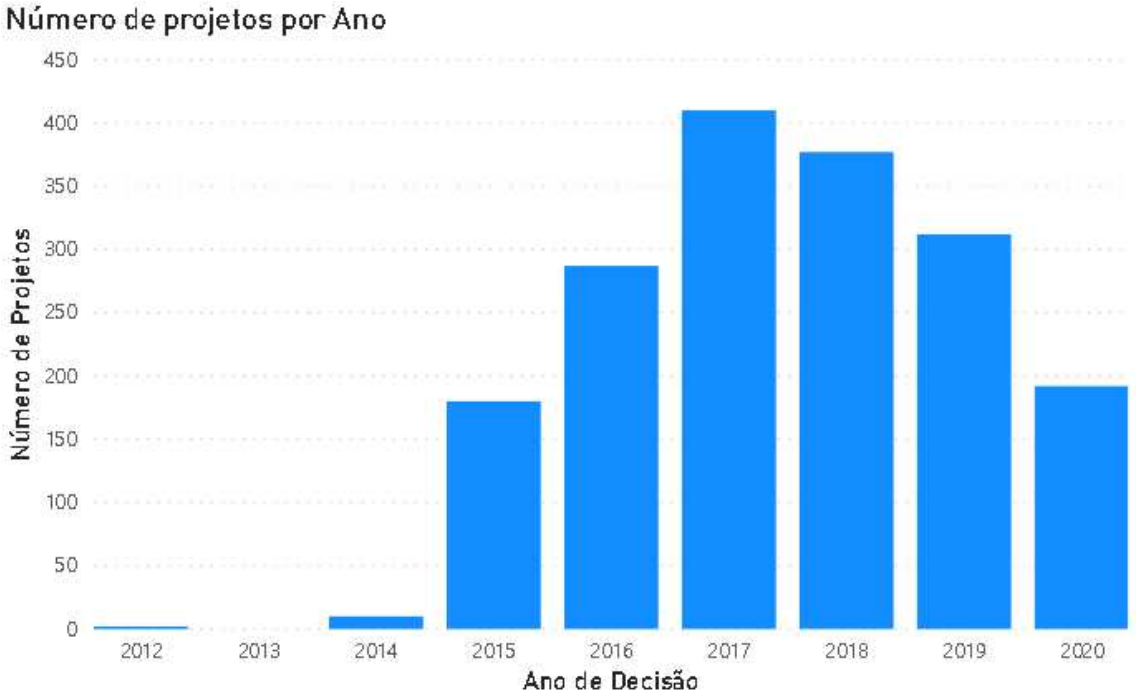


Figure 4.7. Bar chart with the Number of Projects by Decision Year (Ideal Data Mart Prototype)

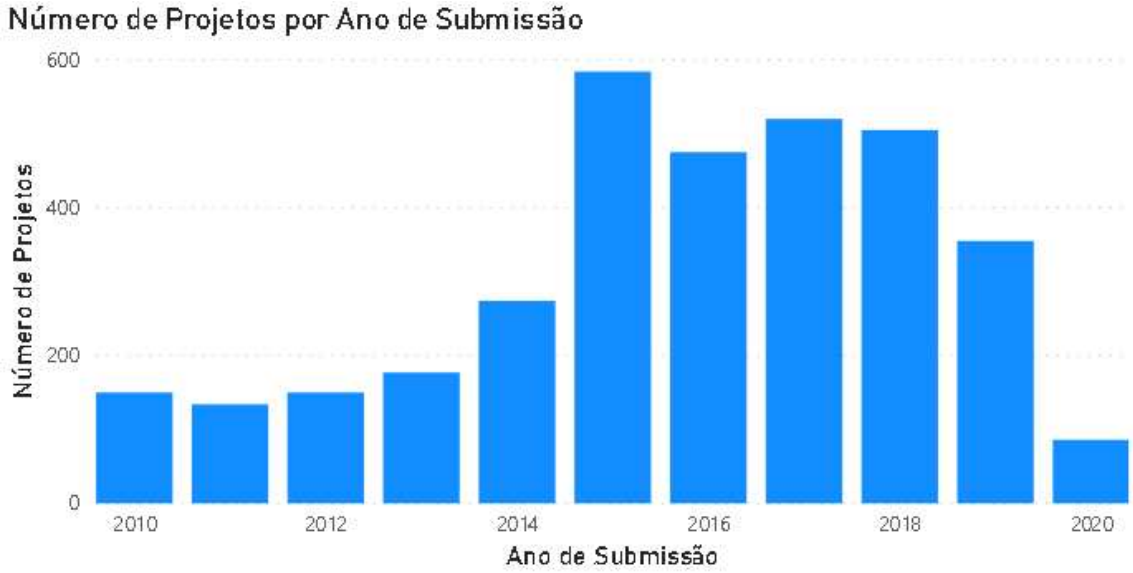


Figure 4.8. Bar chart with the Number of Projects by Submission Year (Ideal Data Mart Prototype)

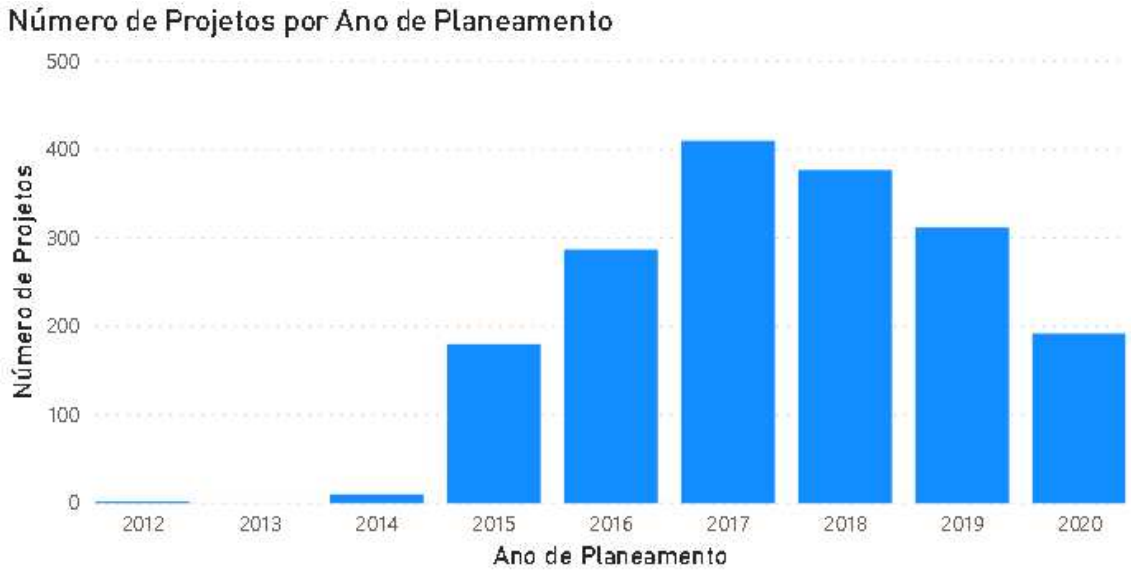


Figure 4.9. Bar chart with the Number of Projects by Planning Year (Ideal Data Mart Prototype)

I5- Semester and Quarter missing on date hierarchy:

On this specific issue, it was possible to add the levels semester and quarter to the date hierarchy, as these were not a part of the hierarchy on the production system. The drill down would lead you automatically from the year into the month. With the Ideal Prototype approach, the system has gained the following visualization options for the business user.

Ideal Data Mart Prototype Result

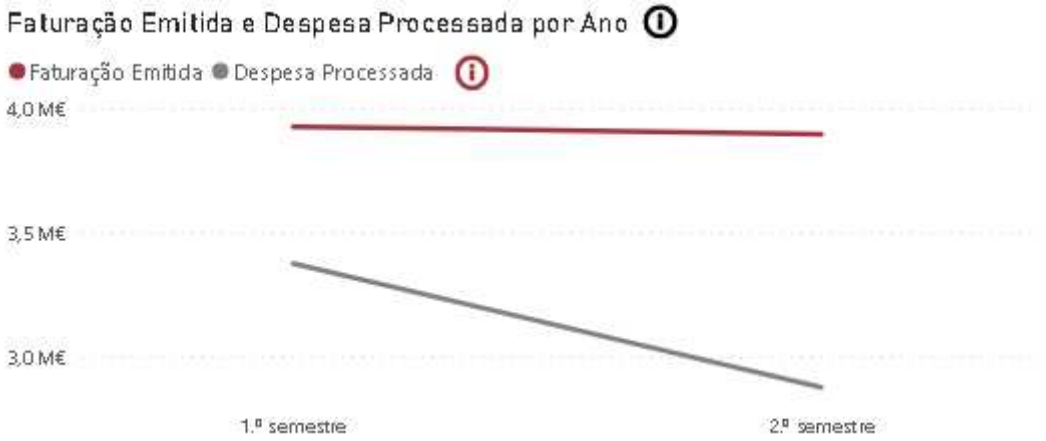


Figure 4.10. Evolution of Revenue by Semester (Ideal Data Mart Prototype)

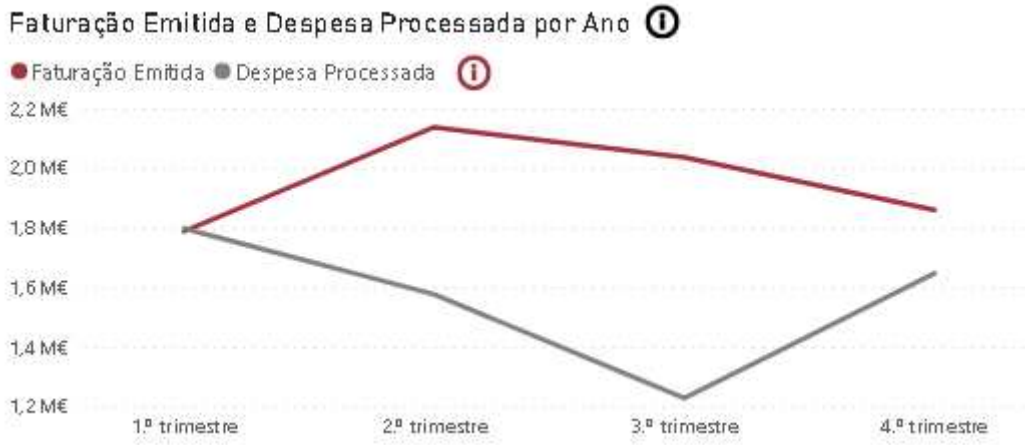


Figure 4.11. Evolution of Revenue by Quarter (Ideal Data Mart Prototype)

I11- Axis Hierarchy missing on Axis Dimension:

On the Axis Dimension, there were missing attributes on a hierarchical level that was not even available on the Baseline Data Mart. Through Data Enrichment and by changing the model significantly concerning this dimension, it became possible to add this hierarchy to the dimension, as proved by the results below.

Baseline Data Mart Result

Results | Messages

	ID_EIXO	CODIGO	DESCRICAO	ORDEM
5	4	E4	Risco e Segurança	4
6	5	E5	Instrumentos para a ...	5

Figure 4.12. Axis Dimension (Baseline Data Mart)

Ideal Data Mart Prototype Result

Results | Messages

	ID_EIXO	TIPO	CODIGO	DESCRICAO	SUBEIXO	ORDEM
11	11	Eixos Estruturantes	E3	Recursos Naturais	Gestão integrada de recursos	3
12	12	Eixos Estruturantes	E3	Recursos Naturais	Utilizações dos recursos	3
13	13	Eixos Transversais	E4	Risco e Segurança	Medidas e tecnologias para redução do risco	4

Figure 4.13. Axis Dimension (Ideal Data Mart Prototype)

4.2. Evaluation

In order to apply the earlier defined evaluation method, it becomes crucial to look at the relevant characteristics of the case study. The operational system had been running in production for a significant amount of time when the DSS development began. Therefore, any new requirements that had a direct impact on the operational system would require applying changes to an in-production system, which by itself is much more complex than making changes to its design in earlier phases before the actual implementation.

Taking this as the starting point for the developments, it was necessary to consider the feasibility of the fulfilment of the data requirements in a system in need of changes to the operational system, methods of data enrichment and business process changes in order to fulfil the gathered requirements.

Understanding the previously stated, both Data Marts can be compared in an unbiased way looking merely at the requirement fulfilment capabilities of each one.

The in-production Data Mart will serve as the baseline in which the requirements fulfilled by this DM are those that were retrieved in a professional environment.

Table 4.1. Requirement comparison table

Data Marts		**Number of Requirements by Category**										
ID	Description	W's							Attributes	Hierarchies	Hierarchy Levels	Measures
		When	How	Who	What	Why	Where	How Many				
1	Baseline Data Mart	4	1	5	1	2	1	2	52	4	16	2
2	Prototype Data Mart	6	1	5	1	2	1	3	56	5	21	3
Gain		2	0	0	0	0	0	1	4	1	5	1

As the table above suggests, the Prototype has significant gains in comparison to the baseline Data Mart allowing for more business queries to be answered. It is also relevant to state that none of the requirements from the baseline Data Mart were lost throughout this development, meaning this, according to our Evaluation Method, we can clearly state that the prototype is more beneficial for the decision-making process.

Of course, this was the expected outcome, although there are a few takeaways from this development and results.

In what concerns DSS design best practices, if these were correctly followed in a professional context, some of these requirements would not be missing, but for many reasons, this is not always the case, and the design of the systems will have flaws that could be answered with what is available. On the other hand, some of the missing requirements would still be within the list as there would not be data to back them up, although these requirements would be categorized as data quality issues rather than missing context or measures, as they will usually be more tied with source system data quality issues that must be solved within the operational context and are not a DSS development team responsibility.

Table 4.2. Data Quality Issues Comparison Table

Data Marts		Number of Data Quality Issues Solved	
ID	Description	Dimensions	Facts
1	Baseline Data Mart	0	0
2	Prototype Data Mart	1	0
Gain		1	0

In what concerns solved Data Quality issues, as table 4.2 suggests we have managed to have gain of one solved issue concerning a dimension. Again, we would have a much greater gain on this side, had the design of the original DSS not been based on the limitations of the Operational System, since most of the flaws that were categorized as missing context or missing measures would be categorized as Data Quality issues.

In any case, DSS design flaws are not the only reason for missing context or measures within table 3.1, and it may also be the case that the business context has changed and the requirements for the DSS are not up to date with this change.

Also, looking into the requirements that required changes to the operational systems, it is clear that adding the DSS requirements to the requirement list of the operational systems in its development stage would make for a perfect integration of both these systems. When the stage the operational system is already in production, it is expected that there is much more reluctance to add new requirements as that will always come with a cost and, in some cases, can be quite complex.

5. Conclusion

As the business world becomes more and more data-hungry and driven, this dissertation shows the importance of early assessment of the data requirements as a strategic advantage for the decision-making process. By prioritizing these requirements, the DSS will be more complete, adjusted to the business needs, insightful for the stakeholder and hopefully will have a positive impact in the business overall by providing a better tool for decision making.

In order to reach this conclusion, a real case study for which a Data Warehouse that was developed in a professional scenario with the limitations that come with a contracted project was used. Out of this Data Warehouse, a Data Mart was chosen to start developing this same Data Mart with the integration of the missing requirements on both the Operational System and the DSS, we called the second Data Mart our Ideal Data Mart Prototype and the first the Baseline Data Mart.

Prior to the development an evaluation method has been defined, in a way that would allow to state that one approach would be an improvement over the other. For this, the first step was to try and make sense of what would be the more relevant requirements and how to quantify their impact on the business. Upon this extensive discussion, it was possible to understand that there was no clear way of quantifying the importance of a requirement for a business as every business is different, and only a thorough analysis of this involving all the stakeholders could potentially lead to a conclusion. Upon this realisation, a much simpler approach to this evaluation method was defined with this principle in mind - if the elaborated system can answer exactly the same business queries as the original system would and more, it is clearly possible to state that the new one will be beneficial. In other words, if the requirements of the baseline DSS are a subset of the requirements of the newly developed DSS, it is safe to state that the latter is more beneficial.

Upon having both Data Marts developed , an evaluation method defined, and clear differences on the presentation layer, a very straightforward comparison was made, showing the added benefits of this approach in what concerns answering business queries and transversal impact on the Data Mart from hierarchies to dimensions to facts.

With this comparison, the main takeaway is that by following DSS design best practices, the missing requirements table (table 3.1) would, in most cases, categorize missing requirements always as data quality issues, has all the dimensions, metrics, aggregation levels etc. should be within the DSS design but would potentially have missing values in case there was no operational data that could be the source for those attributes or measures. The only case in which we would have missing context or measures would then be if the context around the business changed and, therefore, also the requirements for the DSS.

Taking into consideration the information gathered in the Demonstration and Evaluation (Chapter 4), it is possible to answer the Research Questions defined at the beginning of this work.

5.1 Analysis of the Research Questions

The main objective of this dissertation was to be able to answer the following research questions:

[RQ1] How does the integration of DSS requirements in the operational systems design impact the decision-making process?

[RQ2] How does the integration of the DSS requirements in operational systems design affect the design of the latter?

[RQ3] Why does the integration of DSS requirements in the operational systems design impact the decision-making process?

RQ1. The impact of the DSS Requirements integration on the early designs of operational systems will always prove to be beneficial, as there is always room for improvement in every system. This approach will most likely allow many business queries to be answered that would otherwise not be. Looking perhaps into requirement I3 from Table 3.1, if this analytic need was known at the time of the operational system development, the integration of a percentage attribute within the operational tables for the axis and themes would expectedly be a small effort. Even if this was not the case, and this requirement made the system development more complex, it would still make for a more complete operational system that would be more in line with the business events it is capturing. Nevertheless, throughout this dissertation, there were not only missing requirements due to operational system flaws (Table 3.1), but many of these missing requirements were also either due to business process or DSS design flaws. With this, if the operational systems allows for it, both enforcing change in the processes and applying the best practices of DSS design will also have a great impact (table 4.1) at a potential lesser cost than it would be to make changes to an operational system.

RQ2. Taking in consideration the operational systems, the changes that may applied to it will ultimately have no effect on it or potentially have a slight negative effect. An operational system that collects transactional data will keep on doing that with or without these requirements and its purpose will keep on being fulfilled in the same way. Nevertheless, if the business looks at the Operational System and DSS as both being a part of a bigger thing that is an Information System capable of fulfilling needs on both sides, we can clearly see benefits on this approach.

RQ3. As DSS requirements are gathered disregarding the operational systems, many of these requirements may not be available to retrieve on the further steps of the development process. Therefore, integrating these requirements, especially in earlier stages of the development of an

operational system in order to be more cost-effective, will allow for a higher number of business queries to be answered and consequently for a more informed decision-making process. A clear example of this within this dissertation is the requirement I2 (Cost of projects that were not approved) from Table 3.1, as the results of this can be seen in figure 4.1. This analytic capability is directly related to the change of the operational system, and had there not been a change to the operational system, and there would never be an answer to this query.

5.2 Future Work

In what concerns Future Work, a thorough analysis that manages to quantify the cost benefits of the approach used in this dissertation would be very beneficial.

Naturally, businesses will look at the cost indicator more than any other, proving that this assessment could be financially beneficial rather than making an assessment of the DSS requirements in later stages.

On the other hand, developing a framework for deciding which missing requirements to add to an in-production operational system would also be of great value as it would become easier to prioritize and make those decisions on the business side.

In order to do this, both stakeholders/decision makers and operational system developers need to be taken into consideration and with both inputs, an approach much like the one by Kimball [35] (see Figure 5.1) is suggested., Although in this case, the Feasibility would be measured for the fulfilment of the integration of the requirements on the operational system rather than the focus on the DSS as was intended by the author.

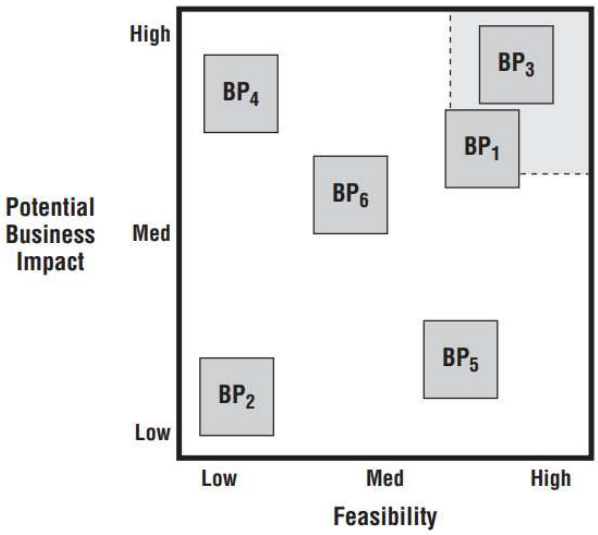


Figure 5.1. Prioritization grid based on business impact and feasibility. [35]

Bibliography

[1] S. T. March and A. R. Hevner, "Integrated decision support systems: A data warehousing perspective," *Decis. Support Syst.*, vol. 43, no. 3, pp. 1031–1043, 2007.

[2] L. Corr, "Agile BI: Welcome to the Business Model Generation," 2013. [Online]. Available: https://tdwi.org/Articles/2013/07/23/Agile-BI-Business-Models.aspx?Page=2.

[3] P. G. W. Keen, "Adaptive Design for Decision Support Systems," *ACM SIGMIS Database*, vol. 12, no. 1–2, pp. 15–25, 1980.

[4] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *J. Manag. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, Dec. 2007.

[5] D. L. Parnas, "SOFTWARE ENGINEERING PRINCIPLES.," *INFOR J.*, vol. 22, no. 4, pp. 303–316, Nov. 1984.

[6] C. Ghezzi, M. Jazayeri, and D. Mandrioli, *Fundamentals of Software Engineering*, 2nd Ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[7] I. Sommerville, *Software Engineering, Global Edition*, 10th ed. NOIDA: Pearson Education Limited, 2016.

[8] James A. O'Brien; George M. Marakas, *File Management in Management Information Systems*, 10th ed. New York: McGraw-Hill/Irwin, 2011.

[9] R. S. Pressman, *Software Quality Engineering: A Practitioner's Approach*, 7th ed., vol. 9781118592. McGraw-Hill Higher Education, 2010.

[10] Software Product Quality, ISO/IEC 25010, 2011.

[11] R. Lagerström, P. Johnson, and M. Ekstedt, "Architecture analysis of enterprise systems modifiability: A metamodel for software change cost estimation," *Softw. Qual. J.*, vol. 18, no. 4, pp. 437–468, 2010.

[12] C. Larman and V. R. Basili, "Iterative and incremental development: A brief history," *Computer*, vol. 36, no. 6. pp. 47–56, 2003.

[13] S. Balaji and M. S. Murugaiyan, "Waterfall vs v-model vs agile : A comparative study on SDLC," *WATERFALL Vs V-MODEL Vs Agil. A Comp. STUDY SDLC*, vol. 2, no. 1, pp. 26–30, 2012.

[14] R. Budde, K. Kautz, K. Kuhlenkamp, and H. Züllighoven, "What is prototyping?," *Information Technology & People*, vol. 6, no. 2–3. pp. 89–95, Feb-1992.

[15] K. Beck *et al.*, "The Agile Manifesto," 2001. [Online]. Available: http://agilemanifesto.org/.

[16] D. Larson and V. Chang, "A review and future direction of agile, business intelligence, analytics and data science," *Int. J. Inf. Manage.*, vol. 36, no. 5, pp. 700–710, 2016.

[17] V. Chandra, "Comparison between Various Software Development Methodologies," *Int.*

J. Comput. Appl., vol. 131, no. 9, pp. 7–10, 2015.

[18] R. R. Young, "Risky Requirements," *J. Def. Softw. Eng.*, no. April, pp. 9–12, 2002.

[19] S. McConnell, *Software Project Survival Guide*, 1st ed., vol. 2009. Microsoft Press, 1998.

[20] S. Tiwari and S. S. Rathore, "A Methodology for the Selection of Requirement Elicitation Techniques," 2017.

[21] S. Chaudhuri and U. Dayal, "An Overview of Data Warehousing and OLAP Technology," 1997.

[22] R. Kimball, L. Reeves, M. Ross, and W. Thornthwaite, *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data*. John Wiley & Sons, Inc., USA., 1998.

[23] R. Sherman, *Business Intelligence Guidebook*. 2015.

[24] Lawrence Corr; Jim Stagnitto, *Agile Data Warehouse Design: Collaborative Dimensional Modeling, from Whiteboard to Star Schema*, 1st ed. Leeds: DecisionOne Press, 2012.

[25] D. Prakash and N. Prakash, "A multifactor approach for elicitation of Information requirements of data warehouses," *Requir. Eng.*, vol. 24, no. 1, pp. 103–117, 2019.

[26] N. Prakash and D. Prakash, *Data Warehouse Requirements Engineering - A Decision Based Approach*, 1st ed. Springer, 2017.

[27] R. Gaskill, H. E. Van Auken, and R. A. Manning, "A Factor Analytic Study of the Perceived Causes of Small Business Failure," *Cemi.Com.Au*, 1993.

[28] R. Kimball and J. Caserta, *The Data Warehouse ETL Toolkit - Practical Techiniques for Extracting, Cleaning, Conforming, and Delivering Dat*, 1st ed. Wiley Publishing, Inc., 2004.

[29] S. H. A. El-Sappagh, A. M. A. Hendawi, and A. H. El Bastawissy, "A proposed model for data warehouse ETL processes," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 23, no. 2, pp. 91–104, 2011.

[30] J. Barateiro and H. Galhardas, "A Survey of Data Quality Tools," *Datenbank-Spektrum*, vol. 14, pp. 15–21, 2005.

[31] Y. Huh, F. Keller, T. Redman, and A. Watkins, "Data quality," *Inf. Softw. Technol.*, vol. 32, no. 8, pp. 559–565, Oct. 1990.

[32] P. Berander and A. Andrews, "Requirements prioritization," in *Engineering and Managing Software Requirements*, Springer Berlin Heidelberg, 2005, pp. 69–94.

[33] A. Shahin and M. A. Mahbod, "Prioritization of key performance indicators: An integration of analytical hierarchy process and goal setting," *Int. J. Product. Perform. Manag.*, vol. 56, no. 3, pp. 226–240, 2007.

[34] P. Achimugu, A. Selamat, R. Ibrahim, and M. N. R. Mahrin, "A systematic literature

review of software requirements prioritization research," *Information and Software Technology*, vol. 56, no. 6. Elsevier BV, pp. 568–585, 01-Jun-2014.

[35] R. Kimball and M. Ross, The Data Warehouse Toolkit, Indianapolis, Indiana: John Wiley & Sons Inc., 2013.