NOVA IMS Information Management School

MGI

Mestrado em Gestão de Informação
Master Program in Information Management

Development of a visual analytical solution

Dashboards creation for Customer Care Area

Miguel Martins Travessa

Project Work report presented as partial requirement for obtaining the Master´s degree in Information Management, with a Specialization in Information Systems and Technologies Management

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

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

DEVELOPMENT OF A VISUAL ANALYTICAL SOLUTION

by

Miguel Martins Travessa

Project Work report presented as partial requirement for obtaining the Master´s degree in Information Management, with a specialization in Information Systems and Technologies Management.

Advisor: Professor Miguel de Castro Neto, PhD

Co-Advisor: Professora Maria Manuela Simões Aparício da Costa, PhD

November 2018

ACKNOWLEDGEMENTS

I would like to thank to all the Professors in the Information Management master program for all the knowledge and interest in the subjects given, in particular to the Professors Miguel Neto and Maria Manuela da Costa, for all their availability and opportunity to develop a thesis about a theme that is very dear to me.

To my girlfriend Sofia and both our families, for always encouraging me to finish this thesis, a huge thank you. To Tiago and João, for always find a time to talk with me about the best way to code, your help is most welcomed and to Sérgio and Linda for helping me with the correct way to express myself.

Finally, I would like to thank Generic Company for giving me the opportunity to study these subjects and the time to conclude them with excellence.

ABSTRACT

Due to the technological revolution in the last decades, organizations nowadays take an extra step to keep competiveness up. One of the key areas that evolved was data analysis and the subjacent tools and applications. The surging emergence of new tools and methods enabled companies to extract information from sources, which was quite difficult before due to the lack of economically viable solutions. This document intends to detail the project carried out at Generic Company, a technological consulting company. The project aims to meet the reporting needs stipulated in the business needs, integrating the data from a Service Management software and an intern application for collaborators hour's allocation. To achieve the goals, under the Design Science Research methodology, it was developed a Business Intelligence Solution supported by a software that enables the extraction, transformation and loading of data, on a custom made Data Warehouse to be used in a business analytical tool. Keeping up to the newest trends in data analytics, it will be used a visual data analysis application to interpret an accurate representation of the state in Customer Care area specifically, and the global performance in the target organization. This Masters Project presents an introduction to the context of the project, the advantages of adopting a Business Intelligence solution to an organization that have a steady income of data ready to be analysed, and the steps needed to adopt the referred solution.

KEYWORDS

Business Intelligence; Data Warehouse; Performance Indicators; Visual Data Analytics; Dashboards;

RESUMO

Devido à revolução tecnológica nas últimas décadas, as organizações de hoje em dia precisam de dar um passo extra para manter a competitividade. Uma das principais áreas que evoluiu foi a análise de dados e as ferramentas e aplicações subjacentes. Através destas novas ferramentas e métodos foi possível que as empresas extraíssem informações dos seus dados, o que antes era bastante difícil dado a falta de soluções economicamente viáveis. Este documento pretende detalhar o projeto realizado na *Generic Company*, uma empresa de consultoria tecnológica. O projeto, em suma, visa atender às necessidades de negócio através da apresentação de relatórios interativos. Para esse fim, serão integrados os dados de uma aplicação de gestão de serviços e de um aplicativo interno para a alocação de horas dos colaboradores. Para atingir os objetivos estipulados, sob a metodologia *Design Science Research*, foi desenvolvida uma solução com base nos conceitos de *Business Intelligence*, suportada por um aplicação que permite a extração, transformação e carregamento de dados, em um Armazém de Dados customizado, para posteriormente utilizar os mesmos dados em uma ferramenta de análise de negócio. Acompanhando as mais recentes tendências em análise de dados será usado um aplicativo de análise de dados visual para representar o estado da área de Suporte ao Cliente, especificamente, e o desempenho da organização, globalmente. Este Projeto de Mestrado apresenta uma introdução ao contexto do projeto, os benefícios ao adotar uma solução de *Business Intelligence* em uma organização e as etapas necessárias para adotar a solução referida.

PALAVRAS-CHAVE

Business Intelligence; Armazém de dados, Indicadores de Performance; Análises de Dados Visuais; *Dashboards*

INDEX

1. Introduction 1
   1.1. Problem Definition and Motivation 2
   1.2. Organization Description 3
   1.3. Study Goals 3
2. Literature Review 5
   2.1. Business Intelligence 5
   2.2. Information Management 8
   2.3. Data Warehousing 11
   2.4. Dashboards Production 12
   2.5. Information Visualization 15
   2.6. Design Science Research Methodology 22
3. Conceptual model Proposal 27
   3.1. DSR Methodology 27
      3.1.1. Constructs 27
      3.1.2. Methods 27
      3.1.3. Models 28
      3.1.4. Instantiation 30
   3.2. Project Management 32
      3.2.1. Project Phases and Tasks 32
      3.2.2. Development plan and Milestones 33
4. Dashboard Prototype Development 34
   4.1. Environment 34
      4.1.1. Resources 34
      4.1.2. Organization 34
   4.2. Database Design 34
      4.2.1. Source Database 35
      4.2.2. Staging Area Database 42
      4.2.3. Data Warehouse Database 45
   4.3. Data Manipulation – Extraction, Transformation and Loading 51
      4.3.1. Designed SQL Functions 51
      4.3.2. Source ETL 55
      4.3.3. STG ETL 55
      4.3.4. DW ETL 56

4.4. Dashboard Design....56
4.4.1. Data Model....56
4.4.2. ETL *on-the-fly*....57
4.4.3. Measures....58
4.4.4. Report....58
4.5. Results and validation....63
5. Project Relevance....67
6. Conclusions....69
7. References....71
Appendix A....74
Visual Studio BI Data Tools....74
Source ETL....74
SA ETL....75
DW ETL....83
Appendix B....84
Operational Dashboard....84
Customer Support Dashboard....85
Clients Dashboard....86
Collaborators Dashboard....87
Projects & Contracts Dashboard....88

LIST OF FIGURES

Figure 1 – Business Intelligence IT Architecture (adopted from (Chaudhuri et al., 2011)) 5
Figure 2 - Business Intelligence benefits (adopted from (Sturdy, 2012)) 7
Figure 3 – Layers of performance indicators (adopted from (Parmenter, 2007)) 9
Figure 4 - The Relative value of data analysis tool (adopted from (Lavalle et al., 2010)) 13
Figure 5 – Types of visual analytical interfaces (adopted from (Eckerson, 2009)) 14
Figure 6 – Technical and Business Architecture in a Performance Management System (adopted from (Eckerson, 2009)) 15
Figure 7 - Line and Bar Chart (charts retrieved from (Microsoft, 2018b)) 18
Figure 8 - Stacked Bar chart (charts retrieved from (Microsoft, 2018b)) 18
Figure 9 - Bar and Line chart combined (charts retrieved from (Microsoft, 2018b)) 19
Figure 10 - Scatter Plot chart (charts retrieved from (Microsoft, 2018b)) 19
Figure 11 - Treemap chart (charts retrieved from (Microsoft, 2018b)) 20
Figure 12 – Pie Chart and Area Chart (charts retrieved from (Microsoft, 2018b)) 20
Figure 13 – Dashboard Canvas relevant areas 21
Figure 14 – Nunamaker, Chen, & Purdin approach to IS Research 22
Figure 15 – Hevner (2007) proposed Design Research cycles. 24
Figure 16 – Project Conceptual model 28
Figure 17 - Dashboard Canvas 29
Figure 18 – Project solution prototype 31
Figure 19 - Source Database Diagram 35
Figure 20 - Source Database Entity Classification 43
Figure 21 - Staging Area Diagram 45
Figure 22 – Data Warehouse Diagram 46
Figure 23 - Designed function COMPUTUS 52
Figure 24 - Designed function UFN_CALC_TIME 53
Figure 25 - Designed function InitCap 54
Figure 26 - Designed function INSTR 55
Figure 27 – PowerBI Data Model 57
Figure 28 – Operational Dashboard 59
Figure 29 – Customer Support Dashboard 60
Figure 30 – Clients Dashboard 61
Figure 31 – Collaborators Dashboard 62
Figure 32 – Projects & Contracts Dashboards 63
Figure 33 - Average Hours passed per Ticket Handler 64

Figure 34 – Tickets overview and effective time spent to resolve 65
Figure 35 – Average time to resolve and respond to Incidents 65

LIST OF TABLES

Table 1 – Decision Support applications (adopted from (Turban et al., 2011)) 13
Table 2 – Design Science Research Methodological Approach 25
Table 3 - Project Phases 32
Table 4 - Project phases chronogram 33
Table 5 – Source table OTRS_TICKET 36
Table 6 – Source table OTRS_STATE 36
Table 7 – Source table OTRS_TICKET_HISTORY 37
Table 8 – Source table OTRS_QUEUE 37
Table 9 – Source table OTRS_TICKET_TYPE 37
Table 10 – Source table SMRS_ACTIVITY 38
Table 11 – Source table SMRS_CONTRACTS_CATEGORY 38
Table 12 – Source table SMRS_USERS_TEAM 38
Table 13 – Source table SMRS_CONTRACT_REPORTS 39
Table 14 – Source table SMRS_PROJECT_TYPE_REVENUE 39
Table 15 – Source table OTRS_LOCK 40
Table 16 – Source table OTRS_TICKET_TIME_ACCOUNTING 40
Table 17 – Source table OTRS_USERS 40
Table 18 – Source table SMRS_CONTRACTS 41
Table 19 – Source table SMRS_PROJECTS 41
Table 20 – Source table Link_Clients_SMRS_OTRS 42
Table 21 – DW table DIM_CLIENTS 47
Table 22 – DW table DIM_DATE 47
Table 23 – DW table DIM_QUEUES 47
Table 24 – DW table DIM_TICKET_TYPE 48
Table 25 – DW table DIM_EMPLOYEES 48
Table 26 – DW table DIM_STATUS 49
Table 27 – DW table DIM_PROJECTS 49
Table 28 – DW table DIM_TICKET 50
Table 29 – DW table FACT_OTRS 50
Table 30 – DW table FACT_SMRS 51

LIST OF ABBREVIATIONS AND ACRONYMS

BI	Business Intelligence
DS	Data Sources
DB	Database
DW	Data Warehouse
DSR	Design Science Research
ETL	Extract Transform Load
ICT	Information and Communication Technologies
IT	Information Technology
NOVA IMS	NOVA Information Management School
OTRS	*Open-Source Ticket Request System*
SA	Staging Area
SMRS	*Service Management Reporting Service*
SQL	Structured Query Language
SSIS	SQL Server Integration Services
STG	Staging

1. INTRODUCTION

Due to the rapidly changing environment in technological companies, with increasingly complex and competitive scenarios, it is up to organizations to find a way to respond efficiently to the challenges they face each day. It is the type of answers a company gives to its challenges that defines its competitive position, differentiating it from the others (Elgendy & Elragal, 2016). Chaudhuri, Dayal, & Narasayya (2011), identified that the evolution and variety of technology that is present nowadays enabled organization to acquire data from multiple sources and storage it much more efficiently (with a significant reduction in its maintenance costs), which in turn led the organization to acquire large volumes of data hoping to pull the maximum amount of information that could give an advantage against their business competitors. With this untapped Information and using the correct methods to analyse it, the management area in a business unit can perceive the information in an easier way and, ultimately, become decision data-driven organization (Karlsen, Eidene, & Olsen, 2012)

The projects aims to present a Business Intelligence solution by delivering customized dashboards for a Customer Care business unit in the area of the IT consultancy for Small and Medium-sized Enterprise (SME), where the author is currently working. Taking in consideration the master's degree and its area of specialization, this project enables the possibility to analyse the various systems that compose the supporting level in an organization and how to integrate the information available, through the exploration of the subjacent concepts in BI and experiment live challenges and difficulties for implementing this type of solution in an organization where most performance and result indicators are still calculated by manual processes. The project will allow the development of several skills, particularly in the development of Business Intelligent systems, which is nowadays one of the most desired area in IT (Cooper, 2012).

In this context the following research question was formulated: *Can we provide a visual analytical solution to the Costumer Care management for decision making in IT functional area?*

To answer the above Research Question the author defined the following main Research Objectives for this project:

i. Develop a Business Intelligence solution that processes the available data to generate information for the company;

ii. Empowering the multiple layers of the organization, with focus on the Costumer Care area, with an efficient tool to analyze the organization performance through an visual interface with pre-defined metrics;

The solution created was developed under the Design Science Research framework and its inherent design processes (Hevner & Chatterjee, 2010). By framing the problem and identifying the benefits, it was proposed a solution based on the current literature related to Business Intelligence, Performance Management and Dashboarding for the Costumer Care area. Through the assessment of the referred literature available for the scope of the problem it will be defined the conceptual model for the development of the final prototype.

The Business Intelligence solution proposed for Generic Company case study will analyse information from OTRS (2018), a Service Management software where all the requests made by the clients that

have maintenances services by contract are registered, and SMRS, an in-house application where all the organization collaborators register the hours allocated to their clients. Having this kind of information available enables the company to analyse the data collected by it, in order to facilitate the performance evaluation of its services and to get a deep knowledge of the business model adopted. It is in this context that this project is framed, to present a visual analytical solution, represented by a dashboard, that contains a set of KPI's, result indicators and performance indicators forming reports with up-to-date values, aligned with the business goals, for different time frames, allowing to infer a superior data analysis at Generic Company.

This document is structured in seven major chapters. Being the first chapter and introduction to this project report, the problem the author proposes to solve and the global objectives needed to be concluded to solve the problem. The second chapter will present the Literature Review used in the project scope and how it applies to the concepts used during the development of this project according to the best practices used nowadays. For the third chapter it will be contextualized in the Design Research Methodology with the solution and presented a conceptual model and definition of the project phases to create the final product described in the fourth chapter, where the dashboards created will be shown and their results will be compared to the results the company previously had. The fifth and sixth chapter correspond to the correlations of the literature review with the artefact developed and the final conclusion for this project thesis. Finally, the seventh chapter contains the list of references used.

1.1. Problem Definition and Motivation

The Generic Company to which this project is related has a Costumer Care area that deals with technical support for the applications made by the Generic Company or that the organization has proficiency in it. The Generic Company use multiple applications for the sole purpose of the registration of its activities, yet, it doesn't have a centralized view to accompany or measure its operation, resulting that all the performance or result indicators (Parmenter, 2007) necessary for board meetings and business assessment are made *ad-hoc* without an automated process to support it. This working model has several restrictions such as: the need of an available operator to extract the data and time it consumes doing so, the information quality depends on the time the information was extracted and the inefficient usage of server storage space with old reports. Therefore, it is necessary to enable the business unit with the tools and means to effectively and efficiently analyse the information available.

At Generic Company it is possible to identify sources of data, related with the Costumer Care area, with an enormous potential to develop metrics that could provide an holistic view of the performance of certain business units that lack analysis and give a clear representation about the ongoing operations and the inexistence of objective indicators that could support the perception of the improvement, or downgrade, in the organization's performance.

With the development of a BI solution we can map every client registered at Generic Company to analyse all the relevant aspects related with the on-going operations, supported by the data of different data sources, becoming possible to evaluate the collaborators' performance for each project, with indicators such as: the average time for request resolution or the number of closed request by client/collaborator. It is expected that the BI solutions will impact the organization at the horizontal

level, enhancing concepts such as: collaboration, information sharing, flexibility, responsiveness, agility, and customer intimacy (Sturdy, 2012).

It is in this context that the project is framed, the development of a BI solution that will use the data made available by OTRS and SMRS and provide insights to all the management levels and specifically to the Customer Care area.

1.2. ORGANIZATION DESCRIPTION

To effectively coordinate and record all the activities done to their clients, in 2012, Generic Company adopted a Service Management Solution, OTRS, that keeps record of all the tickets and requests made by the clients or collaborators that identifies the corrective actions to be made according to the situation. This software has the ability to export its records to an Excel file (Microsoft, 2018a) for anyone that wants to do small analyses of the overall month performance or specific subjects. So far, all the analyses were made by hand.

In mid-2016, the SMRS, Service Management Reporting Service, was developed. This in-house Outsystems application (Portuguese low-code platform used to visually develop custom made applications) synchronizes by web-service with Replicon (2018), a web-based platform for time-tracking and project management where collaborators can manage their time allocation for clients and time-off. Besides the integration with Replicon, SMRS is able to associate each request logged in OTRS and associate it to the project defined in the Replicon.

In the end, we have two applications that keep track of different concepts associated to the client, the time used for problem solving and the time registered for billing. Most of the projects done in the organization are not registered in OTRS. Because it is a service management software it is mostly used for clients that have maintenance contracts.

Currently, the reports made with the available data are done in PowerPivot (Microsoft, 2018c), MS Excel Add-On that is used to analyse and process large volumes of data from multiple Data Sources. A well designed BI solution will result in solving the previously identified flaws, offering the possibility to obtain reports and analyses with the same results for the designated users at the same time in an intuitive form, resulting in a significant productivity boost (Magdalena & Luminita, 2012).

1.3. STUDY GOALS

Regarding the problem explained in the Introduction and the main objectives defined for the purpose of this project, we can divide this objectives in the following milestones required to achieve the purpose of this project:

i. Create a Business Intelligence engine that process the data stored in the platforms used by the company;

ii. Automatic integration between the Data Sources and the BI solution Data Warehouse.

iii. Develop a scalable solution that could be used for future necessities;

iv. Create a solution that is cohesive and feasible regardless of changes in contracts, projects or associated collaborators;

v. Define all the result indicators, performance indicators and KPI´s (set of metrics that evaluates and compares the goals defined in the business Strategy to what actually exists (Parmenter, 2007));

vi. Ensure the integrity of the data extracted and validation of the chosen metrics;

vii. Present a visual analysis solution that enables users to access up-to-date dashboards that contain all information needed for business analysis in a clean presentation;

viii. Enable the managerial level of the organization an efficient analysis regarding the performance of the company Costumer Care operations.

ix. Deployment and configuration of the solution for all the key-users;

2. LITERATURE REVIEW

The literature review presented in this chapter take references the theoretical concepts inherent to the development of BI project followed by what is defined nowadays as the best practices applicable. The goal of this chapter is to thoroughly justify the paths taken during the development of this project and contextualize the different phases to guarantee the success of its implementation.

2.1. BUSINESS INTELLIGENCE

The Business intelligence concept was created to satisfy management level requests of organizations, so they could efficiently and effectively analyze the data in the enterprise in order to understand the situation their business is in (Karlsen et al., 2012).

Business Intelligence cannot be simply defined by the association to a tool or Information System but rather to a complex business-oriented concept that has as its main goal the transformation of data in information and afterwards into knowledge, which adds value to the organization (Chaudhuri et al., 2011). Its application involves the utilization of IT infrastructures, data models and applications that allows the conditions for real-time and interactive access to the analysis and management process of information generated through transactions, processes, activities, or even external sources, and stored in a way that can be accessible and allow an analysis over time (Santos & Neto, 2011).

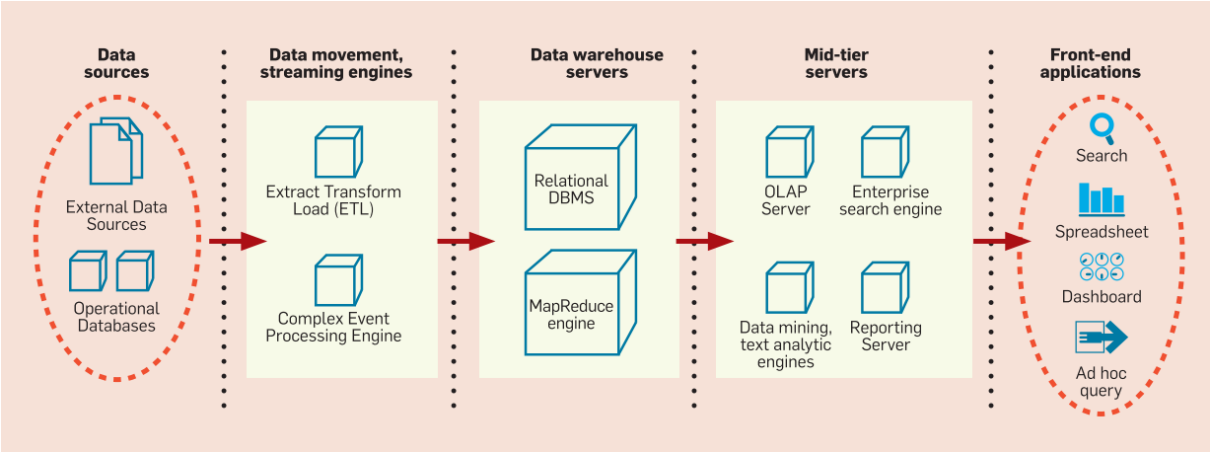


Figure 1 – Business Intelligence IT Architecture (adopted from (Chaudhuri et al., 2011))

To make a clear identification of a BI project, we have to make a distinction between its components and their use as a whole, forming a complete Business Intelligence solution. The software is defined as a product available in the market used in a specific context, the tools and applications used are software products installed and configured for usage in a particular purpose. The BI solution will include all of the above but acts as a collective of applications to reach a concise goal(s). It is essential also to refer that this solution must run under certain conditions defined by the restrictions subjacent to IT infrastructure, such as operating systems, servers, networks and integration platforms (Wieder & Ossimitz, 2015). By defining these concepts, we can formulate what is going to be done in this project: It will perform a certain work, with precise specifications, to enable interactive access to data, in order to allow its analysis.

Hostmann, Ryaner, & Herschel, (2009) identified Business Intelligence solutions as a critical factor for companies aware of market changes since it allows them to adapt and define which strategy to follow based on credible and sustained information instead of just the organizations management level instincts. Usually, authors define Business Intelligence as a technological supported process that reunites all the relevant informant through the centralization and transformation of the available, but dispersed, data for analytical purposes (Barrento, Neto, & Caiado, 2013). The purpose behind this actions is to gain insights and relevant information about business goals, opportunities and support the decision-making at the organisation managerial level, as it is stated in Wieder & Ossimitz (2015).

Still, it is necessary to ensure that the designated BI solution will have the desired impact and for Vodapalli (2009) and Kfouri & Skyrius (2016) these critical factors were identified for the success of a BI implementation:

i. As it was stated previously, there must be a clear business-oriented vision with the anticipation of the future needs and how the solution will respond to them;

ii. The sponsorship from the management level in the organization and the commitment of the organization to act accordingly in the processes that enable a superior analysis.

iii. A reliable Data Management that ensures the quality of the information retrieved;

iv. A solution that meets the demands and needs;

v. A IT infrastructure that ensures a steady performance;

The introduction of BI in an organization is justified by the ability to adapt to the multitude of areas that compose an organization and their businesses. There could be a set of business processes and activities in each organization's business model directly dependent of the inputs generated by solutions that act in the BI domain and are critical for decision making, highlighting the fact that the value of the solution is directly correlated by its scope in the organizational context (Côrte-real, Neto, & Fátima, 2012). A BI solution focused on the organization goals will lead to relevant information available in an early phase which in turn will help identify gaps in the analytical model. Also, it is possible to identify other benefits in this kind of models. The time spent on data cleaning processes can be spent analyzing the importance of the information retrieved and formulate new questions to answer.

In the context of Customer Care management the BI solution intents that its application will result in improvements to the business area and will have outcomes such as reduced customer acquisition costs, improved customer segmentation, customer acquisition, customer retention increased revenues, and increased customer lifetime value, which in turn results in revenues (Sturdy, 2012) as it was shown in figure 2.

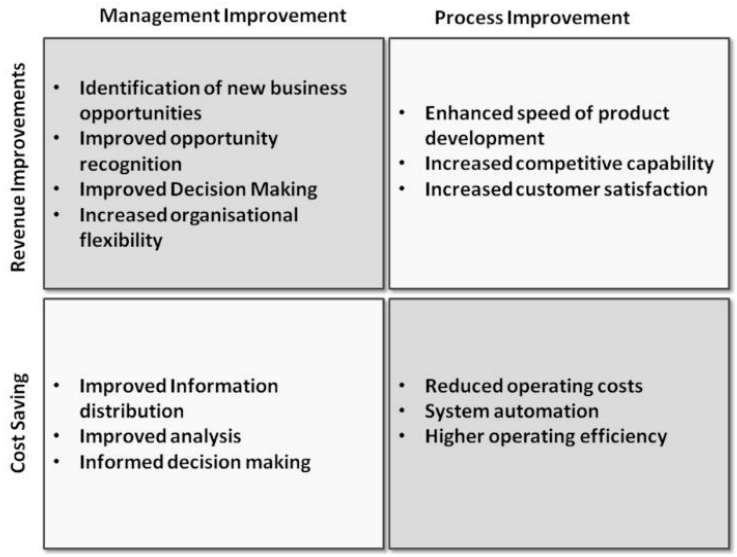


Figure 2 - Business Intelligence benefits (adopted from (Sturdy, 2012))

The previously identified benefits can be contextualized in the core areas in which a BI solution acts, as it was identified by Magdalena & Luminita (2012):

i. **Operational efficiency**, optimization of the organization´s operations;

ii. **Profitability,** enables decisions based on factual information and the identification of non-profitable elements;

iii. **Cost reductions**, reduce the number of operators for some processes and the time associated with those tasks;

iv. **The productivity of employees**, all the information is centralized so that the employees are more available to focus on the important aspects of their activity;

v. **Client satisfaction**, identify and present to the clients an adequate service level, depending on the organizations' performance, increasing client retention;

The introduction of Business Intelligence projects within an organization also leads to an automatic data integration and analytics that, per se, results in the possibility to timely delivery of information to decision-makers roles in an organization. For Abbaszadegan & Grau (2015) this was a critical factor since it enabled multiple roles in its organization to benefit their business since they have the information they need in an expected time interval.

It goes without saying that many other effects of implementing these BI solutions are not visible or can't be directly measurable and the benefits can only be retrieved after a certain amount of time. These benefits are associated with strategy changes due to enhance decision-making based on improved information quality that could lead to a dispersion of benefits within the organization, affecting the business culture itself. (Côrte-real et al., 2012).

2.2. INFORMATION MANAGEMENT

Nowadays, one of the biggest challenges in an organization is how to use the available data to create insights that add value to the business. In a study made by the MIT Sloan Management Review in collaboration with the IBM Institute for Business Value, 60% of the sample interviewed stated that "organizations has more data that it can use effectively" (Lavalle, Hopkins, Lesser, Shockley, & Kruschwitz, 2010). It was taken in consideration a sample of 3000 business executives, managers and analysts from multiple companies, around the globe.

The Information Management in an organization is done by its Chief Information Officer (CIO) and it is his responsibility to assign the correct Management Information System to the business area. Management Information System (MIS) uses and manages information technologies to improve business processes, business decision-making and gain competitive advantages. The basic role of MIS is no more than a structural system in terms of regularly processed preformatted reports that before were only centralized to a more distributed and knowledge driven, user-triggered system (O 'brien, Marakas, & Lalit, 2008).

Management Information Systems can be applied to all society areas, from financial to agricultural since they are focused on the underlying processes that are business related. The MIS enters on these areas since it can be a way to discover new opportunities to improve management since it enables managers that need to make critical decision all the time and support their action based on timely, relevant, updated and organized information. Whether the decision is critically important and complicated or simple and routine, all decisions have to be based on information - which must be prompt available to decision-makers.

This systems identifies the need for data to solve problems and can unfold to form specific types of information systems such as the Decision Support Systems. This concept first appeared in the early 1970s as computer-based support system for management decision makers within an organization (Turban, Sharda, & Delen, 2011). While BI can be defined as a set of processes, technologies, platforms combined to achieve a specific end, Decision Support Systems are mainly viewed as a final product that is supposed to act on specific areas of the enterprise providing and working with the information that is associated with the area, in order to make a company data-driven.

With the evolution of technologies more concepts evolved and were created to support the decision making and it is in that context that the Performance Management Systems (PMS) appeared. As BI, PMS are more than just an application, but a set of methodologies, measures and processes related to follow-up the performance of an organization. This systems are interlinked by two concepts, the data layer that collects all the information from the multiple systems that provide data to be analysed and the business layer where all the metrics and objectives are defined for the purpose of performance management (Santos & Neto, 2011). The metrics established to support the performance analysis in a business with the data that is available serves the organization's strategy and its stakeholders, linking strategy to processes (Eckerson, 2009).

For the application of a PMS, organizations should beforehand define their objectives and the necessary metrics that enable the organization to commit itself to reach its goals. Alas, the establishment of metrics should not be done light-hearted since the imposition of wrong metrics can have unpredicted consequences. The metrics should represent, objectively, the efforts made by the

collaborators within the context of the organization strategy. Without that, the recognition from their work will not be done and leave them tired and frustrated. This should be one of the biggest challenges when defining performance metrics that could be drilled down to an individual level (Petrovski, 2016).

When defining the metrics and measures, David Parmenter (2007) established a set of concepts to define clearly what must be identified in an organization for measuring effects. This *Performance Measures* are used by the managerial level of an organization to measure, report or pinpoint performance improvement, are classified in the following way:

a) Key result indicators (KRIs), present what has been done, under a perspective;

b) Performance indicators (PIs) tell you what to do;

c) Key Performance Indicators (KPIs), tell you what to do to increase performance related with the business success factors.

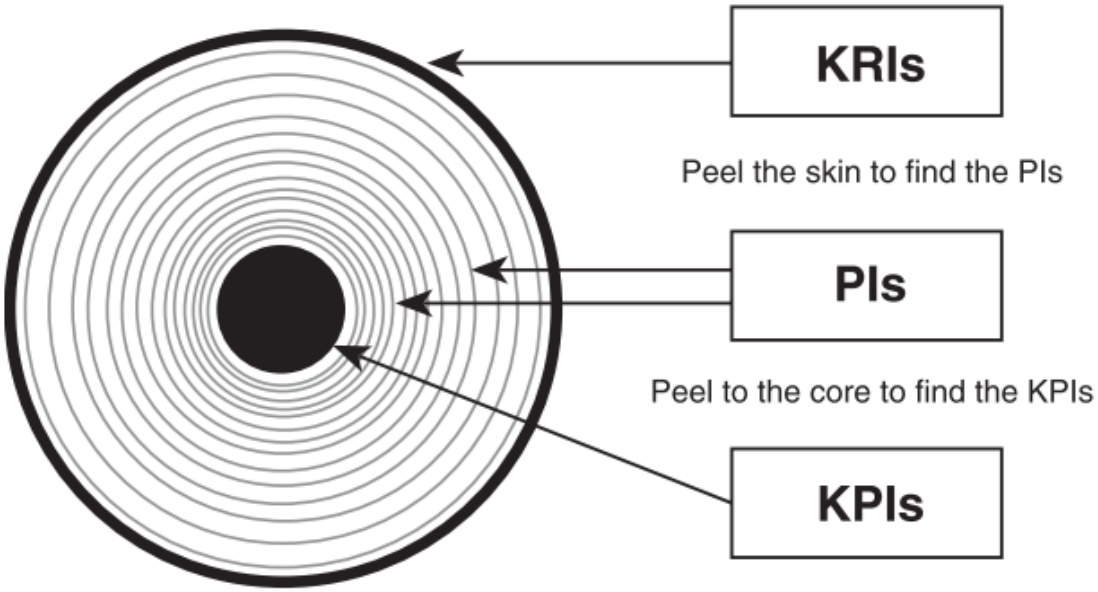


Figure 3 – Layers of performance indicators (adopted from (Parmenter, 2007))

As it is possible to comprehend from the previous figure, the deeper the indicator dwells upon the more it is interconnected with the business strategy.

The Key Results Indicators can be translated in measures that result from the combination of actions and tasks done in the enterprise, they measure the results obtained from the tasks but they do not indicate the necessity for improvement to achieve a certain goal, these type of measures are usually used to analyse large intervals of time. It is possible to identify as KRIs topics such as customer satisfaction or the profitability of clients.

Performance Indicators are between the results indicators and key performance indicators. These measures complement the KPIs and are normally associated to business constraints related with each department. These measures could include the profitability of a percentage of clients, net profit on certain products or collaborators involved in company activities.

Finally, Key Performance Indicators are measures focused on organization's performances critical for its success. The values represented in these measures are not necessarily financial but they oblige the commitment of teams or departments and have an impact in other indicators. The corrective actions required to set the measure in good terms must understandable by its audience.

All these types of measures should be presented together. Separating KRIs from other measures, and the other way around, could have a deep impact on the reporting platforms since it separates the measures that impact governance and those that impact management (Parmenter, 2007).

When defining these measures it is necessary to keep in mind that they should contain a set of attributes so the utilization of the previous indicators are clear for an organization. These attributes include:

- Status, an immediate indicator of the current indicator of the performance of the measure related with its target value;
- Trend, measures the indicator against a previous result timely framed, presenting an indicator where the audience can immediately interpret the results;
- KPI name, clear identification of what the measure is supposed to present and in what conditions;
- Actual value, what was achieved so far for the interval previously parametrized;
- Target value, what is supposed to achieve for the interval previously parametrized;
- Variance, the presentation of the difference between the actual value and the target value, this attribute should be presented in absolute value or/and in a percentage.

For the context of the Masters project, it was not possible, and it was out of the scope, to implement a system that could measure the performance of the whole organization but rather a section of it. Since there are enough systems that could support the data needs for the development of such solution there is an identification of the measures that should be made available followed by an assessment of the data that could provide the necessary inputs. Within the context of the Costumer Care for the area of technical support it was identified the following measures:

- Number of created tickets;
- Number of resolved tickets;
- Number of pending tickets;
- Number of re-open tickets;
- Number of interactions per tickets;
- Number of tickets that had interactions, per day;
- Number of collaborators involved in the ticket;
- Time taken to resolve tickets in working hours;

- Time taken to take first actions onto tickets in working hours;
- Time per type of interaction;
- Total time of interactions per ticket;
- Total time of interactions spent by the collaborator in a day for ticketing activities;
- Time registered per collaborator per ticket.

With the evolution of the tools and technologies that support the effectiveness of this management systems it is clear for the author that this measures will suffice for the scope of the project. The combination of these measures with function aggregators, such as average or maximum, and the dimensions inherent to the organization (clients, collaborators, teams) will be sufficient to create a wide field of performance indicators for analysis.

The measures defined should be clear to all the organization stakeholders since they are the ones that will influence their values. As stated by Parmenter (2017, p.13) "*The ability of any organization to pursue the best-practice path to performance improvement is determined by the effectiveness of its people practices*".

All the indicators defined and implemented in one organization are subjected of reviews over time, either at a result level to detect discrepancy or if the measure itself is still aligned with the organization strategy.

2.3. DATA WAREHOUSING

Data warehouse is a type of database focused on providing a more concentrated source of information from all the data sources available in an organization and related with a subject to help the management information for reporting and analysis across the organization (Kimball & Ross, 2007). It is expected that the use of a Data Warehouse will enable end users to access directly the information using report or analysis tool without the assistance from IT staff. The data kept in a DW is not updatable by an end-user but rather by a batch process, rendering the information available read-only.

In 2000, Moody & Kortink developed a model for designing data warehouses directed for end-users to query data directly without the need for specialized consultants to assist them. In their paper, they use *Dimensional Modelling,* previously developed by Kimball, as the more practical approach for designing data warehouses and data marts. This type of modeling aims to enable end-users to make their analyses and maximize the efficiency of information retrieval by reducing the number of tables inside a database to query.

Kimball & Ross (2007) design method focus on the user's needs and how they want their data related to defining the DW model to apply. This process starts by identifying the metrics that will be aggregated and the attributes that relate to them, with relevance to the solution, and forming star schemas based on these. In this type of model, 80% of database consultancies delegate to a single table, increasing the response time of the system for the user's queries

The basic building block used in dimensional modeling is the star schema. This schema type consists of one large central table called the fact table, and some smaller tables are called dimension tables which radiate out from the central table throught the utilization of primary and foreign keys that guarantee the rereferential integrity of the data available.

2.4. DASHBOARDS PRODUCTION

Organizations nowadays spend too much of their resources analyzing and retrieving information from the data instead of understanding how the information will benefit the company. This is observable in the context of the problem this project proposes to solve, where collaborators are obliged to register all the allocated time in its respective client, leading to a scenario that focuses to much on data management, leaving less time or resources to understand the underlying information that is possible to extract from all that data.

Organizations should implement analytics by first defining what question should a BI solution answer, regarding the business objectives. Only after that phase the company can focus on identifying the data that will answer those questions (Lavalle et al., 2010).

Moreover, to present an answer, it is first necessary to identify the stakeholders whose answers are directed to, and the granularity of information that should be available in those answers. Eckerson (2009) identified that the information made available could be contextualized in following scopes:

i. Operational information for functional personnel track their activities;

ii. Tactical/analitycal information to infer analysis on the multiple fields of the business area;

iii. Strategic information for the organization board to assess the results of the actions taken.

The type of information allied to the type of decision that must be made with the information made available differentiatiates the level of detail and structure of the problem that the a interface should assist.

Table 1 – Decision Support applications (adopted from (Turban et al., 2011))

		Type of Control		
		Operational	Managerial	Strategic
Type of decision	Structured	Accounts receivable Accounts payable Order entry	Budget analysis Short-term forecasting	Financial management Investment portfolio Distribution systems
	Semi structured	Production scheduling Inventory control	Credit Evaluation Project scheduling Reward system design	Mergers & acquisitions New product planning HR Policies Inventory planning
	Unstructured	Approving loans Operating a help desk	Negotiating Buying hardware Recruitments	R&D planning New tech development

After defining the objectives to be achieved in the solution to adopt and the audience it is directed, it is necessary to select the best way to present the information. Already in 2010, a study concluded in the *MIT Sloan Management Review* identified that data visualization would be one of the most valuable in the future years and an upcoming trend.

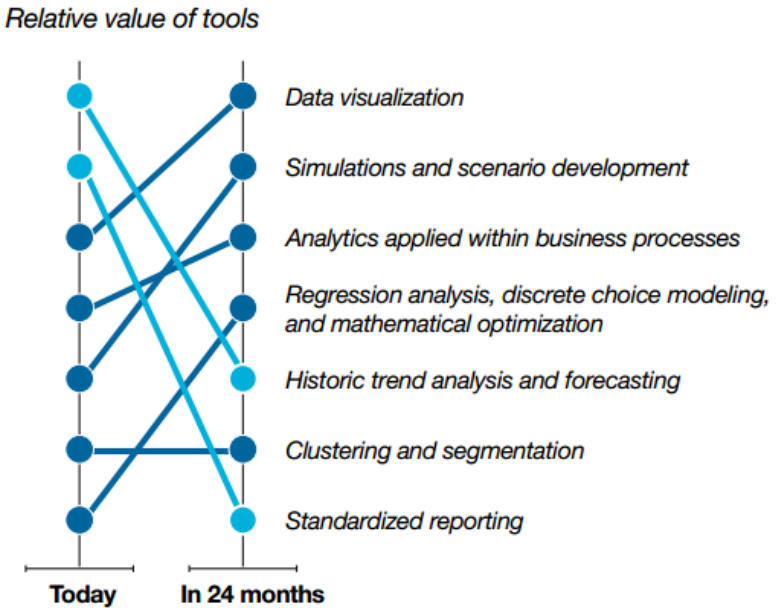


Figure 4 - The Relative value of data analysis tool (adopted from (Lavalle et al., 2010))

This data visualization in the context of this solution can be represented by dashboards. The concept of dashboard appeared to identify a way to present information through easily understandable visual objects to the audience it is destined. These dashboards could be fitted in most of the business areas, as long as there is data as input, and it enables its end-users to monitor, analyze, manage and communicate all this information as a standard procedure, transversely to all the organization (Eckerson, 2009).

Through dashboards, it is possible to:

a) Gather all the information needed to make decisions available to the stakeholders;

b) Guarantee all stakeholders consult the same information;

c) Present metrics exclusive/critical to the organization to be analyzed.

Alas, for the type of control and depending on the level of information that is available the business contents that should compose the information interface should vary according to Eckerson (2009):

	STRATEGIC	TACTICAL	OPERATIONAL
Focus	Execute strategy	Optimize process	Control operations
Use	Management	Analysis	Monitoring
Users	Executives	Managers	Staff
Scope	Enterprise	Departmental	Operational
Metrics	Outcome KPIs	Outcome and driver KPIs	Driver KPIs
Data	Summary	Detailed/summary	Detailed
Sources	Manual, external	Manual/core systems	Core systems
Refresh cycle	Monthly/quarterly	Daily/weekly	Intraday
"Looks like a..."	Scorecard	Portal	Dashboard

Figure 5 – Types of visual analytical interfaces (adopted from (Eckerson, 2009))

Aside from the level of information that is presented in a dashboard, the composition of the dashboard must be directed for a specific purpose. In a business intelligence dashboard, it was identified three clear purposes for its utilization, which are, reporting, scrutinizing, and discovering. Reporting could be compared to the monitoring tasks, where the reason for the dashboard is clear for the users and it is used to monitor the performance. Scrutinizing dashboards are used by managers to find proof of some indicators that are not explicit, while discovering is the hardest type of dashboard since it should present information that is not directly readable and uses typically the combination of the data available in multiple areas of an organization to present indicators that could add value to the business, usually this type of dashboards are supported by powerful data processors algorithms. By contextualizing this type of purposes in the Masters Project, it is safe to assume that the dashboards presented fall in the first and second category.

Alas, it is always necessary to keep in mind that the dashboards to be produced must be on a pair with the level of interpretation of its users.

The performance dashboard contains its measures in a visual information interface system that enables its audience to analyze, monitor, and assess the impact their decisions have done toward the enterprise strategic objectives. It identifies business related events, defines a strategy, measures the outcomes and adjusts the management in conformity with the goals, with the input of the information as the guidance for the actions.

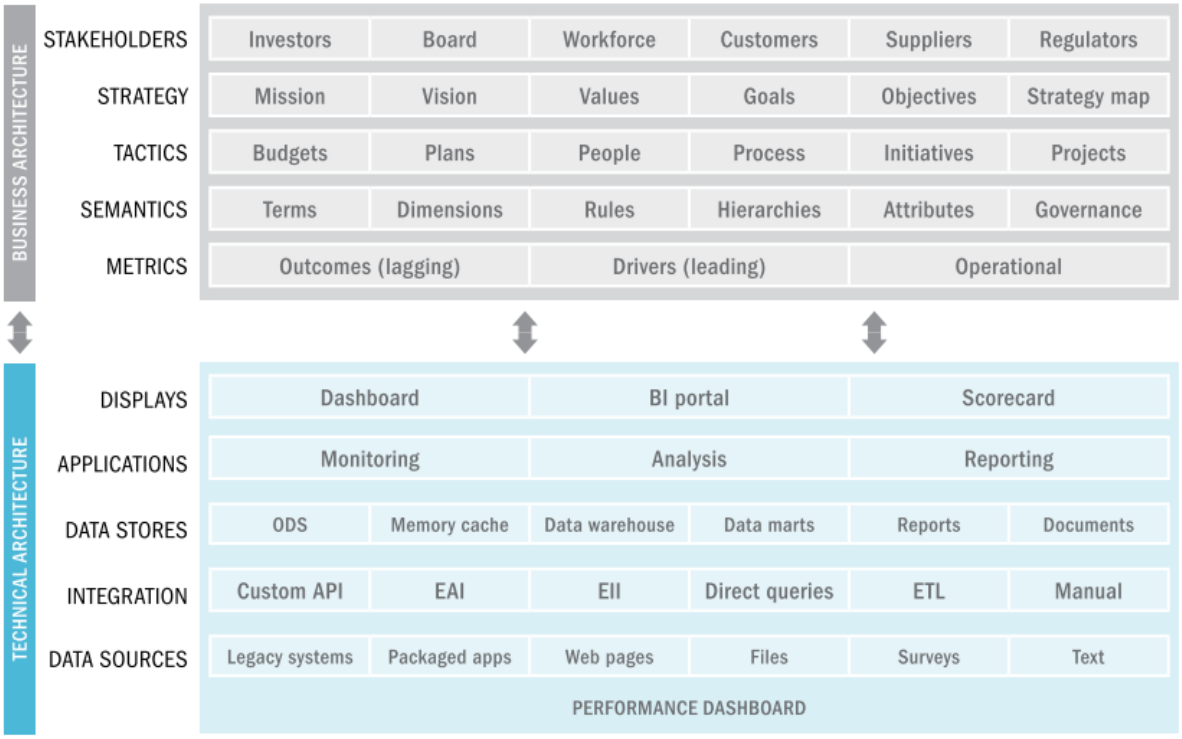


Figure 6 – Technical and Business Architecture in a Performance Management System (adopted from (Eckerson, 2009)*)*

The managerial level must also understand that the indicators that compose the dashboards are there to help improve the effectiveness and efficiency of an organization and not to penalize collaborators. The performance measures should serve as an early indicator to what is wrong and that there is a need to act before it gets worst, and not to point out that things are not performing as expected.

Alas, another important technical subject that the dashboards should be able to do is the ability to present multiple filter methods that enables the company to assess its current situation against its objectives for a specific time interval. Meaning, the solution should be able to collect both near-time data and real-time data, thus enabling that organization to understand its real performance and as such, make informed decisions.

2.5. Information Visualization

Information visualization appears in the context of Business Intelligence through the need to present the audience with information for analysis in a clear, easy and readable way. This concept uses the inputs of the impacts of data perception from the use of some elements such as colours, shapes, and the advances made in Information Systems areas to improve the quality of the information in it. Consequently, through the combination of multiple science fields the final output can be translated in dashboards, as a mean to visually represent data (Aparicio & J. Costa, 2014).

A visual analytical solution should be adapted to the needs of the future users, in this context, the selection of the platform and the contexts should show what is strictly necessary so the user doesn't lose its focus, for example, a user that is required to keep track of the product outcome in a factory

could use a BI solution that would be possible to be acessed in its smartphone while a company manager would use a desktop dashboard for a wider display of the organizations results (Campos, Sharma, Gorostegui, Jantunen, & Baglee, 2017).

In 2007, Few indicated that the dashboard development should be aligned with business to reflect, efficiently, the message it is supposed to send. For that purpose, the visualization should not be placed randomly on the dashboard and should follow a natural order that empowers the user to correlate events and infer superior analysis by reducing the difficulty of the data interpretation.

This dashboard will defined by a composition of elements such as:

- Graphs, mainly two axials elements but can even be one axis. Most graphs are common between dashboarding tools but some editors can provide different elements;
- Images;
- Icons, simple images that should indicate if some subject should have special attention from the dashboard reader;
- Text, mostly used in dashboard for labelling but can also to communicate single values in a simpler way;
- Organizers, represented by tables, geo-referenced graphs or elements that aggregate events related;
- Drawing Objects, normally the combination of the items above and serve to establish a relation between subjects or values that could be implicit or subjective.

Resuming, the information visualization, through the representation of a dashboard, can be defined by the intersection and combination of visual elements that represent information made available by an entity in a personalized view composed of tiles, graphics, and text. The best way to describe a dashboard is by using the definition used by Stephen Few (2006):

"A dashboard is a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance".

Creating an analogy, we can represent the dashboard as a car, the target audience as its driver, the quality of the information visualization as the car's value tier, the quality and quantity of the data available to be processed as the road, and the destination is represented by the business concepts that the dashboard is oriented to. By focusing on the driver and the satisfaction for its journey, all these concepts must be correctly aligned. It is useless having a top grade vehicle if we don't have roads to venture to or the other way around. Alas, the same can be said to its destination, if it is not clear and precise it can lead to misdirection's and outcomes not desired. The vehicle must be adapted to the road it travels and must get the user to its destination with the most comfort possible. Therefore, it is necessary to abide to certain rules so that the dashboards have its desired effect on the organization that was implemented.

Few (2006) in its book, *Information Dashboard Design,* through the analysis of multiple dashboards, from different business contexts, identified a set of common errors that must be avoided at all costs:

- Exceeding the boundaries of a single screen, fragmenting the data into separate screens or requiring *scrolling* in the visual interface;
- Supplying inadequate context for the data;
- Displaying excessive details or precision;
- Choosing a deficient measure;
- Choosing inappropriate display media;
- Introducing meaningless variety;
- Using poorly designed display media;
- Encoding quantities data inaccurately;
- Arranging the data poorly;
- Highlighting important data ineffectively or not at all;
- Cluttering the display with useless decoration;
- Misusing or overusing colour;
- Designing an unattractive visual display.

To avoid most of this errors it is necessary to do a correct assessment of the topics that are being subject to analysis. For that, we first need to identify the set(s) of scales used in dashboards:

a) Nominal, they don't have any specific order and are don't represent a quantity *per se*. This type of scales are related with data that identify something, such as colleagues or clients names.

b) Ordinal, like the nominal scale this type does not indicate a quantity or volume but there is an order associated to them. The typical examples for this scale relates with rankings, such as "small, medium and large" or ages.

c) Interval, are the categorical scales that have an associated order and can represent a measurable value of a subject. This scale starts out as a quantitative one and then by defining a range for its values it's converted to an Interval Scale.

This is important since, the use of the visual element must be intrinsically related with categorical dimension that is being measured. Line graphs are better used when measuring interval dimensions or to describe the distribution of the graph while bar graphs are best used when the intention is to emphasize individual values, enabling the possibility to better compare values with other contexts. Line graphs should be used to analyze the behaviour of the data; its movements, patterns, trends, cycles.

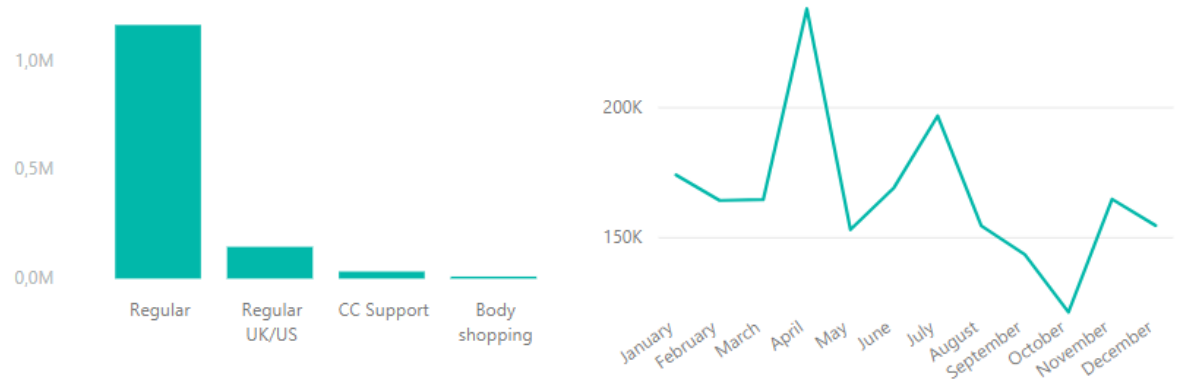


Figure 7 - Line and Bar Chart (charts retrieved from (Microsoft, 2018b))

Stacked bar graphs are the right choice if you want to display the whole and its parts while focusing on the whole.

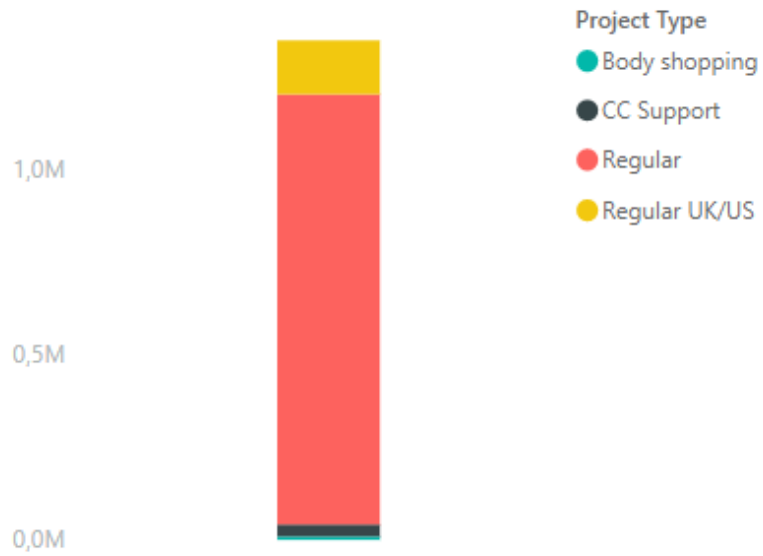


Figure 8 - Stacked Bar chart (charts retrieved from (Microsoft, 2018b))

The combination of bar and line graphs should only be used if we want to compare values with the bar's and use the lines to identify certain trends. Besides adding an extra axis that can hinder the reading, meaning this type of graphics should be used with care.

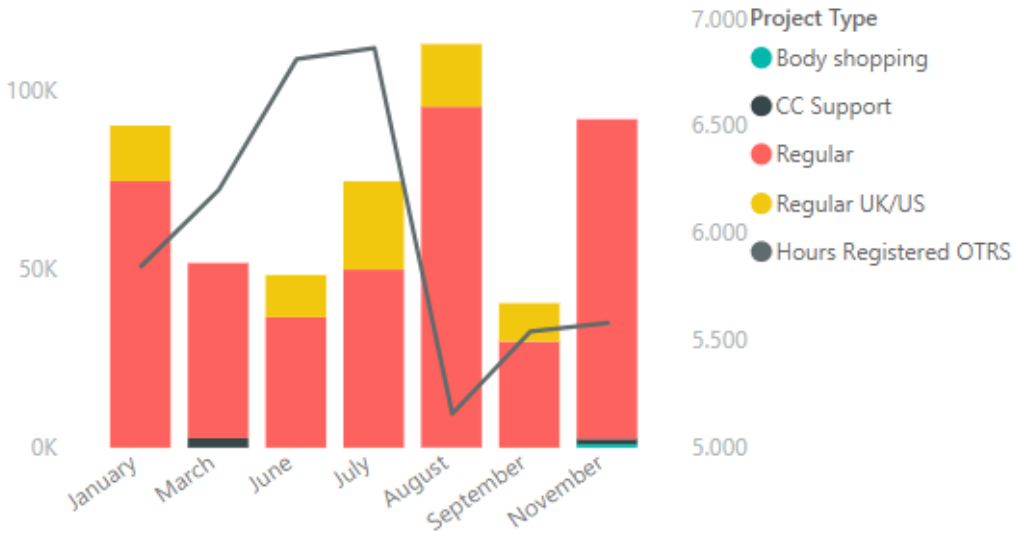


Figure 9 - Bar and Line chart combined (charts retrieved from (Microsoft, 2018b))

Scatter Plot serves to indicate if, and in what direction, two sets of quantitative values are related between them. This object does well its function, yet, it demand some attention from the viewer to interpret the meaning of the values showed.

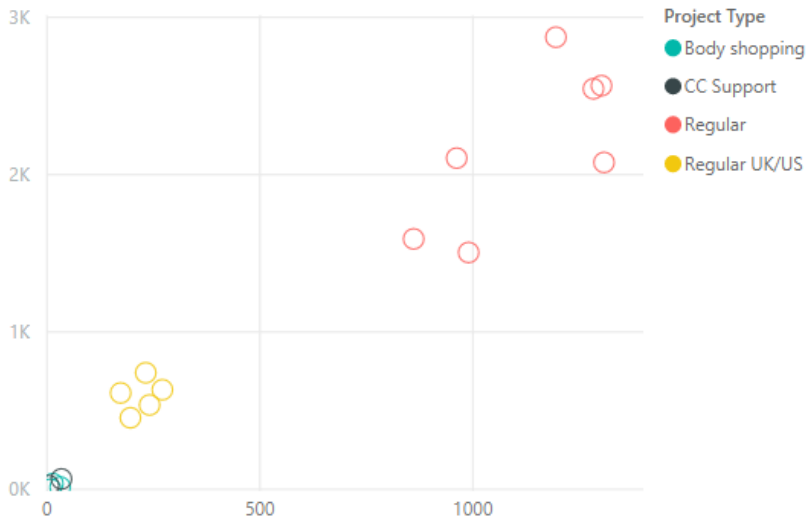


Figure 10 - Scatter Plot chart (charts retrieved from (Microsoft, 2018b))

Treemaps are used to display large sets of hierarchical or categorical data in a space-efficient way, by using all the plot area available to infer analysis.

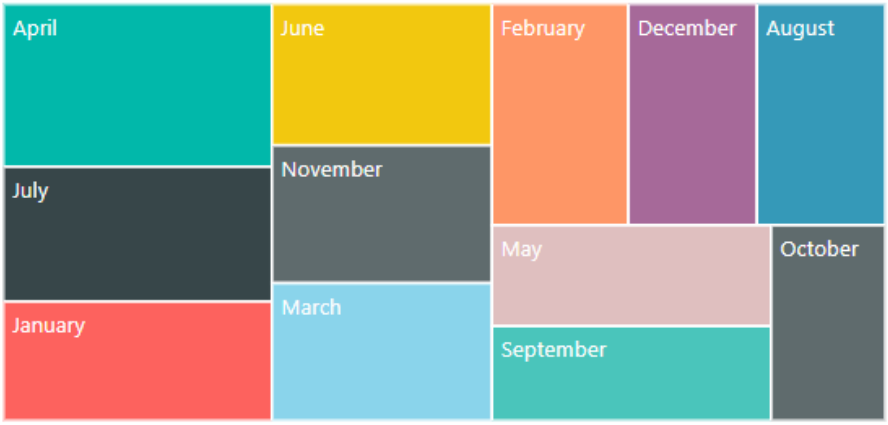


Figure 11 - Treemap chart (charts retrieved from (Microsoft, 2018b))

The tool used to develop dashboards can provide a wider set of visual elements to work, nevertheless, there are elements that should be avoided due to their limitations or that can either occlude data that should be visible to the user. The Pie chart, for example, is an element that has is its usefulness compromised if it has many records or if the records are greatly different between them. The same happens for Area charts.

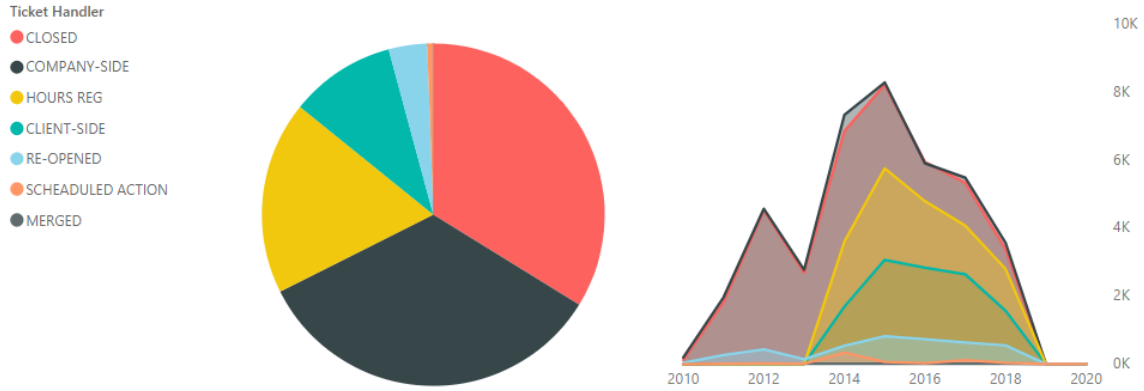


Figure 12 – Pie Chart and Area Chart (charts retrieved from (Microsoft, 2018b))

Besides selecting the correct visual element, for the right objective, when designing dashboards it is considerate good practices following these guidelines:

- Express quantities data at a level of precision that is appropriate to task at hand;
- The element that represents the visualization should be direct, clear and with the minimum form of distraction in the less space possible;
- Maximize the data-area ratio, all the area in a dashboard should be occupied with elements that contain information or add value to the dashboard. Remove unnecessary non-data elements or graphics that only serve as decoration;
- Avoid variations in the contents of the graphics colour if they don't have any real meaning;

- Colours variety should enhance data comprehension and its use should be moderate;
- The use of opposing colours in the same visual element should emphasize comparison;
- Utilize colour saturation to highlight values instead of different colours;
- Avoid the use of graphic borders if there is enough blank space to separate contents;
- Avoid grid lines in tables or graphics unless it is needed to emphasize any specific and expected event;
- Weight of the lines not used to express information should be set to lowest;
- Avoid 3D graphics or elements;
- A direction for the dashboards reading, focusing primarily on the most important topic (high level visualizations to the left and detail to the right and bottom);
- A number of visualizations per dashboard that will not hinder its readability (maximum number of elements);
- Avoid the utilization of elements that simulate real physical objects or merely decorative.

Emphasizing the important data can also be done through the allocation to the correct place in the dashboard. Most authors state that the centre of the dashboard and the top left corner are the place where the dashboard reader will give more attention to the contents and the bottom-right corner as the most unimportant section.

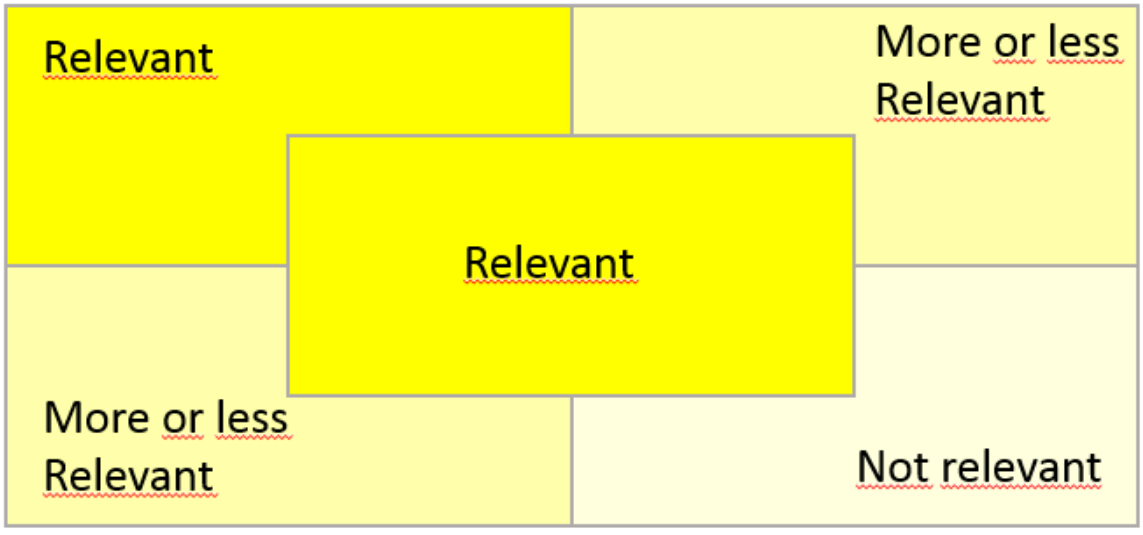


Figure 13 – Dashboard Canvas relevant areas

The best dashboards have an objective-oriented design and tend to answer a question or a set of topics. Not only should the information have a clear intention and be resumed to what really matters to the audience, as it should use the same vocabulary.

2.6. Design Science Research Methodology

In 1991, Nunamaker, Chen, & Purdin presented a *multimethodological* approach to Information Systems research (Fig. 12).

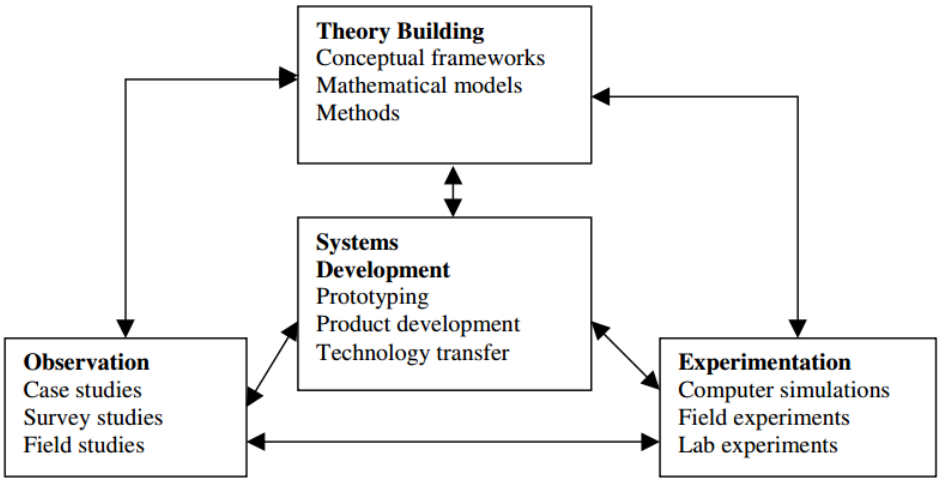


Figure 14 – Nunamaker, Chen, & Purdin approach to IS Research

As the picture describes it, this model is composed by four research strategies that are interconnected and each section contributes to a fuller scientific knowledge of the subject. These strategies are:

- *Theory building* as the section responsible to acquire, develop and aggregate all the scientific knowledge related with the IS research;
- *Experimentation* as the research approach that validates the utility and quality of the systems developed and improving the theories previously defined;
- *Observation* as the approach that reunites all the field studies, case studies and surveys related with the subject to be researched. This approach is useful to get information related to a subject when there isn't much information about it, assisting the experimentation in developing new test cases and identifying theories that could be related with the research object;
- Finally, *Systems development* as the central activity in this multi-methodological framework. This approach is composed by concept design, system architecture conception, product development and technology transfer as the activity responsible to pass the knowledge acquired during the design process to other organizations.

This framework was the early stage of Design Science Research and at the end of this chapter it will be presented how it evolved to current days and the best practices when applying a Design Science Research.

Design Science Research methodology is an Information Technology research methodology specialized to assist and guide in the delivery and development of IT artefacts crucial to solve and understand human problems through the creation of innovative products, models or methods.

This methodology appeared due to the necessity of defining a standard model to resolve transversal hindrances in any IT projects through common guidelines for all the designers. It aims to identify the restraints in the project environment context that could jeopardize the quality of the solution to be delivered. This constraints could be either human related or how the subject to be designed interacts with its surroundings (Hevner & Chatterjee, 2010).

The main goal of this project, defined in the previous chapter, consists on the developing and implementation of top-grade quality dashboards, therefore, it is clear that the goal is to deliver a final product, which in turn must abide to a series of assumptions and solve other problems. The DSR methodology fits in this contexts since it aims to support the IT practitioners and researchers to deliver a feasible solution to the already identified problem and guidelines to evaluate the quality of the IT system delivered (Jarvinen, 2004).

According to March & Smith (1995), DSR is composed by research outputs and research activities. The outputs consists of four types: Constructs, Models, Methods, and Instantiations. While the activities, for the context of this project, are Build and Evaluate. These concepts help to identify the components inherent to any DSR and how they could relate.

The constructs define the syntax used in the context of the project scope, contextualizing the problems and solutions either at a highly formalized level, such as programming language objects descriptions, or informal, which could be used in collaborative tools (e.g.: Slack, a web-browser application which enables users to unify all the information relevant to their organization by simplifying the communications and integrate multiple enterprise products). In this project the main construct would be Business Intelligence, henceforth, other common concepts associated with it would be, for example, Data Warehouses and Dashboards.

A model is a representation of the relationship and interconnectivity between constructs. It can be the technological landscape diagram or simply connects the set of tasks necessarily inherent in the design scope. In the DSR framework the goals of models is to understand the “big picture” and how things are connected and not necessarily a map of the IT hardware landscape or the relationship model in a database.

A method is a particular procedure for accomplishing or approaching something consisting in a series of performed tasks. As the authors clearly define: *“Methods are based on a set of underlying constructs (language) and a representation (model) of the solution space”* (March & Smith, 1995, p. 257). The use of different types of methods, for the same project scope, will alter the constructs and models designed and representation of activities. Its results are clearly defined in the methods used. In the scope of this project we can define the methods, for example, as the set of ETL tasks and the dashboards creation process.

Finally, the instantiation is the equivalent of a test phase of a solution, applying the constructs to the environment they were previously designed. Through instantiations, it is possible to observe and comprehend the practicability and the benefits of the chosen models and methods, acting as an iterative concept of the DSR with the aim to test and improve the artefact.

As it was stated previously, this framework describes how this concepts should be applied, therefore, in DSR every concept must be *built* and *evaluated*. While the *build* activity serves to create something

oriented to the purpose of the DSR, the *evaluate* quantifies the efficiency and quality of the concept delivered.

While the framework presented by March & Smith (1995) enables the designers to identify the components of a project, and the actions needed for each, it lacks the ability to clearly interconnect all the concepts and how to measure the quality of the product of delivered. In 2007, Hevner proposed the identification of three distinct cycles that must be clearly present and defined in any design research project.

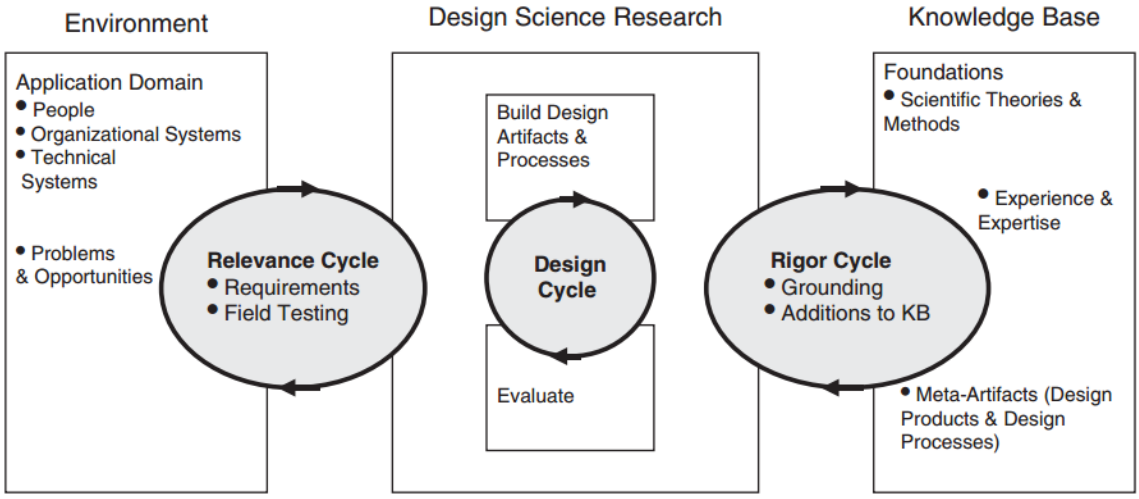


Figure 15 – Hevner (2007) proposed Design Research cycles.

These cycles, has it is depicted in the previous picture, are the *Relevance Cycle*, *Design Cycle* and *Rigor cycle*. These three cycles must be present and clearly identifiable in a design science research project, since through them we are able to:

i. Contextualize the need for the research artefact;

ii. Frame the design activities with the best current practices and technologies;

iii. Evaluate it accordingly to the needs of the organization.

The *Relevance Cycle* domain consists on the identifying all the relevant environments where the design artefact will be inserted and identifying its opportunities and problems, providing the requirements and indicators for the success and usefulness of the project to be developed.

The *Rigor Cycle* contextualizes the project at hands with the all the knowledge, either empirical or theorethical, related with the scope of the research. Through the gathering of information, and the selection of the appropriate method to approach the research object, the research designer ensures the innovation.

The *Design Cycle* is the core of DSR. This cycle produces the results that will interact with the previous cycles, generating a number of interaction between them as many as it is needed to achieve the effectiveness and efficiency desired for the research artefact. While in the *Design Cycle* it is defined a balance between the resources spent on developing the artefact and testing it. In a professional

approach, the tests in this cycle are more related with technical and performance issues while in the *relevance cycle* it is defined how well is the artefact designed for the context to be deployed.

With this cycles defined, Hevner & Chatterjee (2010) elaborated a checklist that guides the design science researchers, and it will be present in the final chapters of this document. The checklist follows the guidelines of DSR and ensures the publications made using this framework are technical enough, in case anyone wants to construct a similar solution, and that the context is understandable to any non-technical person.

The Design Science Research checklist is the following:

1. What is the research question?
2. What is the artefact and how it is represented?
3. What design processes will be used in building the artefact?
4. How are the artefact and the design processes grounded by the knowledge base? What, if any, theories support the artefact design and the design process?
5. What evaluations are performed during the internal design cycles? What design improvements are identified during each design cycle?
6. How is the artefact introduced into the application environment and how is it field tested? What metrics are used to demonstrate artefact utility and improvement over previous artefacts?
7. What new knowledge is added to the knowledge base and in what form (e.g., peer-reviewed literature, meta-artefacts, new theory, and new method)?
8. Has the research question been satisfactorily addressed?

Bearing in mind that "Progress is achieved in design science when existing technologies are replaced by more effective ones"(March & Smith, 1995), the DSR is the chosen methodology for developing the visual analytical solution in this Masters Project.

Alas, in 2004, Vaishnavi and Kuechler presented a set list of processes that composes the Design Science Research and should be followed when applying this methodology.

Table 2 – Design Science Research Methodological Approach

Step	Activity Name	Output
1	Identify Problem and motivation	Performance or result indicators necessary for business analysis in the Costumer Care are made *ad-hoc*.
2	Define objectives for a Solution	Develop a Business Intelligence solution that processes the available data and presents it through a visual interface with pre-defined metrics.
3	Design and Development	Dashboards to be designed under the DSR methodology and through the literature review it will be presented a conceptual model.

4	Demonstration find suitable context	Instantiation of the solution proposed in the organizations context and validate if it answers the problems.
5	Evaluation	Confirm if the dashboards answers all the problems identified or if it is necessary to redesign them.
6	Communication	Is represented by the present document.

3. CONCEPTUAL MODEL PROPOSAL

3.1. DSR METHODOLOGY

The framework used in this project will be de Design Science Research, as it was contextualized in the Literature Review.

While it is observable that the Relevance Cycle is inserted in the Problem Definition and Motivation and Organization Description sections of this document the Design Cycle is explained in the sections that compose the Project Development chapter. Alas, it is only reflected the final product of the design cycle, all the iterations to achieve the final product were done during the development of the practical work of this Masters Project and it will be only presented the justification of the choices.

Finally, the Rigor Cycle is contextualized in the Project Relevance, and Conclusions chapters, where the chosen methods will be contextualized with the authors in the literature review and how the development of this project added value to the scientific community.

Just as it was explained in the DSR literature review, the following sections contain the identification of research outputs identified in this project.

3.1.1. Constructs

The constructs identified in this solution are all the objects and code created during the project development and all the tools that were indirectly used to achieve the project objectives. In this context we can identify:

a. The main repository for the project documents and artifacts;

b. The databases that support the project and inherent SQL code;

c. The tool used to implement the necessary ETL task and its data flow inherent (visible in the Appendix);

d. The dashboards created for the end-users and the tool used to create them.

3.1.2. Methods

The methods are defined as the set of techniques used to achieve one goal, in this solution this will be represented by ETL tasks and the dashboards creation, properly framed through the theoritical background in the second chapter.

3.1.3. Models

For this solution it is presented the conceptual models for the dashboards creative process and the mock-ups' of how dashboards should be developed. It will also be presented the instantiation, or prototype, made for this project.

3.1.3.1. Conceptual Model

To develop a similar project in other organizations that require the same approach it is possible to apply the following model.

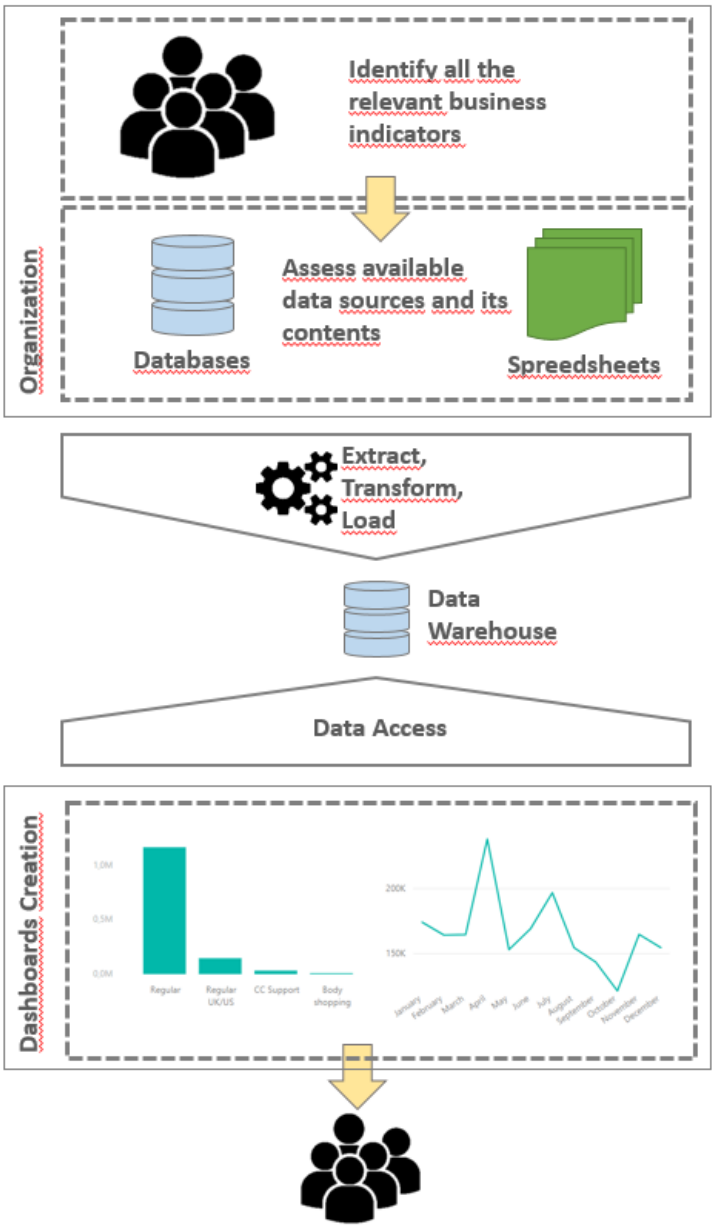


Figure 16 – Project Conceptual model

This model can be described in the following steps:

a) Identification of the subjects to be analysed;

b) Identification and involvement of the stakeholders to pinpoint the necessary indicators;

c) Assessment of the available data related with the subject;

d) Data Extraction, Transformation and Loading;

e) Data warehousing;

f) Data access by dashboards development tool;

g) Dashboards design and development;

h) Dashboards access by the stakeholders.

3.1.3.2. Dashboard Conceptual Model

The dashboards should follow a set of rules unless the contents or objectives of a specific dashboards astray from the common purpose. For most of the situations and through the contents presented in information visualization section it was defined the following standard for the dashboards design.

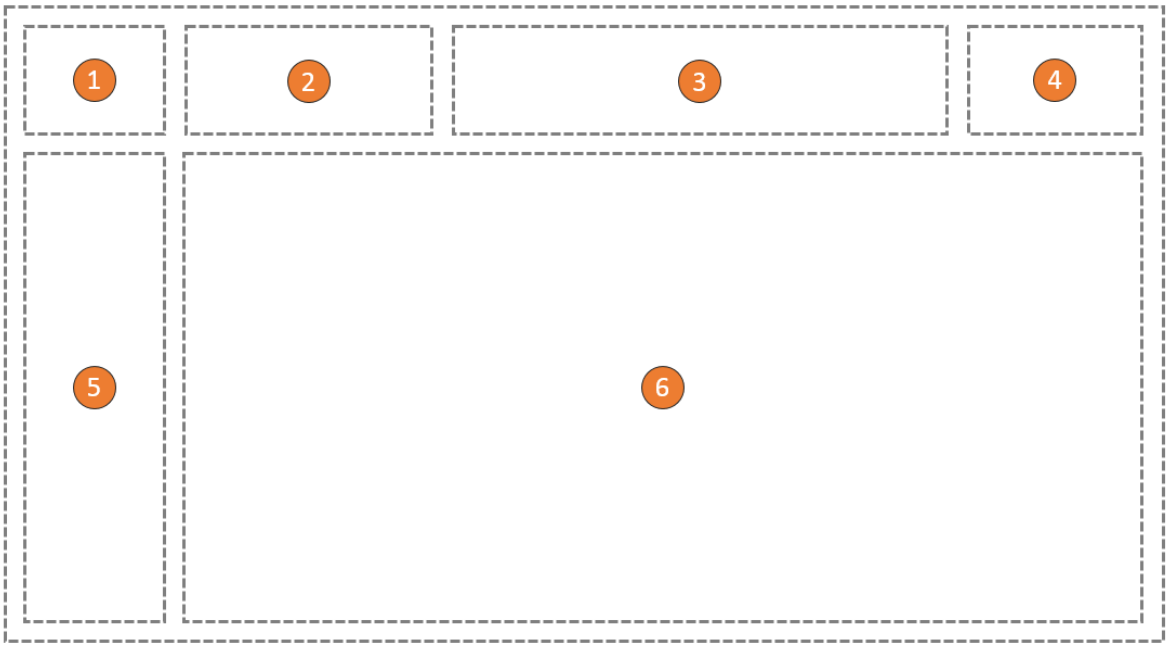


Figure 17 - Dashboard Canvas

The numbers on the dashboard canvas identify different purposes for the areas that will compose a dashboard canvas:

1) Organization logo, the importance of the logo in the dashboard, besides the identification of the organization where the dashboard is destined to, means that we can extract the logo colour(s) as the non-data colour for elements or texts that do not add information value to the dashboard reading;

2) Dashboard identification and objective, trivial element so the reader can identify the subject where the dashboard is related;

3) Filter(s) or Indicator(s) area, depending on the number of elements needed. The element that has the need for more space should use the space associated to the number 5);

4) Timestamp for data load and dashboard refresh date. To ensure all the users are seeing the same information;

5) Filter(s) or Indicator(s) area, depending on the number of elements needed;

6) Information Visualization area. Area that should be composed by the elements (graphs, tables, text) that add real value to the dashboard reading.

3.1.4. Instantiation

Three sequential steps will represent the instantiation. The first step is the deployment of the databases developed explicitly for this project in a productive environment. The second step is to deploy the Visual Studio batches, represented by the ETL tasks, in the previously developed databases. Finally, the last step, is to export the Power BI (Microsoft, 2018b) dashboards to the enterprise Microsoft cloud so users can interact with it.

3.1.4.1. Instantiation Model

After the definition of the constructs and methods we can define the model for the instantiation of this solution as the following figure.

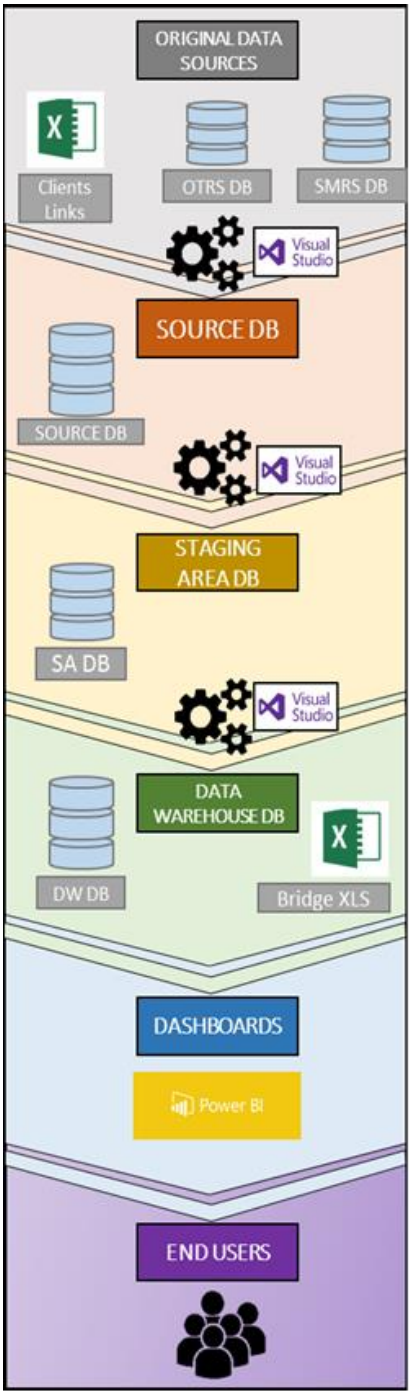


Figure 18 – Project solution prototype

In this model, it is clear which steps precede the current steps and how the constructs and methods are correlates with each other, keeping in mind that this is a solution to be used by people and not just to simplify IT tasks.

3.2. PROJECT MANAGEMENT

3.2.1. Project Phases and Tasks

In order to achieve the objectives defined, it becomes necessary to map all the required tasks for a successful implementation. We can subdivide the tasks into six major phases and with more detail the following activities.

The first phase will be the planning which, regarding methodology, will allow a detailed definition of the activities to be carried out during the whole project as well the resources needed.

The next phase will be the Analysis and Design. This phase will include the requirements mapping, involving Generic Company key-users so that the requirements of all business needs at the reporting level can be documented for the design. In this phase, it will also be analyzed the underlying IT structure of Generic Company to create an optimal BI solution. Only after all the requirements are gathered, will begin the designing of the technologic solution in the appropriated tools. At the end of this phase, a document called "development plan" will be produced, which will explain in detail all the actions and decisions to be executed during the development phase.

The analysis and design will be followed by the development phase, putting to practice the BI solution architecture defined in the previous phase. This phase will be set in a development environment, making sure that there will be no impact in the production environments in the organization.

After all the tests have been carried out successfully, in the development phase, which includes the verification of the results with the exported files in the OTRS application, the production phase will begin. The verification of the results will include the assessment of the KPI´s defined in chapter 2 and the comparison of the results in graphs and tables presented in the visual analytical solution with the values that were calculated in Excel files previously for the same intervals and variables.

Table 3 - Project Phases

Phase	Description
Planning	Detailed planning of all phases of the development of the BI system to be produced
Analysis and Design	Define with the future key-users all the critical metrics needed for the dashboard
	Analysis of the business requirements collected
	Model a Data Warehouse that his ability to contain all the metrics needed for the data analysis
	Define the data transformations necessary for the metrics that are needed
	Choose a visual data analysis software for the Dashboards
Development	Data Warehouse and ETL development
	Extraction of all the information in the data sources related to the OTRS and SMRS applications
	Create the Dashboards with all the pre-defined metrics and calculations
	Define a schedule for the synchronization between the DS and the DW
Testing	Data Warehouse and ETL Testing

Phase	Description
	Manually verify the results of the metrics with the exported files in the OTRS application
Passage to production	Creation or adjustment of new machines to house the BI engine
	Passage to the production environment of the BI solution for the referred machines
	ETL processing
	Configure the application for all the management level collaborators at Generic Company
	Result verification tests

3.2.2. Development plan and Milestones

The next table will present the multiple phases, previously explained, according to the schedule defined for this BI initiative. The green cells represented in the chronogram indicates the months allocated for the phases on the left while the red line indicates the project has reached a milestone.

Table 4 - Project phases chronogram

	Month / Year															
Phases	**Aug.** 2017	**Sep.** 2017	**Oct.** 2017	**Nov.** 2017	**Dec.** 2017	**Jan.** 2018	**Feb.** 2018	**Mar.** 2018	**Apr.** 2018	**May** 2018	**Jun.** 2018	**Jul.** 2018	**Aug.** 2018	**Set.** 2018	**Oct.** 2018	**Nov.** 2018
Planning																
Analysis and design																
Development																
Testing																
Passage to production																
Project Report																
Milestones													**M1**			**M2**

The project's management milestones identified in the previous table indicate the following:

M1 - BI Solution ready to be deployed in Production Environment.

M2 - Delivery of Project Report.

4. DASHBOARD PROTOTYPE DEVELOPMENT

4.1. ENVIRONMENT

4.1.1. Resources

The practical work subjacent to this thesis will be done using a full-stack Microsoft Solution.

The DW will be developed in Microsoft SQL Server 2016 (Microsoft, 2018d); this database will house every information starting with the data sources (Pipe, 1997). The data sources already identified for this project reside in different environments, the OTRS is a MySQL Database while the data from the SMRS application is stored in a Microsoft SQL Server 2014 Database.

The ETL steps will be done using Microsoft SQL Server Data Tools for Visual Studio 2013 (Microsoft, 2018e) since it is a tool with a sufficient range of functionality for this purpose and all the routine jobs for the data integration will be deployed using Microsoft Server Integration Services 2016.

Finally, the data analysis will be done using a visual analytical application using the DW information. The application chosen for that effect is Microsoft PowerBI due to its compatibility performances and since it is an application that has been increasingly used among BI developers (Gartner, 2017).

Additionally, it will be used Microsoft Excel for some manual mapping that has to be done.

4.1.2. Organization

As it was described in the first chapter, the organization where this project is to be inserted uses OTRS for Incident Management, where it reports the time taken for each action/analysis done in an Incident and Replicon to allocate the time used during the work hours to a specific project.

The reports made with the data from both data sources are done in Excel PowerPivot after manually extracting pre-made static reports from the applications.

4.2. DATABASE DESIGN

It is necessary to refer that the sources identified that will be used for this project only contains structured data, meaning that the information is already in organized containers which allows a more natural data extraction and transformation to the future DW (Phillips-Wren, Iyer, Kulkarni, & Ariyachandra, 2015).

Since MySQL uses a different SQL language than that used on SQL Server databases, this projects project will house three different databases:

- Source DB: where it will be replicated the OTRS and SMRS necessary tables and allows a common language for the next phase

- Staging Area DB: where most data transformations will occur under the Visual Studio application
- Data Warehouse DB: where the treated and clean information will be deposited

4.2.1. Source Database

4.2.1.1. Database Diagram

For this project we opted to load all the source data from the OTRS Database and SMRS Database into a single Database, having the same data structure as the following figure.

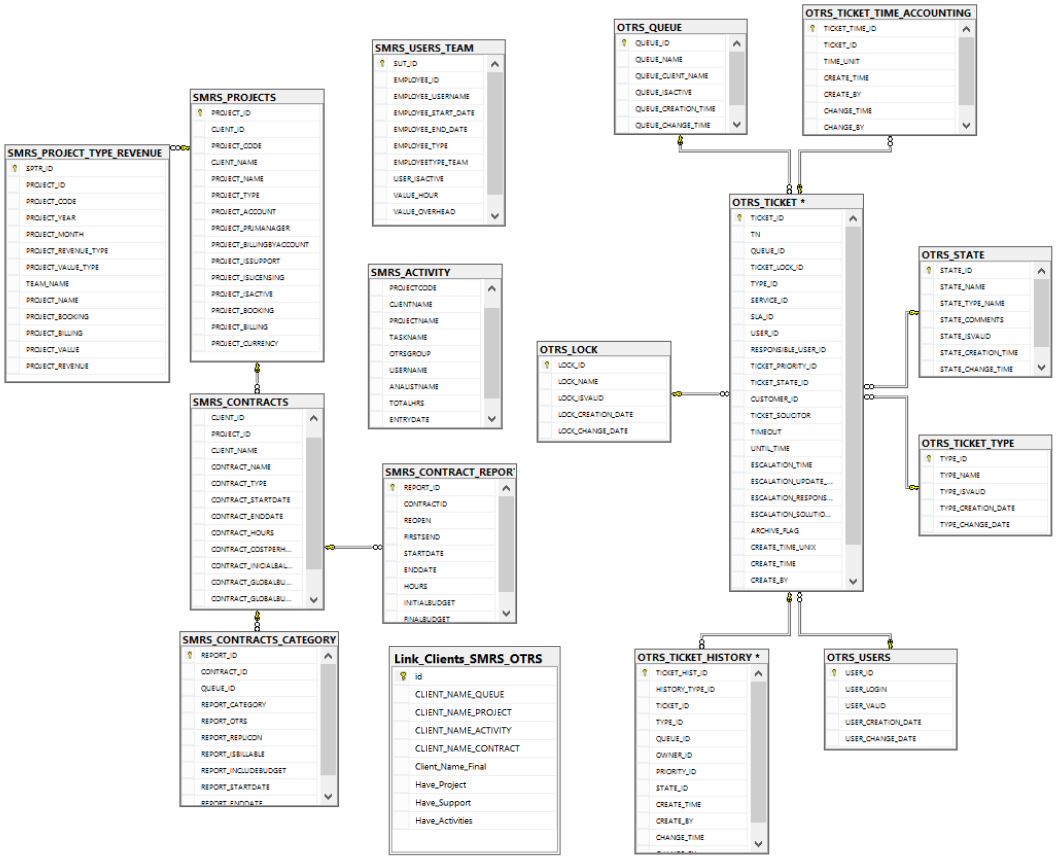


Figure 19 - Source Database Diagram

4.2.1.2. Tables Description

The data available in this database is loaded in the following tables, and it uses the prefix OTRS or SMRS to indicate from which data source the information came.

Table 5 – Source table OTRS_TICKET

Table Name	OTRS_TICKET	
Table Description	The main table of the OTRS table where we have the current status of the ticket to analyze. Will be used to connecting to other components	
Column Name	**Column Data Type**	**Comments**
TICKET_ID	INTEGER	
TN	INTEGER	
QUEUE_ID	INTEGER	
TYPE_ID	INTEGER	
SERVICE_ID	INTEGER	It will not be used
SLA_ID	INTEGER	
USER_ID	INTEGER	
RESPONSIBLE_USER_ID	INTEGER	It will not be used
TICKET_PRIORITY_ID	INTEGER	
TICKET_STATE_ID	INTEGER	
CUSTOMER_ID	NVARCHAR	
TICKET_SOLICITOR	NVARCHAR	
TIMEOUT	INTEGER	It will not be used
UNTIL_TIME	INTEGER	It will not be used
ESCALATION_TIME	INTEGER	It will not be used
ESCALATION_UPDATE_TIME	INTEGER	It will not be used
ESCALATION_RESPONSE_TIME	INTEGER	It will not be used
ESCALATION_SOLUTION_TIME	INTEGER	It will not be used
ARCHIVE_FLAG	INTEGER	It will not be used
CREATE_TIME	DATETIME	
CREATE_TIME_UNIX	INTEGER	It will not be used
CREATE_BY	INTEGER	
CHANGE_TIME	DATETIME	
CHANGE_BY	INTEGER	

Table 6 – Source table OTRS_STATE

Table Name	OTRS_STATE	
Table Description	The table that identifies the possible states that a ticket could be defined. This table is crucial to identify when a ticket must be handled by an employee or to be responded by the client	
Column Name	**Column Data Type**	**Comments**
STATE_ID	INTEGER	
STATE_NAME	NVARCHAR	
STATE_TYPE_NAME	NVARCHAR	
STATE_COMMENTS	NVARCHAR	
STATE_ISVALID	INTEGER	
STATE_CREATION_TIME	DATETIME	

STATE_CHANGE_TIME	DATETIME	

Table 7 – Source table OTRS_TICKET_HISTORY

Table Name	OTRS_TICKET_HISTORY	
Table Description	The table that registers all the events regarding a Ticket. This table will be utilized to identify the course a ticket run up and how much time it took to complete a set of events.	
Column Name	**Column Data Type**	**Comments**
TICKET_HIST_ID	INTEGER	
HISTORY_TYPE_ID	INTEGER	
TYPE_ID	INTEGER	
QUEUE_ID	INTEGER	
OWNER_ID	INTEGER	
PRIORITY_ID	INTEGER	
STATE_ID	INTEGER	
CREATE_TIME	DATETIME	
CREATE_BY	INTEGER	
CHANGE_TIME	DATETIME	
CREATE_BY	INTEGER	

Table 8 – Source table OTRS_QUEUE

Table Name	OTRS_QUEUE	
Table Description	The table in OTRS that identifies the Queues created and assigned to a specific client. The different queues should help to differentiate the projects, contracts or subjects associated with a client.	
Column Name	**Column Data Type**	**Comments**
QUEUE_ID	INTEGER	
QUEUE_NAME	NVARCHAR	
QUEUE_CLIENT_NAME	NVARCHAR	
QUEUE_ISACTIVE	INTEGER	
QUEUE_CREATION_TIME	DATETIME	
QUEUE_CHANGE_TIME	DATETIME	

Table 9 – Source table OTRS_TICKET_TYPE

Table Name	OTRS_TICKET_TYPE	
Table Description	Identifies the type of ticked that is defined. By default, a ticket is defined as "default" or "incident."	
Column Name	**Column Data Type**	**Comments**
TYPE_ID	INTEGER	
TYPE_NAME	NVARCHAR	
TYPE_ISVALID	INTEGER	

TYPE_CREATION_DATE	DATETIME	
TYPE_CHANGE_DATE	DATETIME	

Table 10 – Source table SMRS_ACTIVITY

Table Name	SMRS_ACTIVITY	
Table Description	The table that registers all the hours a collaborator allocated to a project.	
Column Name	**Column Data Type**	**Comments**
ACTIVITY_ID	INTEGER	
PROJECTCODE	NVARCHAR	
CLIENTNAME	NVARCHAR	
PROJECTNAME	NVARCHAR	
TASKNAME	NVARCHAR	
OTRSGROUP	NVARCHAR	
USERNAME	NVARCHAR	
ANALISTNAME	NVARCHAR	
TOTALHRS	NUMERIC(18,2)	
ENTRYDATE	DATETIME	

Table 11 – Source table SMRS_CONTRACTS_CATEGORY

Table Name	SMRS_CONTRACTS_CATEGORY	
Table Description	The table that links the reports with OTRS queues and Contracts.	
Column Name	**Column Data Type**	**Comments**
REPORT_ID	INTEGER	
CONTRACT_ID	INTEGER	
QUEUE_ID	INTEGER	
REPORT_CATEGORY	NVARCHAR	
REPORT_OTRS	NVARCHAR	
REPORT_REPLICON	NVARCHAR	
REPORT_ISBILLABLE	NVARCHAR	
REPORT_INCLUDEBUDGET	NVARCHAR	
REPORT_STARTDATE	DATETIME	
REPORT_ENDDATE	DATETIME	

Table 12 – Source table SMRS_USERS_TEAM

Table Name	SMRS_USERS_TEAM	
Table Description	The table with all the users registered in SMRS, it will be merged to create a full employee table in the DW	
Column Name	**Column Data Type**	**Comments**

SUT_ID	INTEGER	
EMPLOYEE_ID	INTEGER	
EMPLOYEE_USERNAME	NVARCHAR	
EMPLOYEE_START_DATE	DATETIME	
EMPLOYEE_END_DATE	DATETIME	
EMPLOYEE_TYPE	NVARCHAR	
EMPLOYEETYPE_TEAM	NVARCHAR	
USER_ISACTIVE	NVARCHAR	
VALUE_HOUR	NUMERIC(18,2)	
VALUE_OVERHEAD	NUMERIC(18,2)	
CURRENCY_TYPE	NVARCHAR	

Table 13 – Source table SMRS_CONTRACT_REPORTS

Table Name	SMRS_CONTRACT_REPORTS	
Table Description	The table that identifies the reports if associated with a contract. The data will be inserted in the DW to enable future analysis	
Column Name	**Column Data Type**	**Comments**
REPORT_ID	INTEGER	
CONTRACT_ID	INTEGER	
REOPEN	NVARCHAR	
FIRSTSEND	NVARCHAR	It will not be utilized
STARTDATE	DATETIME	
ENDDATE	DATETIME	
HOURS	NUMERIC(18,2)	
INITIALBUDGET	NUMERIC(18,2)	
FINALBUDGET	NUMERIC(18,2)	

Table 14 – Source table SMRS_PROJECT_TYPE_REVENUE

Table Name	SMRS_PROJECT_TYPE_REVENUE	
Table Description	The table that identifies the reports if associated with a contract. The data will be inserted in the DW to enable future analysis	
Column Name	**Column Data Type**	**Comments**
PROJECT_ID	INTEGER	
PROJECT_CODE	INTEGER	
PROJECT_YEAR	INTEGER	
PROJECT_MONTH	INTEGER	
PROJECT_REVENUE_TYPE	NVARCHAR	
PROJECT_VALUE_TYPE	NVARCHAR	
TEAM_NAME	NVARCHAR	
PROJECT_NAME	NVARCHAR	
PROJECT_BOOKING	NUMERIC(18,2)	

PROJECT_BILLING	NUMERIC(18,2)	
PROJECT_VALUE	NUMERIC(18,2)	
PROJECT_REVENUE	NUMERIC(18,2)	

Table 15 – Source table OTRS_LOCK

Table Name	OTRS_LOCK	
Table Description	The table that initially migrated to be used in the Dashboard but it does not have any real value, so it was dropped out.	
Column Name	**Column Data Type**	**Comments**
LOCK_ID	INTEGER	It will not be utilized
LOCK_NAME	NVARCHAR	It will not be utilized
LOCK_ISVALID	INTEGER	It will not be utilized
LOCK_CREATION_DATE	DATETIME	It will not be utilized
LOCK_CHANGE_DATE	DATETIME	It will not be utilized

Table 16 – Source table OTRS_TICKET_TIME_ACCOUNTING

Table Name	OTRS_TICKET_TIME_ACCOUNTING	
Table Description	The table that identifies the time that every employee assigned to a ticket.	
Column Name	**Column Data Type**	**Comments**
TICKET_TIME_ID	INTEGER	
TICKET_ID	INTEGER	
TIME_UNIT	NUMERIC(18,2)	
CREATE_TIME	DATETIME	
CREATE_BY	INTEGER	
CHANGE_TIME	DATETIME	
CHANGE_BY	INTEGER	

Table 17 – Source table OTRS_USERS

Table Name	OTRS_USERS	
Table Description	The table that contains all the list of users registered in OTRS. It will be merged with the data in an SMRS table to contain all the possible existing employee data	
Column Name	**Column Data Type**	**Comments**
USER_ID	INTEGER	
USER_LOGIN	NVARCHAR	
USER_VALID	INTEGER	
USER_CREATION_DATE	DATETIME	
USER_CHANGE_DATE	DATETIME	

Table 18 – Source table SMRS_CONTRACTS

Table Name	SMRS_CONTRACTS	
Table Description	Identifies the contracts that are associated with a Project in Replicon	
Column Name	**Column Data Type**	**Comments**
CONTRACT_ID	INTEGER	
PROJECT_ID	INTEGER	
CLIENT_NAME	NVARCHAR	
CONTRACT_NAME	NVARCHAR	
CONTRACT_TYPE	NVARCHAR	
CONTRACT_STARTDATE	DATETIME	
CONTRACT_ENDDATE	DATETIME	
CONTRACT_HOURS	NUMERIC(18,2)	
CONTRACT_COSTPERHOUR	NUMERIC(18,2)	
CONTRACT_INICIALBALANCE	NUMERIC(18,2)	
CONTRACT_GLOBALBUDGET	NUMERIC(18,2)	
CONTRACT_GLOBALBUDGETREVENUE	NUMERIC(18,2)	
CONTRACT_ROUNDHOUR	NUMERIC(18,2)	
CONTRACT_CURRENCY	NVARCHAR	

Table 19 – Source table SMRS_PROJECTS

Table Name	SMRS_PROJECTS	
Table Description	Identifies the projects that are registered in Replicon and associate them to a client	
Column Name	**Column Data Type**	**Comments**
PROJECT_ID	INTEGER	
CLIENT_ID	INTEGER	
PROJECT_CODE	NVARCHAR	
CLIENT_NAME	NVARCHAR	
PROJECT_NAME	NVARCHAR	
PROJECT_TYPE	NVARCHAR	
PROJECT_ACCOUNT	NVARCHAR	
PROJECT_PRJMANAGER	NVARCHAR	
PROJECT_BILLINGBYACCOUNT	NVARCHAR	
PROJECT_ISSUPORT	NVARCHAR	
PROJECT_ISLICENSING	NVARCHAR	
PROJECT_ISACTIVE	NVARCHAR	
PROJECT_BOOKING	NUMERIC(18,2)	
PROJECT_BILLING	NUMERIC(18,2)	
PROJECT_CURRENTY	NUMERIC(18,2)	

Table 20 – Source table Link_Clients_SMRS_OTRS

Table Name	Link_Clients_SMRS_OTRS	
Table Description	The table created specifically to house the manual correlations of the client's name in the different data sources.	
Column Name	**Column Data Type**	**Comments**
id	INTEGER	
CLIENT_NAME_QUEUE	NVARCHAR	It has data in case we have an OTRS Queue associated with this client
CLIENTE_NAME_PROJECT	NVARCHAR	It has data in case we have a Project associated with this client
CLIENT_NAME_ACTIVITY	NVARCHAR	It has data in case we have registered activities in the client
CLIENT_NAME_CONTRACT	NVARCHAR	It has data in case we have a Contract associated with this client
Client_Name_Final	NVARCHAR	The name to be used in the DW
Have_Project	INTEGER	
Have_Support	INTEGER	
Have_Activities	INTEGER	

4.2.2. Staging Area Database

The relational model used in this solution will be the Galaxy Schema, using two Star Schemas with some shared dimensions.

The use of the Star Schemas brings advantages regarding simplification of future queries since in this type of dimensional model, there is a fact table surrounded by several dimension tables, thus reducing the number of tables and relationships in the database.

This type of model allows any user to easily retrieve the desired information from the database (Moody & Kortink, 2000).

4.2.2.1. Entity Classification

The first step to defining our star schemas is to classify the tables in the Data Source and how they will define the future Data Warehouse. Applying the concepts in the Literature review, it is necessary to classify a table, defining it as *Transactional, Component* or *Classification.*

In the following figure we classify the tables using the data source scheme presented in the **Source Database** chapter.

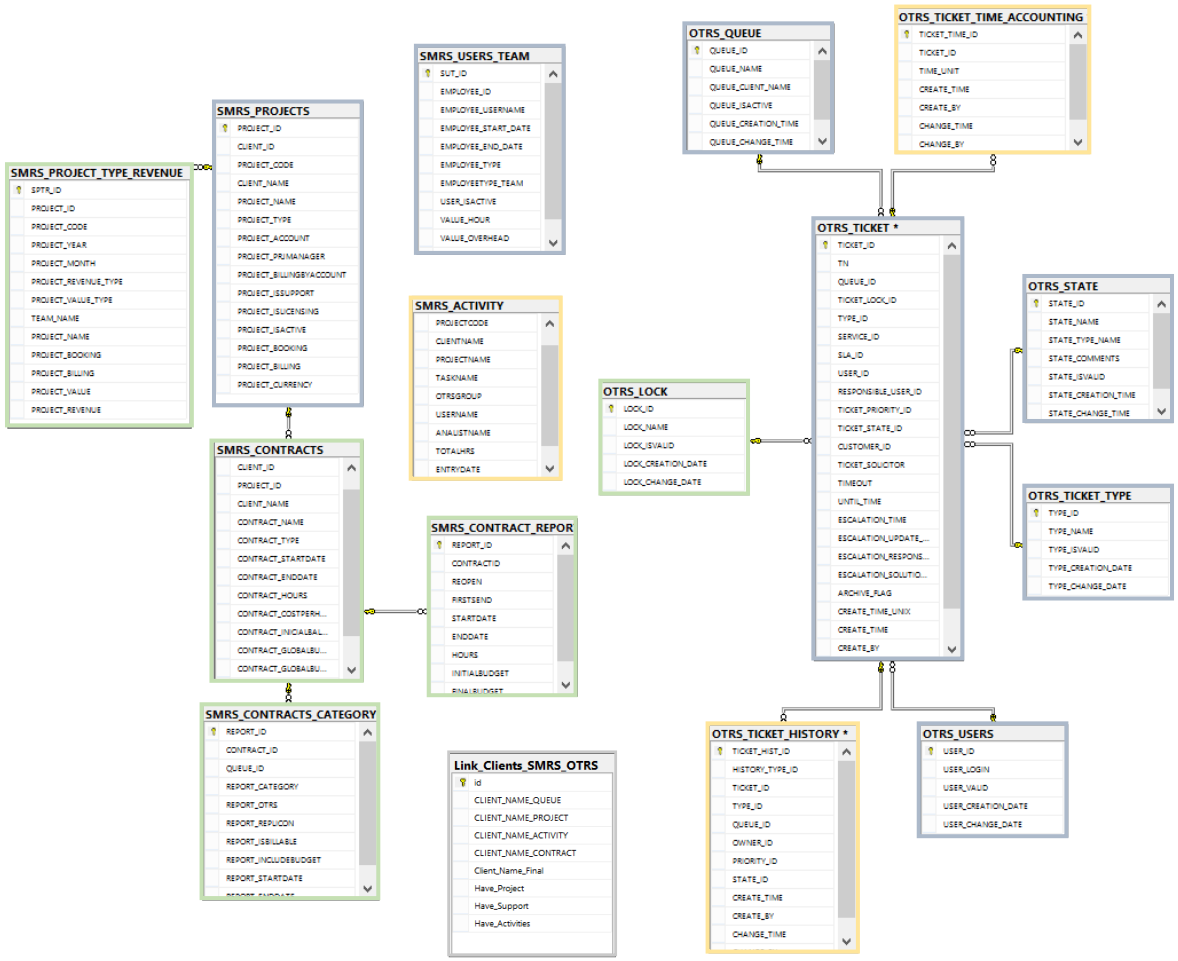


Figure 20 - Source Database Entity Classification

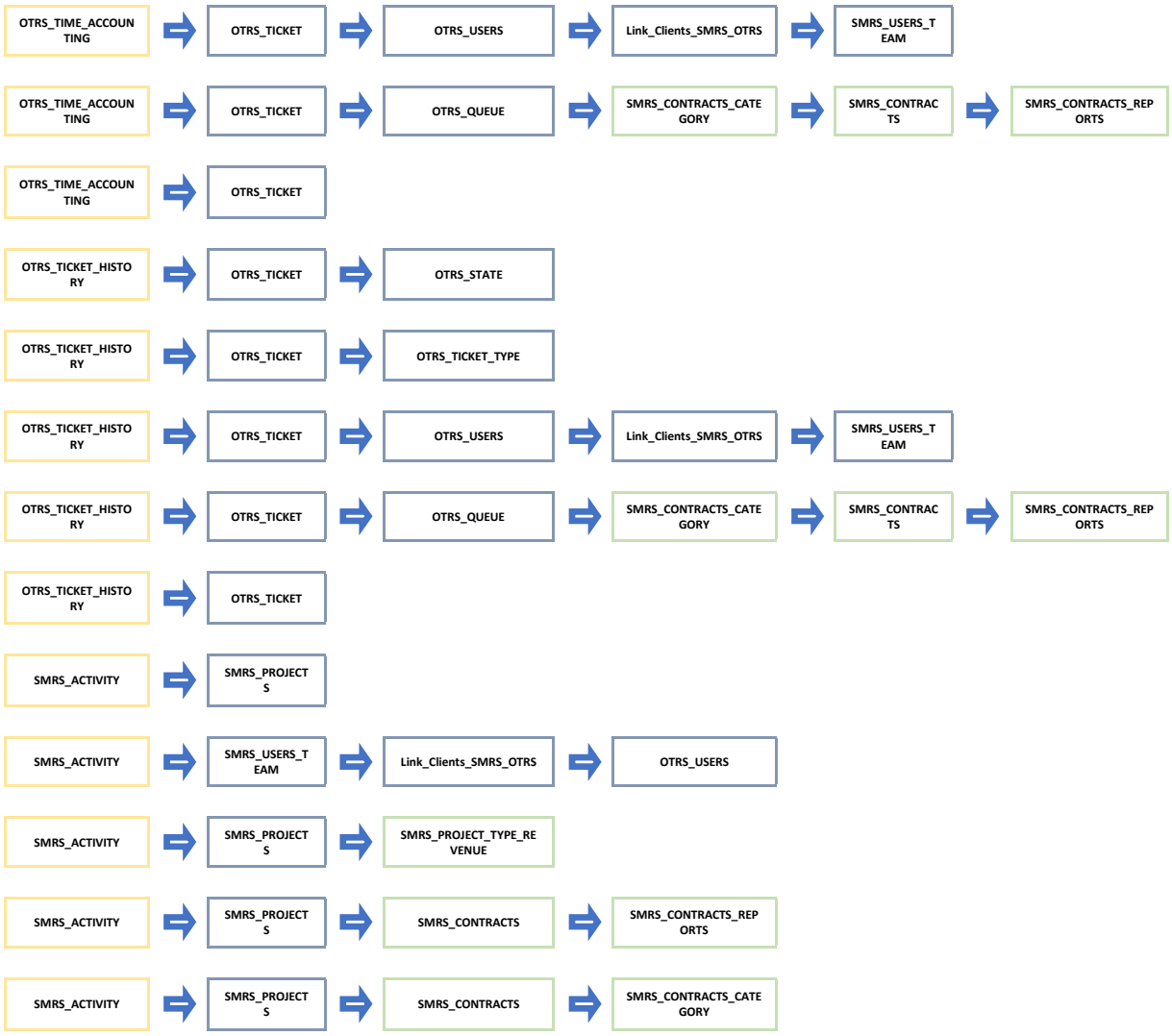
Legend:

- Transactional Entity
- Component Entity
- Classification Entity

4.2.2.2. Hierarchy Identification

When assessing the hierarchy of the entities, we are defining the dimensions that will be present in the DW, and how do they relate with the factual table.

During this assessment, it was defined the following hierarchies:



Legend:

- Transactional Entity
- Component Entity
- Classification Entity

4.2.2.3. Database Diagram

The final model of the Staging Area database can be observed in the following figure.

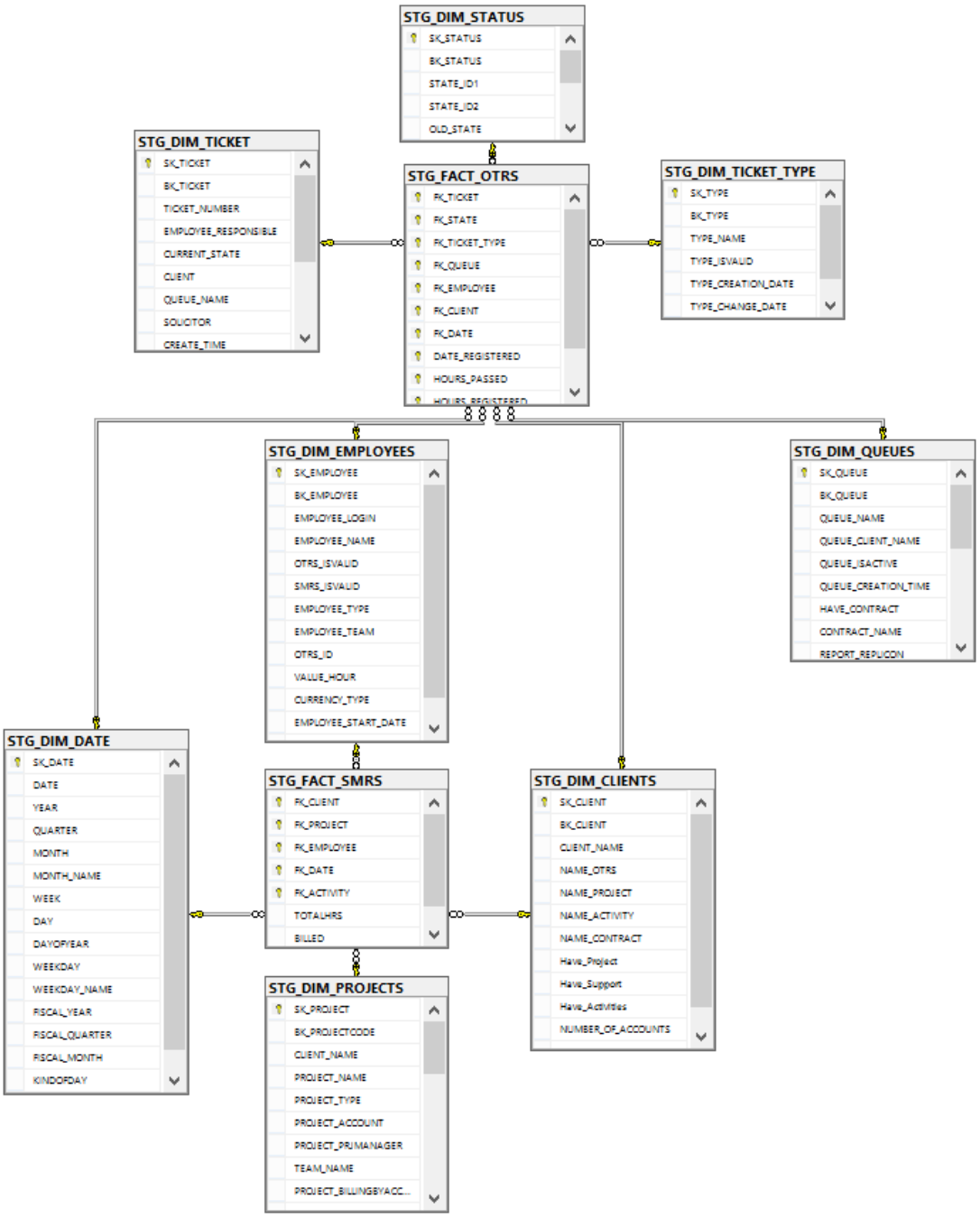


Figure 21 - Staging Area Diagram

Since the tables in the Data Warehouse database are similar in structure and context as the Staging Area tables, the Tables descriptions will be visible in the next chapter.

4.2.3. Data Warehouse Database

The Data Warehouse database represents the final state of the data collection we want to analyze. In this chapter, it is represented how the tables relate between themselves and what type of information they contain.

4.2.3.1. Database Diagram

As it was stated previously, the model of the DW is similar to the SA. The differences that can be observed in both databases is the dropping of the business keys columns and other columns used to map values.

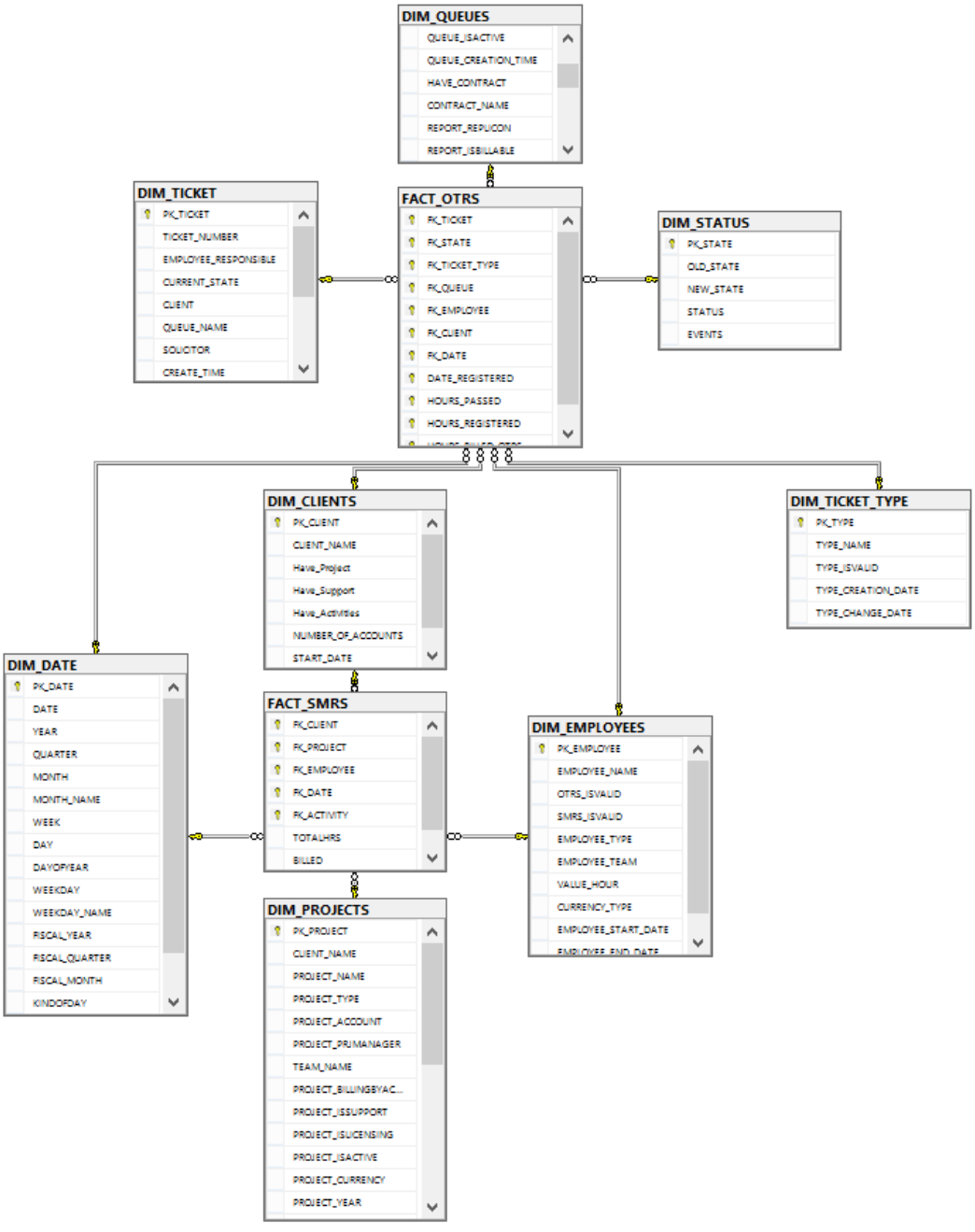


Figure 22 – Data Warehouse Diagram

4.2.3.2. Tables description

In this section, it will be presented the tables that compose the DW and the data that each contains.

Table 21 – DW table DIM_CLIENTS

Table Name	DIM_CLIENTS	
Table Description	Dimension Table that identifies all the clients registered in the Data Sources, after the mapping in the SA it is only presented the final Client name	
Column Name	**Column Data Type**	**Comments**
PK_CLIENT	INTEGER	
CLIENT_NAME	NVARCHAR	
Have_Project	NVARCHAR	
Have_Support	NVARCHAR	
Have_Activities	NVARCHAR	
NUMBER_OF_ACCOUNTS	INTEGER	
START_DATE	DATETIME	

Table 22 – DW table DIM_DATE

Table Name	DIM_DATE	
Table Description	Dimensional Date directly uploaded from the Staging Area. All the dates are calculated automatically and do not use a relationship with the dates registered in the Facts tables.	
Column Name	**Column Data Type**	**Comments**
PK_DATE	INTEGER	
DATE	DATETIME	
YEAR	INTEGER	
QUARTER	INTEGER	
MONTH	INTEGER	
MONTH_NAME	NVARCHAR	
WEEK	DATETIME	
DAY	INTEGER	
DAYOFYEAR	INTEGER	
WEEKDAY	INTEGER	
WEEKDAY_NAME	NVARCHAR	
FISCAL_YEAR	INTEGER	
FISCAL_QUARTER	INTEGER	
FISCAL_MONTH	INTEGER	
KINDOFDAY	VARCHAR	
DESCRIPTION	VARCHAR	

Table 23 – DW table DIM_QUEUES

Table Name	DIM_QUEUES
Table Description	The dimensional table that identifies the queues in OTRS and eventual reports and contracts associated with them. The excessive data this dimensional table has enables future analysis.

Column Name	Column Data Type	Comments
PK_QUEUE	INTEGER	
QUEUE_NAME	NVARCHAR	
QUEUE_CLIENT_NAME	NVARCHAR	
QUEUE_ISACTIVE	NVARCHAR	
QUEUE_CREATION_TIME	DATETIME	
HAVE_CONTRACT	NVARCHAR	
CONTRACT_NAME	NVARCHAR	
REPORT_REPLICON	NVARCHAR	
REPORT_ISBILLABLE	NVARCHAR	
REPORT_INCLUDEBUDGET	NVARCHAR	
REPORT_START_DATE	DATETIME	
REPORT_END_DATE	DATETIME	
REPORTED_HOURS	NUMERIC(18,2)	
INITIALBUDGET	NUMERIC(18,2)	
FINALBUDGET	NUMERIC(18,2)	
CONTRACT_COSTPERHOUR	NUMERIC(18,2)	
CONTRACT_ROUNDHOUR	NUMERIC(18,2)	
CONTRACT_CURRENCY	NVARCHAR	

Table 24 – DW table DIM_TICKET_TYPE

Table Name	DIM_TICKET_TYPE	
Table Description	The dimensional table that identified the tickets type associated with an OTRS Ticket	
Column Name	**Column Data Type**	**Comments**
PK_TYPE	INTEGER	
TYPE_NAME	NVARCHAR	
TYPE_ISVALID	NVARCHAR	
TYPE_CREATION_DATE	DATETIME	
TYPE_CHANGE_DATE	DATETIME	

Table 25 – DW table DIM_EMPLOYEES

Table Name	DIM_EMPLOYEES	
Table Description	Dimension Table that identifies all the employees registered in the Data Sources, after the mapping in the SA. Some fields are not null to enable some calculations in the Fact tables	
Column Name	**Column Data Type**	**Comments**
PK_EMPLOYEE	INTEGER	
EMPLOYEE_NAME	NVARCHAR	
OTRS_ISVALID	NVARCHAR	
SMRS_ISVALID	NVARCHAR	
EMPLOYEE_TYPE	NVARCHAR	

EMPLOYEE_TEAM	NVARCHAR	
VALUE_HOUR	NUMERIC(18,2)	Cost of the Employee per hour
CURRENCY_TYPE	NVARCHAR	
EMPLOYEE_START_DATE	DATETIME	
EMPLOYEE_END_DATE	DATETIME	

Table 26 – DW table DIM_STATUS

Table Name	DIM_STATUS	
Table Description	The dimensional table used to identify the passage of states in a ticket to assess if it is Client-side or Company-side time correctly.	
Column Name	**Column Data Type**	**Comments**
PK_STATE	INTEGER	
OLD_STATE	NVARCHAR	
NEW_STATE	NVARCHAR	
STATUS	NVARCHAR	
EVENTS	NVARCHAR	

Table 27 – DW table DIM_PROJECTS

Table Name	DIM_PROJECTS	
Table Description	The dimensional table that includes all the identified data related to a project. The same project will be repeated according to the months it was opened so it could be mapped to the Factual table.	
Column Name	**Column Data Type**	**Comments**
PK_PROJECT	INTEGER	
CLIENT_NAME	NVARCHAR	
PROJECT_NAME	NVARCHAR	
PROJECT_TYPE	NVARCHAR	
PROJECT_ACCOUNT	NVARCHAR	
PROJECT_PRJMANAGER	NVARCHAR	
TEAM_NAME	NVARCHAR	
PROJECT_BILLINGBYACCOUNT	NVARCHAR	
PROJECT_ISSUPORT	NVARCHAR	
PROJECT_ISLICENSING	NVARCHAR	
PROJECT_ISACTIVE	NVARCHAR	
PROJECT_CURRENCY	NVARCHAR	
PROJECT_YEAR	INTEGER	
PROJECT_MONTH	INTEGER	
PROJECT_DATE_ID	INTEGER	Column created so it could be mapped in Power BI
ACTIVE_DATE_START	DATETIME	

ACTIVE_DATE_END	DATETIME	
PROJECT_REVENUE_TYPE	NVARCHAR	
PROJECT_VALUE_TYPE	NVARCHAR	
PROJECT_BOOKING	NUMERIC(18,2)	
PROJECT_BILLING	NUMERIC(18,2)	
PROJECT_VALUE	NUMERIC(18,2)	
PROJECT_REVENUE	NUMERIC(18,2)	
HAVE_CONTRACT	NVARCHAR	
CONTRACT_COSTPERHOUR	NUMERIC(18,2)	
CONTRACT_ROUNDHOUR	NUMERIC(18,2)	
CONTRACT_CURRENCY	NVARCHAR	

Table 28 – DW table DIM_TICKET

Table Name	DIM_TICKET	
Table Description	The dimensional table that contains every registered ticket at OTRS. While the factual table registers all the events recorded in a ticket this table shows the current state of the incident for quick tracking.	
Column Name	**Column Data Type**	**Comments**
PK_TICKET	INTEGER	
TICKET_NUMBER	NVARCHAR	
EMPLOYEE_RESPONSIBLE	NVARCHAR	
CURRENT_STATE	NVARCHAR	
CLIENT	NVARCHAR	
QUEUE_NAME	NVARCHAR	
SOLICITOR	NVARCHAR	
CREATE_TIME	DATETIME	
LAST_CHANGE_TIME	DATETIME	
TICKET_FIRST_RESPONSE_DATE	DATETIME	
TICKET_FIRST_RESPONSE_TIME	NUMERIC(18,2)	
TICKET_CLOSE_DATE	DATETIME	
TICKET_CLOSE_TIME	NUMERIC(18,2)	

Table 29 – DW table FACT_OTRS

Table Name	FACT_OTRS	
Table Description	Fact table that contains all the events related to a ticket. The time it took that event to pass and the time an employee registered for the solving.	
Column Name	**Column Data Type**	**Comments**
FK_TICKET	INTEGER	
FK_STATE	INTEGER	
FK_TICKET_TYPE	INTEGER	
FK_QUEUE	INTEGER	

FK_EMPLOYEE	INTEGER	
FK_DATE	INTEGER	
DATE_REGISTERED	DATETIME	
HOURS_PASSED	NUMERIC(18,2)	
HOURS_REGISTERED	NUMERIC(18,2)	
HOURS_BILLED_OTRS	NUMERIC(18,2)	

Table 30 – DW table FACT_SMRS

Table Name	FACT_SMRS	
Table Description	Fact table that contains the time registered by an employee for their activities at a Replicon project during the day	
Column Name	**Column Data Type**	**Comments**
FK_CLIENT	INTEGER	
FK_PROJECT	INTEGER	
FK_EMPLOYEE	INTEGER	
FK_DATE	INTEGER	
FK_ACTIVITY	INTEGER	
TOTALHRS	NUMERIC(18,2)	
BILLED	NUMERIC(18,2)	

4.3. DATA MANIPULATION – EXTRACTION, TRANSFORMATION AND LOADING

In this chapter, it will be explained the logic behind the *Control Flow's* flux in Visual Studio.

Due to a necessity of loading performance and error avoidance, the transformations were done using SQL manipulation. Both the SQL Selects as well as the Data Loads in Visual Studio can be seen in the Appendix.

4.3.1. Designed SQL Functions

It is important to refer that some functions were created in this DB to enhance future analysis.

While some functions are available in other SQL development platforms, it was necessary to create this functions in the databases for the desired transformations.

4.3.1.1. COMPUTUS

Although the function in itself requires little code, it is used when populating the STG_DIM_DATE.

Function Name	COMPUTUS
Description	Computus is the calculation of the date of Easter in the Christian calendar.
SQL Code	

```
CREATE FUNCTION [dbo].[COMPUTUS]
(
   @Y INT -- THE YEAR WE ARE CALCULATING EASTER SUNDAY FOR
)
RETURNS DATETIME
AS
BEGIN
DECLARE
@A INT,
@B INT,
@C INT,
@D INT,
@E INT,
@F INT,
@G INT,
@H INT,
@I INT,
@K INT,
@L INT,
@M INT

SET @A = @Y % 19
SET @B = @Y / 100
SET @C = @Y % 100
SET @D = @B / 4
SET @E = @B % 4
SET @F = (@B + 8) / 25
SET @G = (@B - @F + 1) / 3
SET @H = (19 * @A + @B - @D - @G + 15) % 30
SET @I = @C / 4
SET @K = @C % 4
SET @L = (32 + 2 * @E + 2 * @I - @H - @K) % 7
SET @M = (@A + 11 * @H + 22 * @L) / 451
RETURN(DATEADD(MONTH, ((@H + @L - 7 * @M + 114) / 31)-1, CAST(CAST(@Y AS VARCHAR) AS DATETIME))
+ ((@H + @L - 7 * @M + 114) % 31))
END
GO
```

Figure 23 - Designed function COMPUTUS

4.3.1.2. UFN_CALC_TIME

One of the critical functions of this project, the UFN_CALC_TIME. This function enables to only consider the working time in the difference of two *datetime* values.

Through this function, it is possible to correctly assess many tickets metrics in the dashboards without adding the time that wouldn't be correct to the evaluation.

Function Name	UFN_CALC_TIME
Description	A function that sums the working time passed between two dates (from 9 AM to 6 PM only on workdays).
SQL Code	

```
CREATE FUNCTION [dbo].[UFN_CALC_TIME]
(
 -- Add the parameters for the function here
  @START_DATE    DATETIME
    ,@END_DATE     DATETIME
)
RETURNS INT
AS
BEGIN

   DECLARE @TIME INT = 0
   DECLARE @SAMEDAY INT = 0
   DECLARE @DIFERENTDAY INT = 0

   SET @START_DATE = CAST(CAST(@START_DATE AS DATE) AS DATETIME) + CAST((SELECT CASE WHEN
CAST(@START_DATE AS TIME) >= '18:00:00' THEN '18:00:00' ELSE CAST(@START_DATE AS TIME) END) AS
DATETIME)
   SET @START_DATE = CAST(CAST(@START_DATE AS DATE) AS DATETIME) + CAST((SELECT CASE WHEN
CAST(@START_DATE AS TIME) <= '09:00:00' THEN '09:00:00' ELSE CAST(@START_DATE AS TIME) END) AS
DATETIME)

   SET @END_DATE = CAST(CAST(@END_DATE AS DATE) AS DATETIME) + CAST((SELECT CASE WHEN
CAST(@END_DATE AS TIME) >= '18:00:00' THEN '18:00:00' ELSE CAST(@END_DATE AS TIME) END) AS
DATETIME)
   SET @END_DATE = CAST(CAST(@END_DATE AS DATE) AS DATETIME) + CAST((SELECT CASE WHEN
CAST(@END_DATE AS TIME) <= '09:00:00' THEN '09:00:00' ELSE CAST(@END_DATE AS TIME) END) AS
DATETIME)

   SET @SAMEDAY = CASE WHEN CAST(CONVERT(VARCHAR(8),@START_DATE,112) AS INT) =
CAST(CONVERT(VARCHAR(8),@END_DATE,112) AS INT) THEN 1 ELSE 0 END
   SET @DIFERENTDAY = CASE WHEN CAST(CONVERT(VARCHAR(8),@START_DATE,112) AS INT) <>
CAST(CONVERT(VARCHAR(8),@END_DATE,112) AS INT) THEN 1 ELSE 0 END

   SET @TIME = @TIME + DATEDIFF(SECOND,@START_DATE, CAST(CAST(@START_DATE AS DATE) AS
DATETIME) + '18:00:00') * (SELECT CASE WHEN KINDOFDAY = 'WORKDAY' THEN 1 ELSE 0 END
                                                        FROM [Thesis_SA].[dbo].[STG_DIM_DATE] D
                                                        WHERE D.SK_DATE =
CAST(CONVERT(VARCHAR(8),@START_DATE,112) AS INT)) * @DIFERENTDAY

   SET @TIME = @TIME + DATEDIFF(SECOND, CAST(CAST(@END_DATE AS DATE) AS DATETIME) +
'09:00:00',@END_DATE) * (SELECT CASE WHEN KINDOFDAY = 'WORKDAY' THEN 1 ELSE 0 END
                                                            FROM [Thesis_SA].[dbo].[STG_DIM_DATE]
D
                                                        WHERE D.SK_DATE =
CAST(CONVERT(VARCHAR(8),@END_DATE,112) AS INT)) * @DIFERENTDAY

   SET @TIME = @TIME + DATEDIFF(SECOND, @START_DATE,@END_DATE) * @SAMEDAY

   SET @TIME = @TIME +      (SELECT COUNT(DISTINCT SK_DATE) * 9 * 3600        AS NUMBER_OF_SECONDS
                     FROM [Thesis_SA].[dbo].[STG_DIM_DATE] D
                     WHERE 1=1
                       AND D.SK_DATE > CAST(CONVERT(VARCHAR(8),@START_DATE,112) AS INT) AND
D.SK_DATE < CAST(CONVERT(VARCHAR(8),@END_DATE,112) AS INT)
                       AND D.KINDOFDAY = 'WORKDAY')
--    SELECT @TIME / 3600.00
   RETURN @TIME

END
GO
```

Figure 24 - Designed function UFN_CALC_TIME

As it is possible to observe in the code, the function itself only works if the date dimension is already created and populated in the database.

4.3.1.3. InitCap

Function mainly used to normalize clients and collaborators name in the database.

Function Name	InitCap
Description	Transforms the first letter in a word in a capital letter.

SQL Code

```
CREATE FUNCTION [dbo].[InitCap] (@inStr VARCHAR(8000))
 RETURNS VARCHAR(8000)
 AS
 BEGIN
  DECLARE @outStr VARCHAR(8000) = LOWER(@inStr),
 @char CHAR(1),
 @alphanum BIT = 0,
 @len INT = LEN(@inStr),
          @pos INT = 1;

  -- Iterate through all characters in the input string
  WHILE @pos <= @len BEGIN

   -- Get the next character
   SET @char = SUBSTRING(@inStr, @pos, 1);

   -- If the position is first, or the previous characater is not alphanumeric
   -- convert the current character to upper case
   IF @pos = 1 OR @alphanum = 0
    SET @outStr = STUFF(@outStr, @pos, 1, UPPER(@char));

   SET @pos = @pos + 1;

   -- Define if the current character is non-alphanumeric
   IF ASCII(@char) <= 47 OR (ASCII(@char) BETWEEN 58 AND 64) OR
 (ASCII(@char) BETWEEN 91 AND 96) OR (ASCII(@char) BETWEEN 123 AND 126)
 SET @alphanum = 0;
   ELSE
 SET @alphanum = 1;

  END

 RETURN @outStr;
 END
GO
```

Figure 25 - Designed function InitCap

4.3.1.4. INSTR

Combined with the *InitCap* to normalize values and remove undesired characters in a string.

Function Name	INSTR
Description	Returns the position of a string inside a string.

SQL Code

```
CREATE FUNCTION [dbo].[INSTR] (@str VARCHAR(8000), @substr VARCHAR(255), @start INT, @occurrence
INT)
 RETURNS INT
 AS
 BEGIN
DECLARE @found INT = @occurrence,
  @pos INT = @start;

WHILE 1=1
BEGIN
 -- Find the next occurrence
 SET @pos = CHARINDEX(@substr, @str, @pos);

 -- Nothing found
 IF @pos IS NULL OR @pos = 0
  RETURN @pos;

 -- The required occurrence found
 IF @found = 1
  BREAK;

 -- Prepare to find another one occurrence
 SET @found = @found - 1;
 SET @pos = @pos + 1;
END

RETURN @pos;
 END
GO
```

Figure 26 - Designed function INSTR

4.3.2. Source ETL

After analyzing the OTRS and SMRS original databases, it was possible to conclude that there were not too many records that obligated to define an Extraction model that would look up the date of the last value loaded in the local Source Database and dates posteriors to that in the originals database.

As it is visible in the **Appendix A**, it was adopted a model where all the tables in the Source Database were truncated (remove all the available data in a table, without deleting it, and committing the action, making impossible to retrieve information previously available) first and then the loaded with all the available information in the original databases.

Between each set of actions, in the Visual Studio defined as *Container*, were created small logs to keep track of the concluded actions.

4.3.3. STG ETL

More than 70% of all the ETL tasks undergone in this project, adding final data transformations done in the Dashboard Design phase, were defined in the Staging Area.

Using the *Visual Studio BI Data Tools,* it was defined the loading scheme from the Source Database to the SA database and the necessary transformations to load the information successfully. The code used for that end can be seen in the **SA ETL** section in the Appendix.

Using the same logic defined in the Source ETL, all the current data in the tables of the Staging Area Database are truncated, being the Factual table truncated first and Dimensional table after due to foreign keys restrictions.

Between each set of actions, were created small logs to keep track of the concluded actions.

4.3.4. DW ETL

Finally, the DW ETL is the last package created in the Visual Studio to be run once the solution is deployed. It loads all the data in the SA DB to the DW DB without the additional mappings or keys that are used to distinguish dimensional values or facts.

Between each set of actions, were created small logs to keep track of the concluded actions.

4.4. DASHBOARD DESIGN

After years of rigid dashboard design, organizations are finally adopting tools that enable their collaborators to create and explore data in the way they best see fit.

Traditional dashboard design focuses almost exclusively on defining the right success metrics, then piecing together some charts and gauges on a single page. The way these dashboards appears to their general public can be confusing and give very little insight to unaware users that don't know what they are searching for. Today's dashboards have the power to do more and be more than their predecessors. While they still seek to have the right information for the right audience, they must evolve (Juice, 2015).

In this section it will be explained the primary goal of this Masters Project, to develop up to date dashboards using the most recent techniques while captivating the target audience attention, presenting rigorous metrics and relate facts between each page.

4.4.1. Data Model

When loading the information in the DW to the Microsoft Power BI, we can notice that the same tables that are present in the DW are also loaded in the BI application.

In this dashboard application, we also introduce a new concept that is the "Bridge" tables. These tables are used to relate concepts that share common attributes. For example, the "Client Name" column in the DIM_TICKET table is different in the DIM_CLIENT since the same client can be registered with different names in both data sources.

When creating a bridge table, we are creating the possibility to assume the same filter, for the same concept, in all of the dashboards. As it is observable in the following figure, we have two bridge tables, one for clients and other for employees.

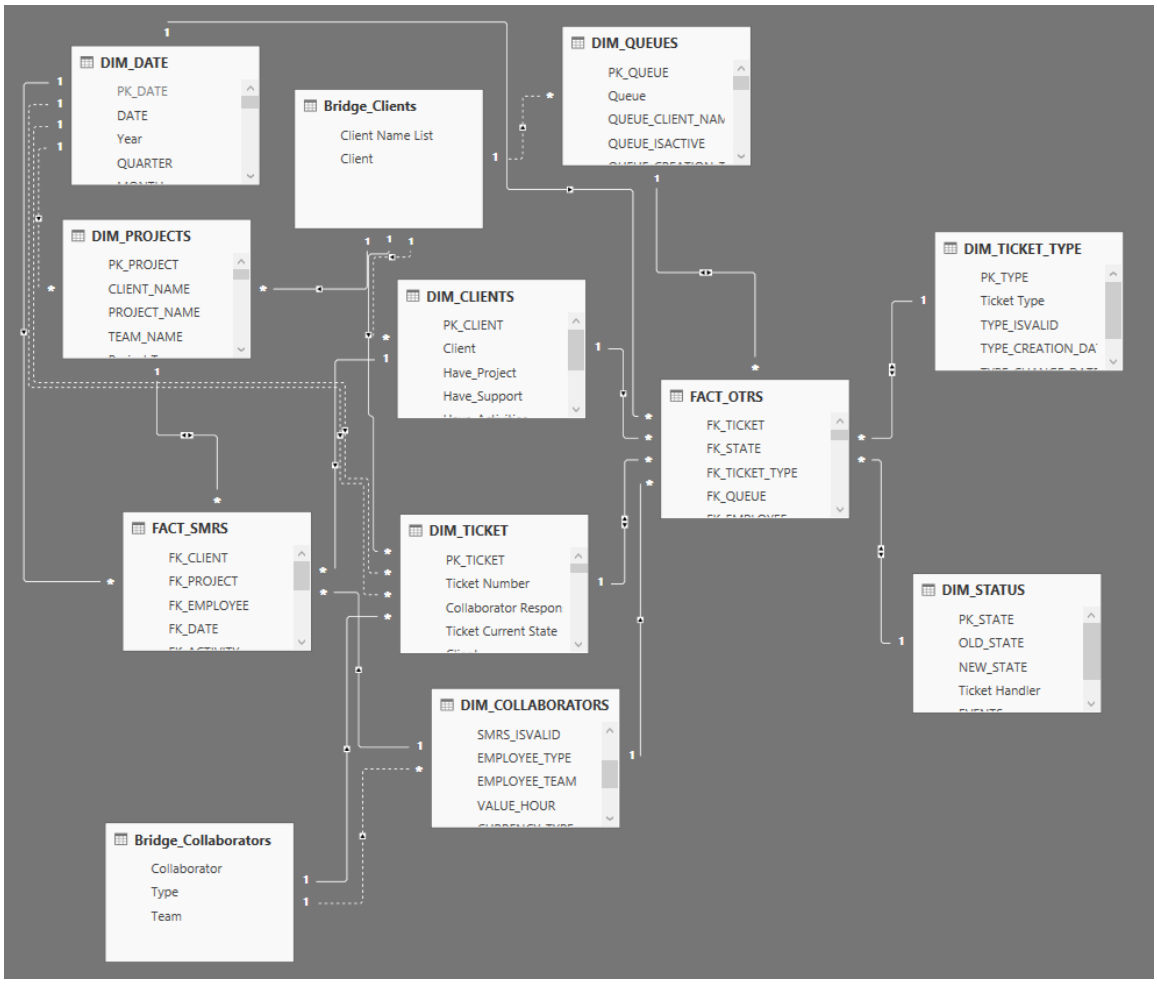


Figure 27 – PowerBI Data Model

Since PowerBI is a self-serving BI tool, the table name DIM_EMPLOYEES was changed to DIM_COLLABORATORS in case these dashboards were open to further alterations, and to not hurt susceptibilities.

Additionally, for time-analysis functions, PowerBI demands that the primary date column to be defined. It was defined DIM_DATE.DATE as the primary date column for this project.

4.4.2. ETL *on-the-fly*

While loading data to the dashboard application tool, PowerBI also enables users to additional ETL actions on the application. This type of ETL is commonly now as ETL *on-the-fly.*

In theory, it is always better to do all the ETL tasks before the Dashboards designs, for performance issues and solution design method, but since BI applications can have different operating methods for the same purpose, it may be necessary to tune our data to the application needs.

In this project, the ETL on-the-fly is primarily represented by:

a) Loading two additional excel sheets that map the clients and collaborators between dimensional tables and enables additional calculation functions, previously explained as "Bridge Tables."

b) Adding columns to the tables to make calculated metrics (to be analyzed in the next section) easier and add value to the solution

c) Minor twitches like datatype alterations, Columns ordering, and Date formats.

4.4.3. Measures

The measures in PowerBI have calculated metrics that enable the BI developers to create data analysis solutions by performing calculations in the data as the end-user interacts with the reports. In PowerBI the language underlying the measures is denominated by DAX (Data Analysis eXpressions), and most of the functions available are already shared with Microsoft Excel.

Each measure is associated to the table where it was created and relates with columns on it. It is possible to relate the data available in other tables to the measure, alas, it is necessary to be cautious on how the data model is defined since one dimension can filter the other, but the other way around may not be possible.

While in most data models it is normal that the vast majority of the metrics are defined in the factual table, in this project we have a high number of metrics created in the DIM_TICKET since it made calculation easier for this solution objectives.

The identification of the measures in the dashboards must also follow a global standard for what they mean and measure, such as:

- CY – Current Year;
- PY – Previous Year;
- YTD – Year-to-Date;
- AVG – Average.

4.4.4. Report

A report is a composition of dashboards that are related to the same subject and using the same dataset. The visualizations in a report, represented by dashboards, are focused in a subject of the theme to be analyzed. A report can be defined by a single visualization or pages full of visualizations.

It is important to refer that there are common settings in the dashboards that compose this report. In the top left screen of the dashboard, we have the company logo followed by the text identifying the dashboard subject and the problem it proposes to answer. In top right corner we have the timestamp of the most recent data integration and the last refresh date, enabling the user to confirm with his peers if the same subset of data is being analyzed. The reports can be seen at the **Appendix B**.

4.4.4.1. Operational Dashboard

The Operational Dashboard is designed to 1st and 2nd level Support to track incidents and to grant those teams a more dynamic solution to visualize the Incidents that require a company side action. This Dashboard will also be used by customer support management tier to follow-up specific incidents or just to follow-up global operations.

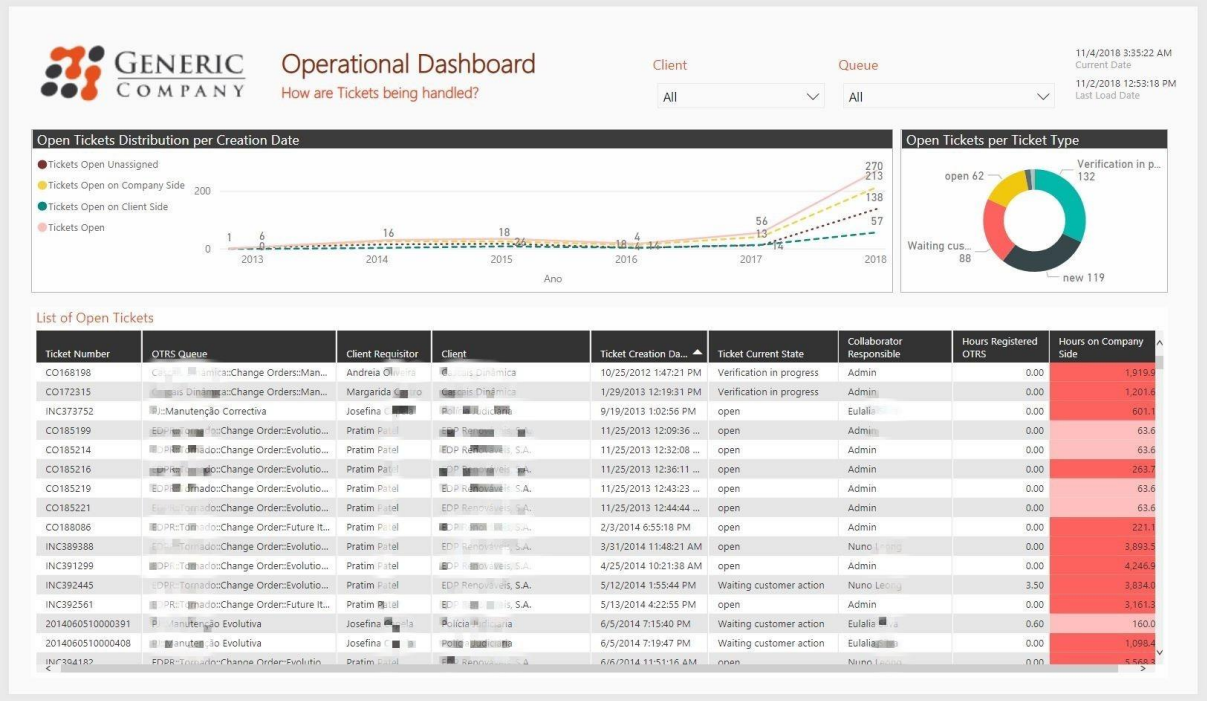


Figure 28 – Operational Dashboard

Due to the dynamic properties of PowerBI, when selecting any attribute in the widgets that compose the dashboard it is not necessary to add many filters to the screen. The *Client* and incident *Queue* are sufficient to filter the incidents in the area we want to analyse and selecting one of the measures presented in the graph (*Tickets Unassigned* or *Ticket Awaiting for Collaborator Response*) enables additional filtering.

Since we only have three types of ticket, the Donut chart was the selected visualization for this dimension, allowing an easy visualization without compromising data interpretation.

In the lower half of this dashboard, we present a table that contains only the Incidents that are not closed, ordered by the most recent creation date. This table has two additional facts; hours already registered in this incident and hours passed on company-side. The last measure has a color gradient to indicate the time this request is taking to be resolved.

4.4.4.2. Customer Support Dashboard

The Customer Support Dashboard is designed for a management level tier to analyze the evolution of the operations that are registered in OTRS.

As previously stated, OTRS is mainly used for Costumer Support operations, alas, it is possible to associate projects or change requests on existing applications. With that logic, it was defined three major filters for this dashboard: the client to analyze, the queues associated with the client, and the collaborator.

By adding the collaborator dimension to this dashboard, we can comprehend the evolution or importance of one person to the client or project we want to scrutinize.

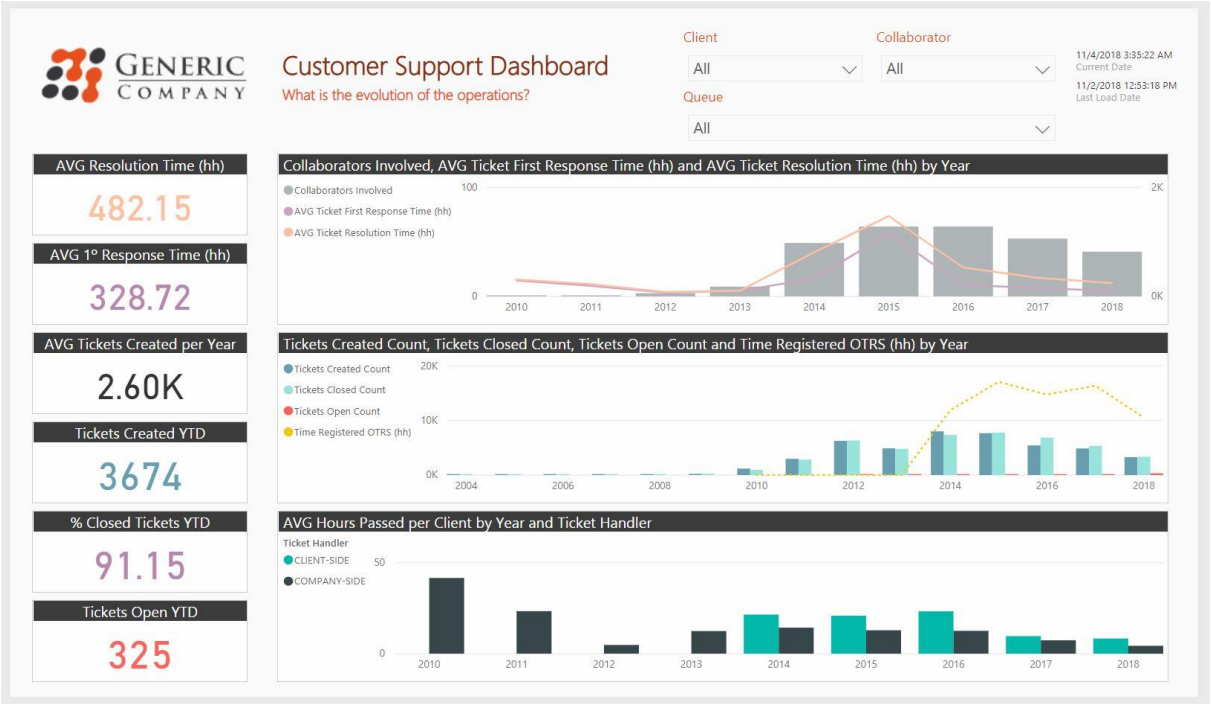


Figure 29 – Customer Support Dashboard

All the graphics used in this dashboard are time-related, using the year primarily on their axis (with the possibility to go to month or days), so the reader of this dashboard could focus more on the evolution over time. Static information about the dimensions presented are presented in the next dashboards.

The discrepancy between the years presented in the top chart to the other two below is due to the migration of previous tickets from another incident management system to OTRS. Due to that migration, we have requests registered from past time, but we do not have the same set of data as OTRS makes available.

4.4.4.3. Clients Dashboard

The client's Dashboard, as the name indicates, is related to global operations associated with a client and the collaborators that performed actions with the selected entity.

This is the first dashboard that uses both the SMRS and OTRS fact table. From the SMRS data, it is possible to infer the hours that a user registered for a project under a client and the profit associated with it.

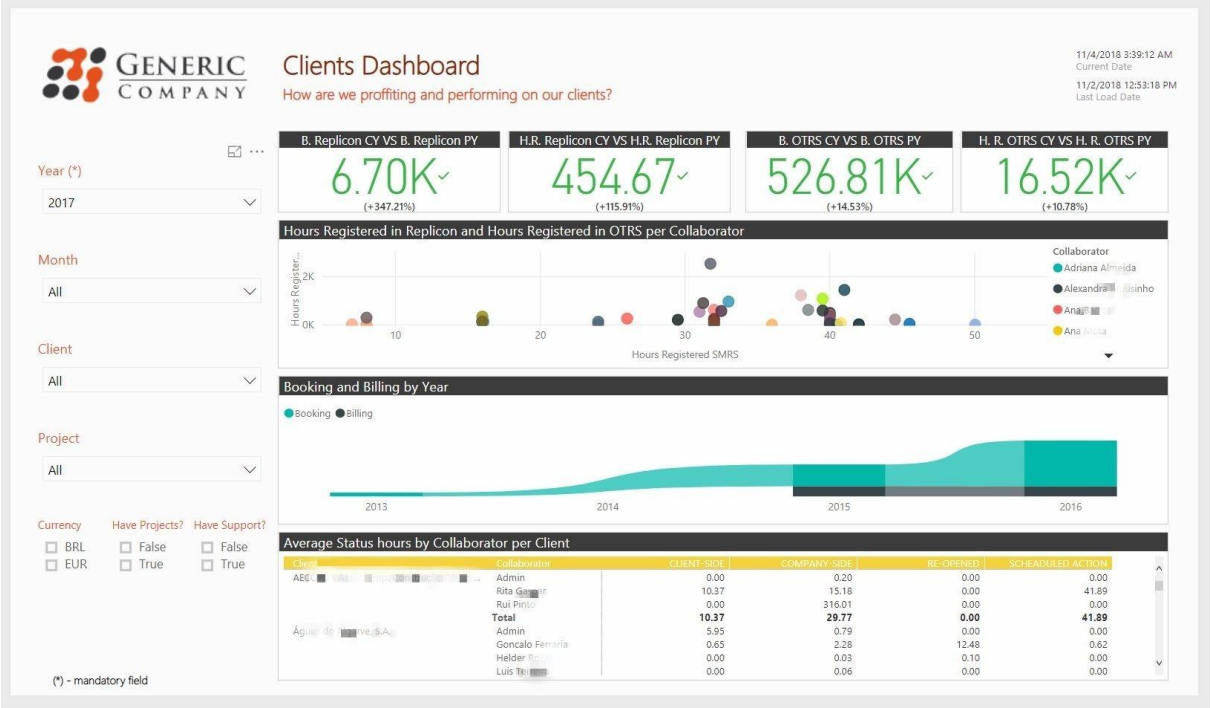


Figure 30 – Clients Dashboard

In this dashboard, we have additional filters due to two main reasons. It is needed a time slicer (represented by Year and Month filter) since most of the visualizations are textual (KPI's and Table). Adding a time dimension to the visualization would compromise its interpretation (Scatter Chart). The second reason is that the end-user would need to access a specific level of detail in the client that is considered in previous graphic all merged. It is mandatory that a year is selected in the corresponding filter

The KPI's in the top are related with hours registered in OTRS, Replicon and the associated billing to those hours. It is expected that we have more hours registered in Replicon than OTRS since in Replicon we register all the time passed at work regardless if it is a client or project registered in OTRS. The target used to compare if the measure is positive or negative is the comparison with the year before of the year selected.

4.4.4.4. Collaborators Dashboard

The Collaborators Dashboard explores PowerBI potentials in filters synchronizations enabling the end user to compares in the same dimension different subsets.

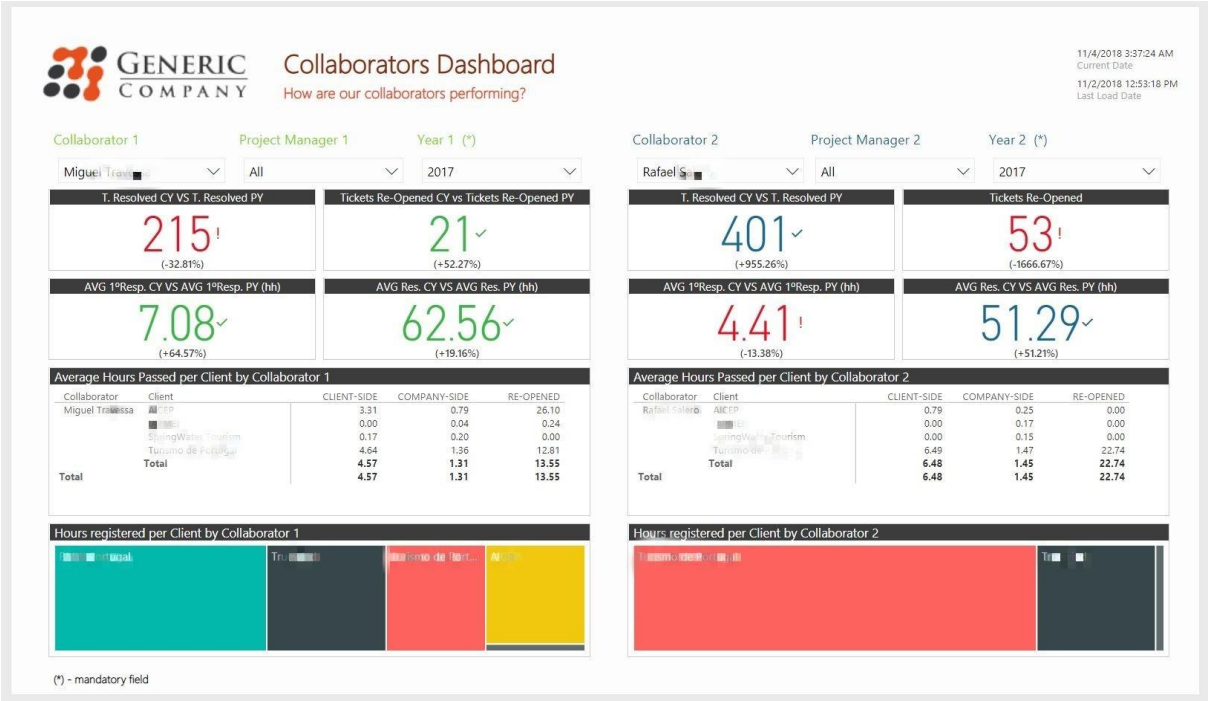


Figure 31 – Collaborators Dashboard

This is the dashboard with most significant number of visualizations embedded, alas, half of them are repeated, so the subjects to be compared are analyzed under the same metrics.

The KPI's uses as a target, to compare if the measure is positive or negative, the values of the year previous to the year currently selected. The metrics presented are concepts that are displayed in previous dashboards, as the average resolution and first response time, number of incidents closed and incidents re-opened.

Because most of the visualizations in this dashboard uses the data in the bottom of the dashboards, it is presented a *treemap* with the time registered in a specific client, giving a clear perspective of the percentage of time a collaborator dispends to a client.

4.4.4.5. Projects & Contracts Dashboard

The projects and contracts dashboard was created to present two subsets of data that were not presented before in previous dashboards, contracts and projects forecasts.

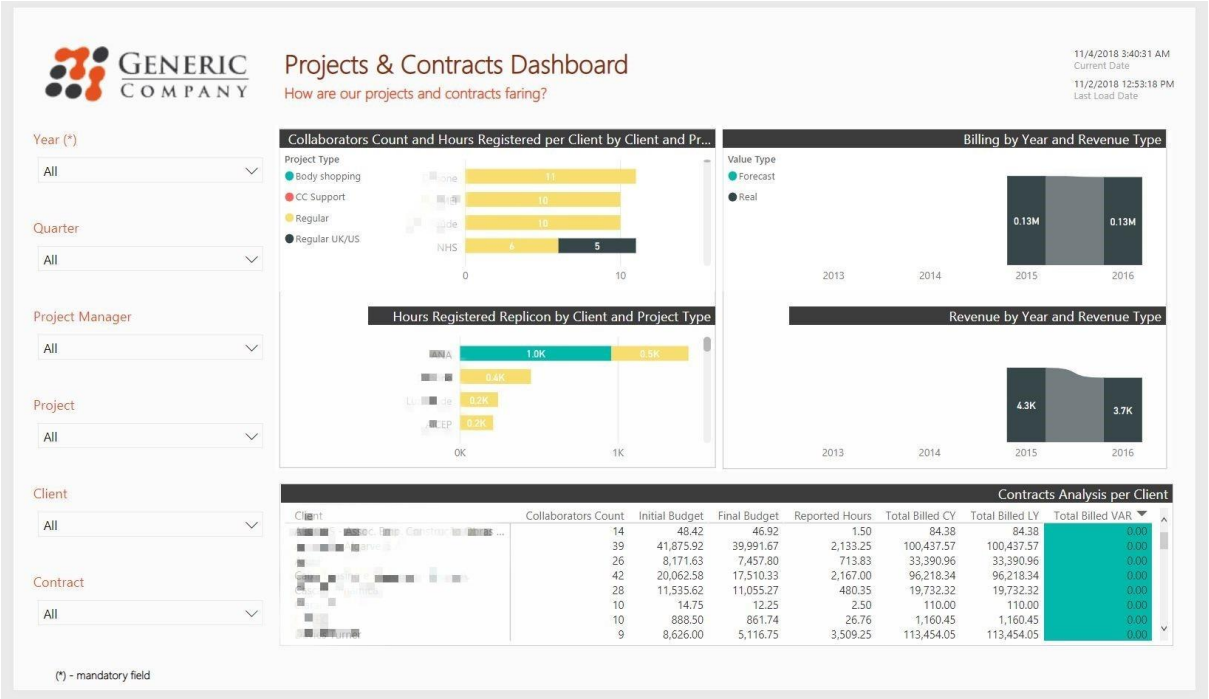


Figure 32 – Projects & Contracts Dashboards

The information made available is represented in dimensional tables of the DW instead of the factual tables, which led to an increasingly difficulty to dashboard design since the tables that contain this information do not relate between themselves. This situation was overcome using the Bridge tables described in the Data Model section of the current chapter.

In the dashboard half-top, we have a bar chart that presents the top 5 clients that have more time and collaborators allocated. Besides those bar charts, it is presented two graphics that show the evolution of Revenue and Billing, either effective or forecasted. The values forecasted are manually inputted in SMRS and are not calculated by any algorithm in this solution. Also, the time-related filters do not affect the values presented.

At the half bottom part of the dashboard, it is presented a table containing the values defined for the contracts associated with the client. This table contains the variation of the amounts billed in the selected year against the previous year and the budgets allocated to the fulfillment of the contract for the timeline analyzed.

This dashboard suffered the most with the lack of data from SMRS and had to be redesigned from the initial concept.

4.5. RESULTS AND VALIDATION

For the OTRS values, the validations in this solution were done by the comparison of some metrics by the old method, using a subset of data in an excel sheet, with the tabular results exported by graphics that compose the dashboard. Until the values extracted by both methods reached an exactness percentage above 95%, recalculations were made, on a first level through SQL query to the metadata

available in the Data Warehouse to the measures calculated in PowerBI, in case the results extracted from the database matched the Excel values.

Regarding the SMRS data, the values presented in the original data source and metrics calculated directly on that database match with the one on the Data Warehouse, alas, we have concluded that the SMRS only have a small percentage of all the records that are presented in Replicon. Unfortunately, this situation was only detected during the final stage of this project. Due to that reason, it will be addressed the results retrieved from the OTRS database to a more detailed level than the ones that would be possible to infer through SMRS information.

When analysing the results made available in the solution it is necessary to keep in mind that there is a subjacent reason for the values shown. Most of the causes for the values are familiar to the managers, and there are even technical reasons, so, for some subjects (either clients or collaborators) it is necessary to contextualize the situations with the values obtained.

The OTRS application was only adopted in 2014, yet, we even have data from 2004. That is due to the fact that the organization used another application for incident management and knowledge base at the same time. This led to two errors in the interpretation of the results in the dashboards

i. The incidents registered before 2014 do not have a backlog of the actions done, which in turn induces the algorithm used to calculate some metrics incorrect for those cases.

ii. Some incidents were created by the organization's users to identify general problems or tasks that are not meant to be solved at all but just to point them out.

To analyze the results obtained and verify the previously described situations it will assess three main graphics using the information available in all the dimensions.

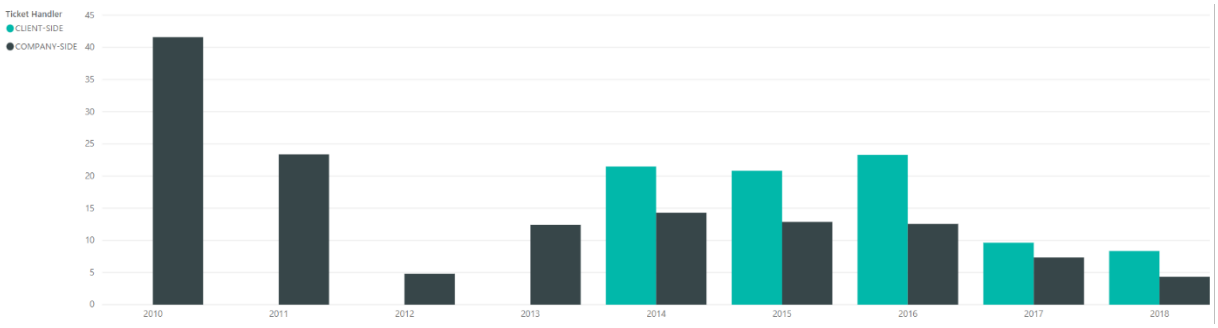


Figure 33 - Average Hours passed per Ticket Handler

Only after, and during, 2014 that we can state that most of the average time of a ticket lifespan is spent waiting for the client feedback than waiting for the organization collaborator to act on the report situation. Moreover, this analysis is done at a global client level, which would include incidents that are clear outliers on the average values but still weight on average values generally in a negative way by incrementing the time passed on the company side.

Although the solution developed can present information previous to 2014, in the next graphics we will only use the information made available after 2014 and due to the fact that only after that date we have all the data used in our measures and by reducing the timescale we have a better visualization.

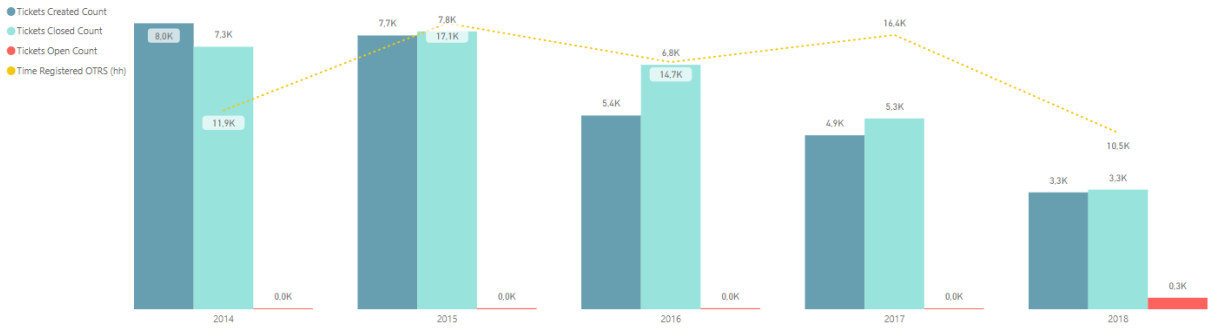


Figure 34 – Tickets overview and effective time spent to resolve

At first look, this results could induce the reader in error since we have more tickets resolved than created but that is due to the legacy tickets, created before 2014 that were only solved or closed years after.

This assessment is aligned with the company policies during the first three years when the creation of an official Customer Care department with a dedicated team to resolve a high number of pending situations.

In 2017, after most of the pending situations were resolved and some measures to prevent repetitive actions took place, it was possible to dedicate to more delicate problems having a reduced number of tickets resolved, in comparison to previous year, but maintaining the number hours registered.

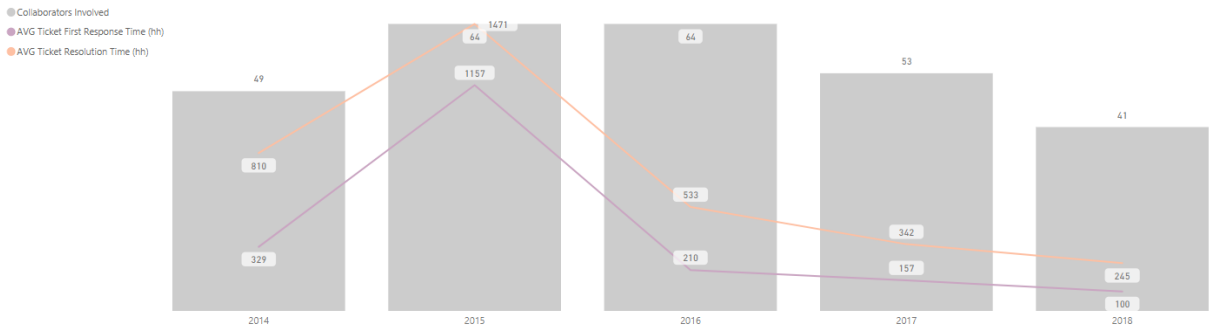


Figure 35 – Average time to resolve and respond to Incidents

Through this graphic, it is possible to correlate the number of collaborators that used OTRS for incident solving per year with performance indicators resolution time and first response time.

In conformity of the analysis done so far, in 2015 a high number of legacy incidents were resolved in the organization. Only when incidents were registered at OTRS that it was possible to record if an incident is resolved or first answered. The data processing engine under the solution can present value and for that reason, we have the highest value for those indicators on that year, since it is used for its calculations the summed up working hours for a number years.

During the current Year we have the lowest values ever achieved for resolution, and first response time, still, the values are quite high on a global perspective, so it is always necessary to assess the information by client/collaborator knowing the context that they are inserted.

5. PROJECT RELEVANCE

After the development of the solution using the Design Science Research methodology and implementing it in a real production environment, it is possible to review the checklist presented in the Literature Review chapter to evaluate the project so far. The checklist included the following topics:

1. Identification of the research question;
2. Concept to be developed and how it is presented;
3. Identification of the design processes associated with this solution;
4. Contextualization of the literature review that supports this projection;
5. Validations made to evaluate the quality of the solution;
6. Scientific knowledge formulated during the development of the project:
7. Achievement of the goals underlying the research question.

The focus of the research question in this Project Masters is to assess the possibility to provide a visual analytical solution to the Costumer Care management in the Information Technologies area to assist the decision-makers in an organization.

For that goal, the author justified that the dashboards, as a mean to present information, would achieve that end. The referred dashboards were contextualized in a BI Project due to the involvement of the business areas and the technological process to design and produce dashboards that would address the problem to solve.

In a technical perspective, it was necessary to support the design processes with the research done by Moody & Kortink (2000) to create the necessary foundations for data warehousing and high-performance data models. Without these foundations, the dashboards would not be as nearly as useful and it would not be possible to present as much information.

The dashboard's design was based on the researches of Few and Eckerson. While Eckerson's study assisted in identifying the key stakeholders for the dashboards and focus the themes in each dashboard for its respective audience, mitigating unnecessary or irrelevant information and how to present performance measures that add real value to the organization, Few describes a series of good practices to present information and captivate the users' attention. It is necessary to refer that the project itself is scalable. In case we want to add complementary data sources the solution does not need a complete redesign but an assessment on how to fit the current metadata with the one we want to add, besides the possible need to create a new dashboard for some concepts that the management level want to explore. Most of the information that was already collected will be used in an evolution of the current solution. Information associated with clients, collaborators and time is transversal for most BI projects.

Aside from the best practices identified in the literature review to develop a solution that would add value to the company, all the project development was assisted and accompanied by the sponsors in the organization this solution was implemented. Through their support and guidance, it was identified the metrics and performance indicators that should be presented in the dashboards and validated the

results through the comparison with the old methods used to retrieve that type of information. It is also necessary to point out that some metrics that were only manually calculated in semi-annual evaluations, due to the amount of time required to calculate those values, are now available on a daily base.

The knowledge retrieved from the development of this analytical solution is not just measured by just the definition of a scientific method, constructs or theory to approach the types of problems that were identified for this project. This type of projects are already being done for more than two decades, but the underlying knowledge and proficiency to use the best tools that are available in the market to present a solution that answers the requirements of the costumers.

The research question is satisfactorily answered at the moment the organization receives and uses an information system that condenses all the data made available by the daily operations and transforms it in valuable information that helps the company to make decisions based on facts.

6. CONCLUSIONS

To present a conclusion for this Masters Project it is necessary to review the goals that were defined for this project and assess the quality of the deliverables by contextualizing them with the objectives established in the begging. This project proposed to create a solution that would fit the following global requirements:

I. Develop a Business Intelligence solution that processes the available data to generate information for the company;
II. Empowering the multiple layers of the organization, with focus on the Costumer Care area, with an efficient tool to analyze the organization performance through a visual interface with pre-defined metrics.

The first objective can be contextualized in the Database Design and Data Manipulation chapter, and through it, we create the foundations to transform the data that is available in information that could be presented through a visual analytical tool. This step took over 70% of the practical work, either in modeling the databases that would house the information, the definition of the necessary ETL tasks and errors eliminations through the comparison of the data extracted to the designated recipients with the data in the original data sources. This project phase had to be revisited many times, especially during the dashboards creation, since many indicators that are presented in the current dashboards lacked the necessary inputs to be correctly formulated.

The second objective, after various iterations under the designated dashboard tool, was achieved after the development of a performance analysis tool with a visual interface. The iterations required for the product conception consisted of different approaches to how to present the visualizations for the desired indicators without compromising other metrics. For that, it was necessary to understand how the tool performed, under what circumstances and what did it differ from similar platforms available on the market. Although it was not necessary to explore the full potential of the platform, like programming in R, It is necessary to refer that the author needed to consolidate its *know-how* on the best practices of this software to deliver a solution that would fill its needs. Yet, it is still too early to correctly assess the achievement of the goal proposed since it is directly connected with the use of the tool by the decision makers in the organization. While the initial feedback was positive and already justified some changes on the personnel available for some projects, due to evaluation of the performance on some clients only in the long term, through the continuous use, is possible to state that the solution served its ends and will enhance end users to search for additional metrics and identify necessities that weren't assessed initially.

This project also had its share of limitations that impeded the use of its full potential even if all the design processes had been correctly done. Unfortunately, and unexpectedly, one of the data sources did not have all the data that it was initially expected to have and limited the scope of the solution. The SMRS database does not have all the data that is registered in Replicon by each collaborator, that situation alone undermined more than half of the initial dashboards planned for this project.

Considering that the Replicon is the application where all the collaborators in the organization register what they have done during their work hours, to which client, for which project, that alone would add a completely new dimension for this solution since that information would be made possible to infer a complete analysis on what is most important in a consultancy company: it's personnel.

Besides the lack of data from one data source, it was also detected two other situations that could compromise the integrity of the information presented in the dashboards:

- Some information that should be filled by the managerial level was missing, such as contracts associated to clients for customer care or forecast associated for existing projects that would be migrated (and were) to the next year. All these situations led that the dashboards would be more centralized in the information made available by the OTRS database and the possible measures that we could extract from that data.
- There are some customer care projects where the incidents management is being done by the internal client solution instead of using OTRS to record those tickets.

Alas, most of these situations can be resolved by solving the problems in the applications upstream and by the indication of the board level to the managerial level to correctly fill up all the information associated with the clients in a single repository. Through these two indications, it is possible to solve the problems reported and evolve the dashboards presented in this solution.

On a technical level, and after using other dashboard development tools, it is safe to state that the DW database model is not the optimal solution for the tool chosen to develop this solution. The perfect example of this statement is the necessity to create bridge tables, as explained in the section Data Model from the Dashboard Design.

Also, due to the fact that the same client or collaborator could be registered in different ways for different platforms, there is always a need to inspect and correct the information in all the data source, for this project the mapping was done through an Excel sheet in the future all the information should go directly in application to eliminate the use of easily losable files.

In the future, the company should try to integrate all possible data sources with information from the organization projects or activities, to the data warehouse to enhance analysis and enable the possibility to create more dashboards, more focused in specific subjects instead of showing global indicators. With more information, it will also be viable to install data mining algorithms that enable the organization to predict future outcomes.

7. REFERENCES

Abbaszadegan, A., & Grau, D. (2015). Assessing the Influence of Automated Data Analytics on Cost and Schedule Performance. *Procedia - Engineering*, *123*, 3–6. https://doi.org/10.1016/j.proeng.2015.10.047

Aparicio, M., & J. Costa, C. (2014). Data Visualization. *Communication Design Quarterly Review*, *3*(November), 7–11. https://doi.org/10.1145/2721882.2721883

Barrento, M., Neto, M. de C., & Caiado, J. C. (2013). Business intelligence admission to hospital emergency [Admissão da business intelligence às urgências hospitalares]. *Iberian Conference on Information Systems and Technologies, CISTI*, 1–5. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-84887970056&partnerID=40&md5=1899f8bb5b4e1deb743d7ca5c536c842

Campos, J., Sharma, P., Gorostegui, U., Jantunen, E., & Baglee, D. (2017). A Big Data Analytical Architecture for the Asset Management. *Procedia CIRP*, *64*, 369–374. https://doi.org/10.1016/j.procir.2017.03.019

Chaudhuri, S., Dayal, U., & Narasayya, V. (2011). An overview of business intelligence technology. *Communications of the ACM*, *54*(8), 88–98. https://doi.org/10.1145/1978542.1978562

Cooper, L. F. (2012). Global Business Intelligence Software Survey : Companies Desire Smaller , Better Targeted End-User Solutions. *BizTechReports*, *Internatio*, 1–22.

Côrte-real, N., Neto, M. de C., & Fátima, N. (2012). Business Intelligence Maturity Assessment Model for organizations. *7th Iberian Conference on Information Systems and Technologies (CISTI)*, 1–7.

Eckerson, W. W. (2009). How to Create and Deploy Effective Metrics. *TDWI Best Practices Report*, 1–11. Retrieved from https://www.microstrategy.com/Strategy/media/downloads/white-papers/TDWI_Performance-Management-Strategies.pdf

Elgendy, N., & Elragal, A. (2016). Big Data Analytics in Support of the Decision Making Process. *Procedia - Procedia Computer Science*, *100*, 1071–1084. https://doi.org/10.1016/j.procs.2016.09.251

Few, S. (2006). Information Dashboard Design: The Effective Visual Communication of Data. In *The effective visual communication of data* (pp. 4–35). O'REILLY. https://doi.org/10.1017/S0021849904040334

Few, S. (2007). Pervasive Hurdles to Effective Dashboard Design. *Perceptual Edge, Visual Business Intelligence Newsletter*, (January), 1–7.

Hevner, A., & Chatterjee, S. (2010). Design Research in Information Systems, *22*, 9–23. https://doi.org/10.1007/978-1-4419-5653-8

Hostmann, B., Ryaner, N., & Herschel, G. (2009). A practical framework for business intelligence and planning in midsize companies. *Gartner Report*, (October), 1–20.

Jarvinen, P. H. (2004). Research Questions Guiding Selection of an Appropriate Research Method.

Juice. (2015). A Guide to Creating Dashboards People Love to Use v2.0. Retrieved from www.juiceanalytics.com

Karlsen, I. A., Eidene, M. N., & Olsen, D. H. (2012). *Real Time Business Intelligence and Decision*

Making. University of Agder. Faculty of Economics and Social Sciences Department.

Kfouri, G., & Skyrius, R. (2016). *Factors influencing the implementation of business intelligence among small and medium enterprises in Lebanon* (No. ISSN 1392-0561). *Informacijos mokslai* (Vol. 76). Vilnius. https://doi.org/https://doi.org/10.15388/Im.2016.76.10384 Factors

Kimball, R., & Ross, M. (2007). *The Data warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. *Journal of Experimental Psychology: General* (3rd ed., Vol. 136). John Wiley & Sons, Inc.

Lavalle, S., Hopkins, M. S., Lesser, E., Shockley, R., & Kruschwitz, N. (2010). Analytics : The New Path to Value. *MIT Sloan Management Review*, 1–24. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Analytics+:+The+New+Path+to+Value#0

Magdalena, R., & Luminita, S. (2012). Optimizing time in business with Business Intelligence solution. *Procedia - Social and Behavioral Sciences*, *62*, 638–648. https://doi.org/10.1016/j.sbspro.2012.09.106

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Elsevier - Decision Support Systems*, *15*, 251–266.

Microsoft. (2018a). Excel. Retrieved from https://products.office.com/pt-pt/excel

Microsoft. (2018b). Power BI. Retrieved from https://powerbi.microsoft.com/en-us/

Microsoft. (2018c). PowerPivot. Retrieved from https://support.office.com/en-us/article/power-pivot-powerful-data-analysis-and-data-modeling-in-excel-a9c2c6e2-cc49-4976-a7d7-40896795d045

Microsoft. (2018d). SQL Server Management Studio. Retrieved from https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-2017

Microsoft. (2018e). Visual Studio. Retrieved from https://visualstudio.microsoft.com/

Moody, D., & Kortink, M. A. . (2000). From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design. *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2000)*, *2000*, 5–16.

Nunamaker, J., Chen, M., & Purdin, T. (1991). Systems development in Information Systems research. *Journal of Management Information Systems*. https://doi.org/10.1109/ISIE.1992.279627

O 'brien, J. A., Marakas, G. M., & Lalit, M. R. (2008). *Management Information @BULLET -Systems* (7th Editio, Vol. 4). Tata Mc-Graw Hills. https://doi.org/10.1108/eb000831

OTRS. (2018). OTRS. Retrieved from https://otrs.com/pt/home/

Parmenter, D. (2007). *Key performance indicators : developing, implementing, and using winning KPIs* (First). Hoboken, New Jersey: John Wiley & Sons, Inc., Hoboken, New Jersey.

Petrovski, F. (2016). *Development of Information Management solution for Measuring People Performance*. Universidade Nova de Lisboa, Instituto Superior de Estatística e Gestão de Informação.

Phillips-Wren, G., Iyer, L. S., Kulkarni, U., & Ariyachandra, T. (2015). Business analytics in the context of big data: A roadmap for research. *Communications of the Association for Information*

Systems, *37*, 448–472.

Pipe, P. (1997). The Data Mart: A New Approach to Data Warehousing. *International Review of Law Computers*, *11*(2), 251–261.

Replicon. (2018). Replicon. Retrieved from https://www.replicon.com/

Rita L. Sallam , Cindi Howson , Carlie J. Idoine , Thomas W. Oestreich , James Laurence Richardson, J. T. (2017). Magic Quadrant for Business Intelligence and Analytics Platforms.

Santos, A. F. S. dos, & Neto, M. de C. (2011). Best practices in enterprise performance management systems implementation projects at financial sector enterprises. *Information Systems and Technologies (CISTI), 2011 6th Iberian Conference On*, 1–6.

Sturdy, G. R. (2012). *Customer Relationship Management Using Business Intelligence*. Cambridge Scholars Publishing.

Turban, E., Sharda, R., & Delen, D. (2011). *Decision Support and Business intelligence Systems* (Ninth). Pearson.

Vodapalli, N. K. (2009). *Critical Success Factors of BI Implementation*. University of Copenhagen. Retrieved from http://www.itu.dk/~navvod/CSFsOfBIimpl.pdf

Wieder, B., & Ossimitz, M.-L. (2015). The Impact of Business Intelligence on the Quality of Decision Making – A Mediation Model. *Procedia - Computer Science*, *64*, 1163–1171. https://doi.org/10.1016/j.procs.2015.08.599

APPENDIX A

VISUAL STUDIO BI DATA TOOLS

Source ETL

Control Flow

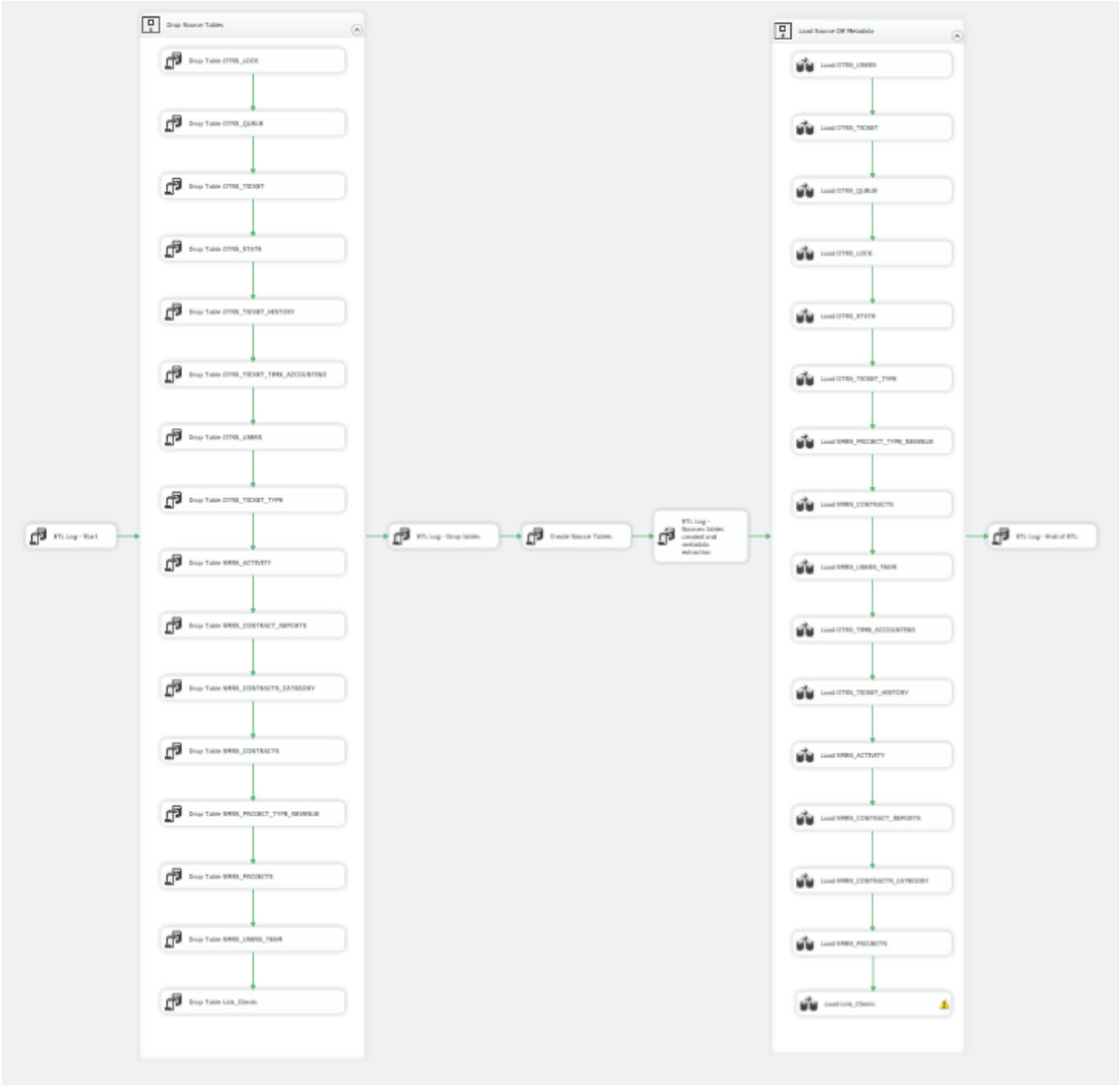


Figure A1 – Source ETL Control Flow

Data Flows

All the loadings are done directly from the original databases to the Source database without any transformation and maintaining the same number of columns.

SA ETL

Control Flow

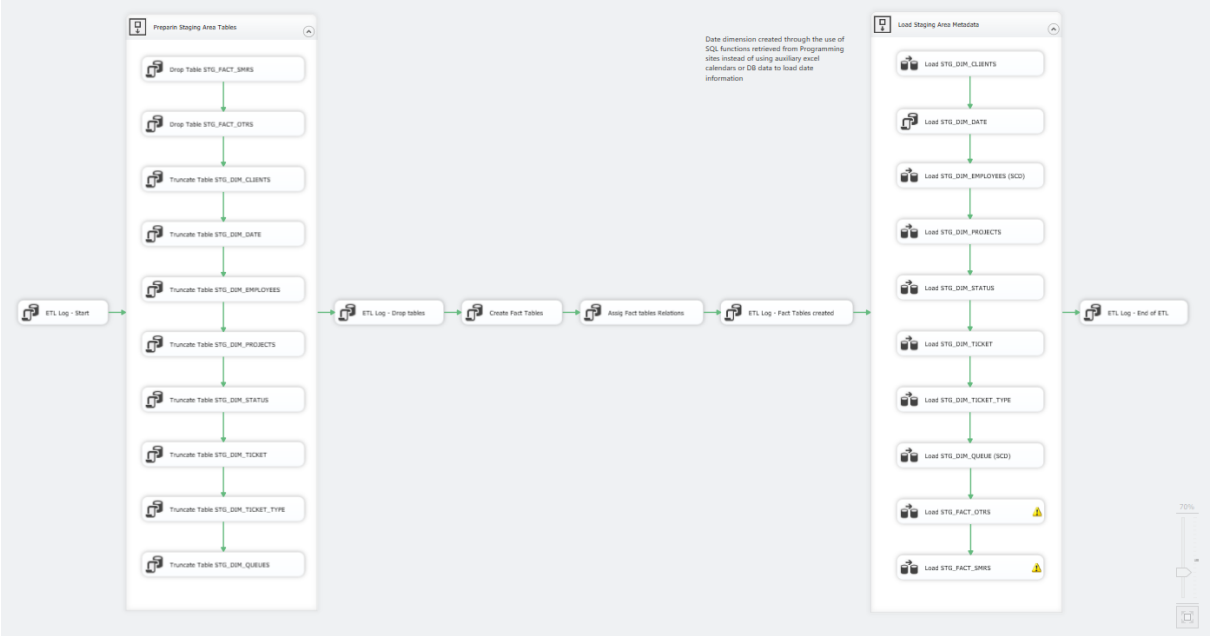


Figure A2 – SA ETL Control Flow

Data Flow

Table A1 – SA STG_DIM_CLIENTS Data Flow

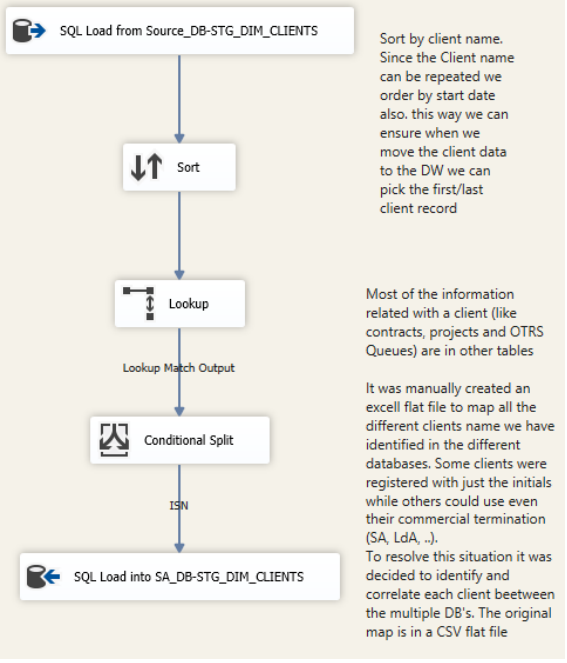
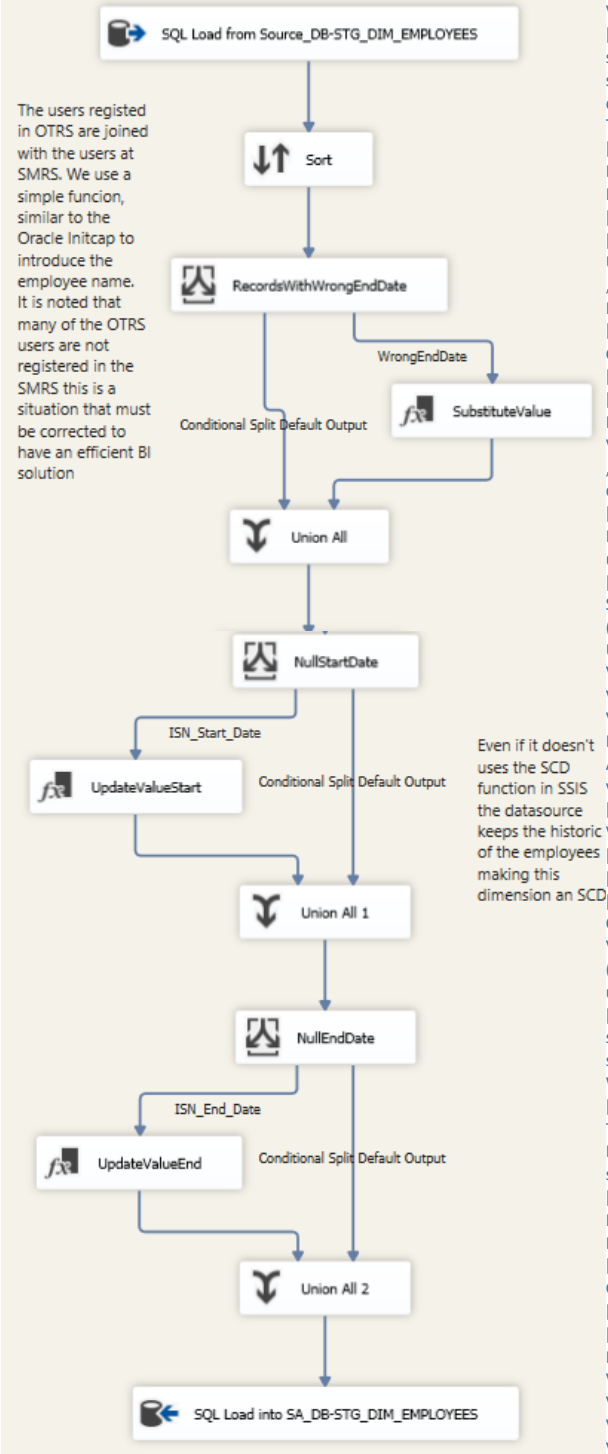
Data Flow Name: STG_DIM_CLIENTS	
Schema	**SQL Query**
	Select SO.id AS [BK_CLIENT] , SO.Client_name_Final AS [CLIENT_NAME] , SO.CLIENT_NAME_QUEUE AS [NAME_OTRS] , SO.CLIENT_NAME_PROJECT AS [NAME_PROJECT] , SO.CLIENT_NAME_ACTIVITY AS [NAME_ACTIVITY] , SO.CLIENT_NAME_CONTRACT AS [NAME_CONTRACT] , SO.Have_Project , SO.Have_Support , SO.Have_Activities, COUNT(DISTINCT P.PROJECT_ACCOUNT) AS [NUMBER_OF_ACCOUNTS] , MIN(A.ENTRYDATE) AS [START_DATE] FROM [Thesis_Source].[dbo].[Link_Clients_SMRS_OTRS] SO LEFT JOIN [Thesis_Source].[dbo].[SMRS_PROJECTS] P ON (P.CLIENT_NAME = SO.CLIENT_NAME_PROJECT) LEFT JOIN [Thesis_Source].[dbo].[SMRS_ACTIVITY] A ON (A.PROJECTCODE = P.PROJECT_CODE) GROUP BY SO.id , SO.Client_name_Final , SO.CLIENT_NAME_QUEUE , SO.CLIENT_NAME_PROJECT , SO.CLIENT_NAME_ACTIVITY , SO.CLIENT_NAME_CONTRACT , SO.Have_Project , SO.Have_Support , SO.Have_Activities ORDER BY SO.Client_name_Final

Table A2 – SA STG_DIM_EMPLOYEES Data Flow

Data Flow Name: STG_DIM_EMPLOYEES	
Schema	**SQL Query**




```
SELECT ROW_NUMBER() OVER (ORDER BY V2.[EMPLOYEE_LOGIN],
V2.[EMPLOYEE_START_DATE], V2.[EMPLOYEE_END_DATE]) AS
[BK_EMPLOYEE] , V2.*  FROM (SELECT CASE WHEN
sut.EMPLOYEE_USERNAME IS NULL THEN us.USER_LOGIN ELSE
sut.EMPLOYEE_USERNAME END  AS [EMPLOYEE_LOGIN] ,
dbo.InitCap(REPLACE(CASE  WHEN sut.EMPLOYEE_USERNAME IS NULL
THEN us.USER_LOGIN ELSE sut.EMPLOYEE_USERNAME END,'.',' ') )  AS
[EMPLOYEE_NAME] ,  CASE WHEN ISNULL(US.USER_VALID,0) = 0 THEN 'NO'
ELSE 'YES'  END AS [OTRS_ISVALID] ,  CASE WHEN
ISNULL(sut.USER_ISACTIVE,'NO') = 'NO' THEN 'NO'  ELSE 'YES'  END AS
[SMRS_ISVALID] ,  REPLACE(sut.EMPLOYEE_TYPE,'TC','Truewind') AS
[EMPLOYEE_TYPE]  , sut.EMPLOYEETYPE_TEAM AS [EMPLOYEE_TEAM] ,
US.USER_ID AS [OTRS_ID] ,  ISNULL(sut.VALUE_HOUR, 0) AS [VALUE_HOUR]
,  ISNULL(SUT.CURRENCY_TYPE, 'EUR') AS [CURRENCY_TYPE] ,
ISNULL(sut.EMPLOYEE_START_DATE, CAST('2000-01-01' AS datetime))  AS
[EMPLOYEE_START_DATE] ,  ISNULL(sut.EMPLOYEE_END_DATE,
GETDATE()) AS [EMPLOYEE_END_DATE]  FROM
[Thesis_Source].[dbo].[SMRS_USERS_TEAM] SUT FULL OUTER JOIN
[Thesis_Source].[dbo].[OTRS_USERS]  US  ON (LOWER(US.USER_LOGIN) =
LOWER(SUT.EMPLOYEE_USERNAME))  UNION SELECT
V.[EMPLOYEE_LOGIN]  , dbo.InitCap(REPLACE(V.[EMPLOYEE_LOGIN],'.',' '))
, 'NO'  , 'NO'  , NULL  , NULL  , NULL  , 0  , 'EUR'  , CAST('2000-01-01' AS
datetime)  , GETDATE()  FROM (SELECT SA.ANALISTNAME AS
[EMPLOYEE_LOGIN]  FROM [Thesis_Source].[dbo].[SMRS_ACTIVITY] SA
EXCEPT SELECT CASE WHEN sut.EMPLOYEE_USERNAME IS NULL THEN
us.USER_LOGIN ELSE sut.EMPLOYEE_USERNAME END  AS
[EMPLOYEE_LOGIN]  FROM [Thesis_Source].[dbo].[SMRS_USERS_TEAM]
SUT FULL OUTER JOIN [Thesis_Source].[dbo].[OTRS_USERS]  US  ON
(LOWER(US.USER_LOGIN) = LOWER(SUT.EMPLOYEE_USERNAME)) ) V
UNION  SELECT V6.[EMPLOYEE_LOGIN] ,  V6.[EMPLOYEE_NAME]  ,
V6.[OTRS_ISVALID]  ,  V6.[SMRS_ISVALID]  , V6.[EMPLOYEE_TYPE]  ,
V6.[EMPLOYEE_TEAM]  , V6.[OTRS_ID]  , V6.[VALUE_HOUR]  ,
V6.[CURRENCY_TYPE]  , V6.[EMPLOYEE_START_DATE] ,
MIN(V6.[EMPLOYEE_END_DATE])  FROM ( SELECT  V3.[EMPLOYEE_LOGIN]
AS [EMPLOYEE_LOGIN] , V3.[EMPLOYEE_NAME]  AS [EMPLOYEE_NAME] ,
v3.[OTRS_ISVALID]  AS [OTRS_ISVALID] , v3.[SMRS_ISVALID]  AS
[SMRS_ISVALID] , V3.[EMPLOYEE_TYPE]  AS [EMPLOYEE_TYPE] ,
V3.[EMPLOYEE_TEAM]  AS [EMPLOYEE_TEAM] ,  V3.[OTRS_ID] AS
[OTRS_ID] , 0  AS [VALUE_HOUR] , V3.[CURRENCY_TYPE]  AS
[CURRENCY_TYPE] , CAST('2000-01-01' AS datetime)  AS
[EMPLOYEE_START_DATE] , CASE  WHEN V3.[EMPLOYEE_START_DATE] =
CAST('2000-01-01' AS datetime) THEN  V3.[EMPLOYEE_START_DATE] ELSE
V3.[EMPLOYEE_START_DATE]-1  END AS [EMPLOYEE_END_DATE] FROM
(SELECT CASE WHEN sut.EMPLOYEE_USERNAME IS NULL THEN
us.USER_LOGIN ELSE sut.EMPLOYEE_USERNAME END AS
[EMPLOYEE_LOGIN] ,  dbo.InitCap(REPLACE(CASE WHEN
sut.EMPLOYEE_USERNAME IS NULL  THEN us.USER_LOGIN ELSE
sut.EMPLOYEE_USERNAME END,'.',' ') )  AS [EMPLOYEE_NAME]  ,  CASE
WHEN ISNULL(US.USER_VALID,0) = 0 THEN 'NO' ELSE 'YES' END  AS
[OTRS_ISVALID] ,  CASE WHEN ISNULL(sut.USER_ISACTIVE,'NO') = 'NO'
THEN 'NO' ELSE 'YES' END AS [SMRS_ISVALID] ,
REPLACE(sut.EMPLOYEE_TYPE,'TC','Truewind')  AS [EMPLOYEE_TYPE]  ,
sut.EMPLOYEETYPE_TEAM  AS [EMPLOYEE_TEAM] ,  US.USER_ID AS
[OTRS_ID] ,  ISNULL(sut.VALUE_HOUR, 0) AS [VALUE_HOUR] ,
ISNULL(SUT.CURRENCY_TYPE, 'EUR') AS [CURRENCY_TYPE] ,
ISNULL(sut.EMPLOYEE_START_DATE, CAST('2000-01-01' AS datetime))  AS
[EMPLOYEE_START_DATE] ,  ISNULL(sut.EMPLOYEE_END_DATE,
GETDATE()) AS [EMPLOYEE_END_DATE]  FROM
[Thesis_Source].[dbo].[SMRS_USERS_TEAM] SUT FULL OUTER  JOIN
[Thesis_Source].[dbo].[OTRS_USERS]  US  ON (LOWER(US.USER_LOGIN) =
LOWER(SUT.EMPLOYEE_USERNAME))  ) V3 ) V6 GROUP BY
V6.[EMPLOYEE_LOGIN]  , V6.[EMPLOYEE_NAME]  ,  V6.[OTRS_ISVALID]  ,
V6.[SMRS_ISVALID]  , V6.[EMPLOYEE_TYPE]  , V6.[EMPLOYEE_TEAM]  ,
V6.[OTRS_ID]  , V6.[VALUE_HOUR]  , V6.[CURRENCY_TYPE]  ,
V6.[EMPLOYEE_START_DATE]  ) V2 WHERE 1=1  AND
V2.[EMPLOYEE_START_DATE] <> V2.[EMPLOYEE_END_DATE];
```

Table A3 – SA STG_DIM_PROJECTS Data Flow

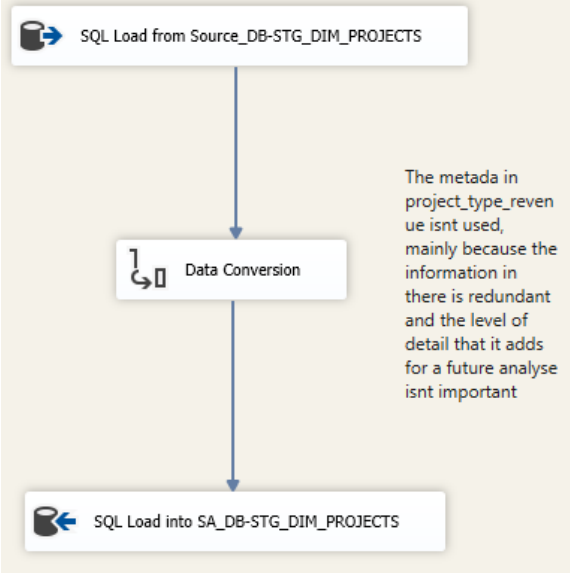
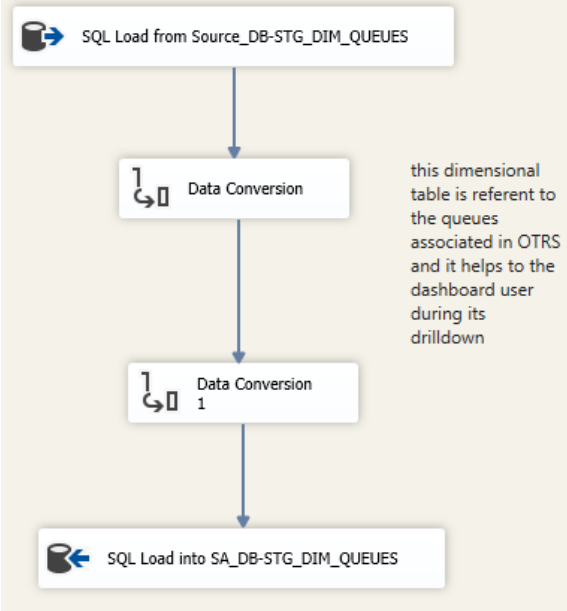
Data Flow Name: STG_DIM_PROJECTS	
Schema	**SQL Query**
	SELECT P.[PROJECT_CODE] ,P.[CLIENT_NAME] ,P.[PROJECT_NAME] ,P.[PROJECT_TYPE] ,dbo.InitCap(REPLACE(p.[PROJECT_ACCOUNT],'.',' ')) AS [PROJECT_ACCOUNT] ,dbo.InitCap(REPLACE(p.[PROJECT_PRJMANAGER],'.',' ')) AS [PROJECT_PRJMANAGER] ,ISNULL(PTR.[TEAM_NAME], 'N/D') AS [TEAM_NAME] ,P.[PROJECT_BILLINGBYACCOUNT] ,P.[PROJECT_ISSUPPORT] ,P.[PROJECT_ISLICENSING] ,P.[PROJECT_ISACTIVE] ,P.[PROJECT_CURRENCY] ,PTR.[PROJECT_YEAR] ,PTR.[PROJECT_MONTH] ,CAST(CONCAT(PTR.[PROJECT_YEAR],RIGHT(CONCAT('0',PTR.[PROJECT_MONTH]),2),'01') AS INT) AS PROJECT_DATE_ID ,ISNULL(DATEADD(MONTH,PTR.[PROJECT_MONTH]-1,CAST(CONVERT (datetime,convert(char(8),PTR.[PROJECT_YEAR])) AS DATETIME)),CAST('2000-01-01' AS DATETIME)) AS ACTIVE_DATE_START ,ISNULL(CAST(EOMONTH (DATEADD(MONTH,PTR.[PROJECT_MONTH]-1,CAST(CONVERT (datetime,convert(char(8),PTR.[PROJECT_YEAR])) AS DATETIME))) AS DATETIME),GETDATE()) AS ACTIVE_DATE_END ,PTR.[PROJECT_REVENUE_TYPE] ,PTR.[PROJECT_VALUE_TYPE] ,PTR.[PROJECT_BOOKING] ,PTR.[PROJECT_BILLING] ,PTR.[PROJECT_VALUE] ,PTR.[PROJECT_REVENUE] ,CASE WHEN ISNULL(C.[CONTRACT_ID],0) = 0 THEN 'No' ELSE 'Yes' END AS HAVE_CONTRACT ,ISNULL(C.CONTRACT_COSTPERHOUR,0) AS CONTRACT_COSTPERHOUR ,ISNULL(C.CONTRACT_CURRENCY, 'EUR') AS CONTRACT_CURRENCY ,ISNULL(C.CONTRACT_ROUNDHOUR, 0) AS CONTRACT_ROUNDHOUR FROM [Thesis_Source].[dbo].[SMRS_PROJECTS] P LEFT JOIN [Thesis_Source].[dbo].[SMRS_PROJECT_TYPE_REVENUE] PTR ON (P.[PROJECT_ID] = PTR.[PROJECT_ID]) LEFT JOIN [Thesis_Source].[dbo].[SMRS_CONTRACTS] C ON (C.[PROJECT_ID] = P.[PROJECT_ID]) UNION ALL --Entries in the DIM_proj to map activitys to this dimension, when an activity with an entrydate outside of the project active_date_start and date_end as it originally mapped SELECT DISTINCT P.[PROJECT_CODE] , P.[CLIENT_NAME] , P.[PROJECT_NAME] , P.[PROJECT_TYPE] , dbo.InitCap(REPLACE(p.[PROJECT_ACCOUNT],'.',' ')) AS [PROJECT_ACCOUNT] , dbo.InitCap(REPLACE(p.[PROJECT_PRJMANAGER],'.',' ')) AS [PROJECT_PRJMANAGER] , MIN(ISNULL(PTR.[TEAM_NAME], 'N/D')) AS [TEAM_NAME] , P.[PROJECT_BILLINGBYACCOUNT] AS [PROJECT_BILLINGBYACCOUNT] , P.[PROJECT_ISSUPPORT] AS [PROJECT_ISSUPPORT] , P.[PROJECT_ISLICENSING] AS [PROJECT_ISLICENSING] , P.[PROJECT_ISACTIVE] AS [PROJECT_ISACTIVE] , P.[PROJECT_CURRENCY] AS [PROJECT_CURRENCY] , YEAR(sa.[ENTRYDATE]) AS YEAR , MONTH(sa.[ENTRYDATE]) AS MONTH , CAST(FORMAT(sa.[ENTRYDATE],'yyyyMMdd') as int) AS PROJECT_DATE_ID , ISNULL(DATEADD(MONTH,MONTH(sa.[ENTRYDATE])-1,CAST(CONVERT (datetime,convert(char(8),YEAR(sa.[ENTRYDATE])) AS DATETIME)),CAST('2000-01-01' AS DATETIME)) AS ACTIVE_DATE_START , ISNULL(CAST(EOMONTH (DATEADD(MONTH,MONTH(sa.[ENTRYDATE])-1,CAST(CONVERT (datetime,convert(char(8),YEAR(sa.[ENTRYDATE])) AS DATETIME))) AS DATETIME),GETDATE()) AS ACTIVE_DATE_END , 'Real' AS A , 'Hours Register' AS B , 0 AS C , 0 AS D , 0 AS F , 0 AS G , 'N/A' AS HAVE_CONTRACT , MAX(ISNULL(C.CONTRACT_COSTPERHOUR,0)) AS CONTRACT_COSTPERHOUR , 'N/A' AS CONTRACT_CURRENCY , MAX(ISNULL(C.CONTRACT_ROUNDHOUR, 0)) AS CONTRACT_ROUNDHOUR FROM [Thesis_Source].[dbo].[SMRS_PROJECTS] P LEFT JOIN [Thesis_Source].[dbo].[SMRS_PROJECT_TYPE_REVENUE] PTR ON (P.[PROJECT_ID] = PTR.[PROJECT_ID]) LEFT JOIN [Thesis_Source].[dbo].[SMRS_CONTRACTS] C ON (C.[PROJECT_ID] = P.[PROJECT_ID]) JOIN [Thesis_Source].[dbo].[SMRS_ACTIVITY] sa ON (SA.PROJECTCODE = P.[PROJECT_CODE]) WHERE 1=1 GROUP BY CONCAT(CAST(FORMAT(sa.[ENTRYDATE],'yyyyMMdd') as int), P.[PROJECT_CODE]) , SA.ACTIVITY_ID , P.[PROJECT_CODE] , P.[CLIENT_NAME] , P.[PROJECT_NAME] , P.[PROJECT_TYPE] , dbo.InitCap(REPLACE(p.[PROJECT_ACCOUNT],'.',' ')) , dbo.InitCap(REPLACE(p.[PROJECT_PRJMANAGER],'.',' ')) , P.[PROJECT_BILLINGBYACCOUNT] , P.[PROJECT_ISSUPPORT] , P.[PROJECT_ISLICENSING] , P.[PROJECT_ISACTIVE] , P.[PROJECT_CURRENCY] , YEAR(sa.[ENTRYDATE]) , MONTH(sa.[ENTRYDATE]) , CAST(FORMAT(sa.[ENTRYDATE],'yyyyMMdd') as int) , ISNULL(DATEADD(MONTH,MONTH(sa.[ENTRYDATE])-1,CAST(CONVERT (datetime,convert(char(8),YEAR(sa.[ENTRYDATE])) AS DATETIME)),CAST('2000-01-01' AS DATETIME)) , ISNULL(CAST(EOMONTH (DATEADD(MONTH,MONTH(sa.[ENTRYDATE])-1,CAST(CONVERT (datetime,convert(char(8),YEAR(sa.[ENTRYDATE])) AS DATETIME))) AS DATETIME),GETDATE())

Table A4 – SA STG_DIM_QUEUES Data Flow

Data Flow Name: STG_DIM_QUEUES	
Schema	**SQL Query**




```
SELECT V.* FROM ( --Get the Queues created in OTRS and associate them with reports and contracts in SMRS, this way we can validate if the number of hours reported are coincident with the time reported in OTRS  SELECT q.[QUEUE_ID]
,q.[QUEUE_NAME] ,q.[QUEUE_CLIENT_NAME] ,q.[QUEUE_ISACTIVE]
,q.[QUEUE_CREATION_TIME] ,CASE  WHEN ISNULL(CC.[CONTRACT_ID],0) = 0 THEN 'No'  ELSE 'Yes'  END  AS HAVE_CONTRACT  ,ISNULL(C.CONTRACT_NAME, '-')  AS CONTRACT_NAME ,CC.[REPORT_REPLICON] ,CC.[REPORT_ISBILLABLE]
,CC.[REPORT_INCLUDEBUDGET]
,ISNULL(CR.[STARTDATE],q.[QUEUE_CREATION_TIME]) AS REPORT_START_DATE
,ISNULL(CR.[ENDDATE],GETDATE())  AS REPORT_END_DATE
,ISNULL(CR.[HOURS], 0)  AS REPORTED_HOURS ,ISNULL(CR.[INITIALBUDGET], 0) AS [INITIALBUDGET] ,ISNULL(CR.[FINALBUDGET], 0) AS [FINALBUDGET]
,ISNULL(C.CONTRACT_COSTPERHOUR,0) AS CONTRACT_COSTPERHOUR
,ISNULL(C.CONTRACT_CURRENCY, 'EUR') AS CONTRACT_CURRENCY
,ISNULL(C.CONTRACT_ROUNDHOUR, 0) AS CONTRACT_ROUNDHOUR  FROM [Thesis_Source].[dbo].[OTRS_QUEUE] Q  LEFT  JOIN
[Thesis_Source].[dbo].[SMRS_CONTRACTS_CATEGORY]  CC ON (Q.QUEUE_ID = CC.QUEUE_ID)  JOIN [Thesis_Source].[dbo].[SMRS_CONTRACT_REPORTS]  CR ON (CR.CONTRACTID = CC.CONTRACT_ID)  LEFT  JOIN
[Thesis_Source].[dbo].[SMRS_CONTRACTS]  C  ON (CC.CONTRACT_ID = C.CONTRACT_ID) UNION ALL --In case we have reported hours in OTRS outside the reports in SMRS we have to build an additional entry so we can map this reports, this select is for entrys before the contract or reports start date  SELECT q.[QUEUE_ID] ,q.[QUEUE_NAME] ,q.[QUEUE_CLIENT_NAME]
,q.[QUEUE_ISACTIVE] ,q.[QUEUE_CREATION_TIME] ,CASE  WHEN ISNULL(CC.[CONTRACT_ID],0) = 0 THEN 'No'  ELSE 'Yes'  END AS HAVE_CONTRACT  ,ISNULL(C.CONTRACT_NAME, '-')  AS CONTRACT_NAME ,'-' ,'NO' ,'NO' ,CAST('2000-01-01' AS DATETIME)  AS REPORT_START_DATE
,MIN(ISNULL(CR.[STARTDATE],GETDATE()))-1 AS REPORT_END_DATE ,0  AS REPORTED_HOURS ,0  AS [INITIALBUDGET] ,0  AS [FINALBUDGET]
,MIN(ISNULL(C.CONTRACT_COSTPERHOUR,0))  AS CONTRACT_COSTPERHOUR ,'EUR'  AS CONTRACT_CURRENCY ,0  AS CONTRACT_ROUNDHOUR  FROM
[Thesis_Source].[dbo].[OTRS_QUEUE] Q  LEFT  JOIN
[Thesis_Source].[dbo].[SMRS_CONTRACTS_CATEGORY]  CC ON (Q.QUEUE_ID = CC.QUEUE_ID)  LEFT  JOIN [Thesis_Source].[dbo].[SMRS_CONTRACT_REPORTS]  CR ON (CR.CONTRACTID = CC.CONTRACT_ID)  LEFT  JOIN
[Thesis_Source].[dbo].[SMRS_CONTRACTS]  C  ON (CC.CONTRACT_ID = C.CONTRACT_ID) GROUP  BY q.[QUEUE_ID] ,q.[QUEUE_NAME]
,q.[QUEUE_CLIENT_NAME] ,q.[QUEUE_ISACTIVE] ,q.[QUEUE_CREATION_TIME]
,CASE  WHEN ISNULL(CC.[CONTRACT_ID],0) = 0 THEN 'No'  ELSE 'Yes'  END
,ISNULL(C.CONTRACT_NAME, '-')  UNION ALL --In case we have reported hours in OTRS outside the reports in SMRS we have to build an additional entry so we can map this reports, this select is for entrys after the contract or reports start date  SELECT q.[QUEUE_ID] ,q.[QUEUE_NAME] ,q.[QUEUE_CLIENT_NAME]
,q.[QUEUE_ISACTIVE] ,q.[QUEUE_CREATION_TIME] ,CASE  WHEN ISNULL(CC.[CONTRACT_ID],0) = 0 THEN 'No'  ELSE 'Yes'  END AS HAVE_CONTRACT  ,ISNULL(C.CONTRACT_NAME, '-')  AS CONTRACT_NAME ,'-' ,'NO' ,'NO' ,MAX(ISNULL(CR.ENDDATE,GETDATE()))+1  AS REPORT_START_DATE
,GETDATE()+2  AS REPORT_END_DATE ,0  AS REPORTED_HOURS ,0  AS [INITIALBUDGET] ,0  AS [FINALBUDGET]
,MAX(ISNULL(C.CONTRACT_COSTPERHOUR,0))  AS CONTRACT_COSTPERHOUR ,'EUR'  AS CONTRACT_CURRENCY ,0  AS CONTRACT_ROUNDHOUR FROM
[Thesis_Source].[dbo].[OTRS_QUEUE] Q  LEFT JOIN
[Thesis_Source].[dbo].[SMRS_CONTRACTS_CATEGORY]  CC ON (Q.QUEUE_ID = CC.QUEUE_ID)  LEFT JOIN [Thesis_Source].[dbo].[SMRS_CONTRACT_REPORTS]  CR ON (CR.CONTRACTID = CC.CONTRACT_ID)  LEFT  JOIN
[Thesis_Source].[dbo].[SMRS_CONTRACTS]  C  ON (CC.CONTRACT_ID = C.CONTRACT_ID) GROUP  BY q.[QUEUE_ID] ,q.[QUEUE_NAME]
,q.[QUEUE_CLIENT_NAME]  ,q.[QUEUE_ISACTIVE] ,q.[QUEUE_CREATION_TIME]
,CASE WHEN ISNULL(CC.[CONTRACT_ID],0) = 0 THEN 'No' ELSE 'Yes'
,ISNULL(C.CONTRACT_NAME, '-')  ) V
WHERE   V.REPORT_START_DATE < V.REPORT_END_DATE
```

Table A5 – SA STG_DIM_STATUS Data Flow

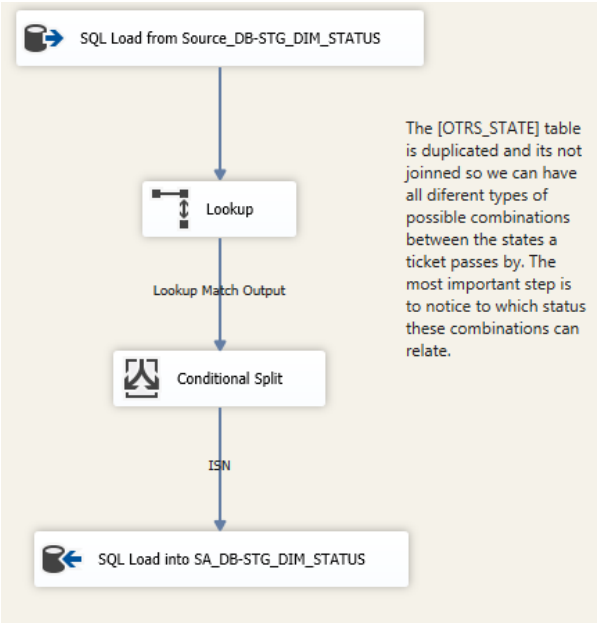
Data Flow Name: STG_DIM_STATUS	
Schema	**SQL Query**
	SELECT ROW_NUMBER() OVER (ORDER BY V.[OLD_STATE], V.[NEW_STATE]) AS [BK_STATUS] , V.[STATE_ID1] , ISNULL(V.[STATE_ID2],CAST('99' AS INT)) AS [FK_STATE2] , V.[OLD_STATE] , V.[NEW_STATE] , V.STATUS , CASE WHEN V.OLD_STATE IN ('New', 'Open') and V.NEW_STATE in ('Approved','Resolution by others','Verification in progress','Resolution by others','Waiting deployment','Waiting implementation date','Waiting customer action') THEN 'First time Response' WHEN V.OLD_STATE IN ('New', 'Open') and V.NEW_STATE in ('Rejected','Closed successful','Closed unsuccessful', 'Canceled','Removed') THEN 'First time Response and Conclusion' ELSE 'No Event' END AS [EVENT] FROM (SELECT s1.STATE_ID AS [STATE_ID1] , s2.STATE_ID AS [STATE_ID2] , UPPER(LEFT(s1.STATE_NAME,1))+LOWER(SUBSTRING(s1.STATE_NAME,2,LEN(s1.STATE_NAME))) AS [OLD_STATE] , UPPER(LEFT(s2.STATE_NAME,1))+LOWER(SUBSTRING(s2.STATE_NAME,2,LEN(s2.STATE_NAME))) AS [NEW_STATE] , CASE WHEN S1.[STATE_ID] IN (1,4,10,15,17) THEN 'COMPANY-SIDE' WHEN S1.[STATE_ID] IN (12,16) THEN 'CLIENT-SIDE' WHEN S1.[STATE_ID] IN (6,7,8,13,18) THEN 'SCHEADULED ACTION' WHEN S1.[STATE_ID] IN (2,3,5,11,14) AND (S2.[STATE_ID] IN (2,3,5,11,14,9, 99)) THEN 'CLOSED' WHEN S1.[STATE_ID] IN (2,3,5,11,14) AND S2.[STATE_ID] NOT IN (2,3,5,11,14,9) THEN 'RE-OPENED' WHEN S1.[STATE_ID] IN (9) THEN 'MERGED' WHEN S1.[STATE_ID] = 98 AND S2.[STATE_ID] = 98 THEN 'HOURS REG' WHEN S1.[STATE_ID] = 98 AND (S2.[STATE_ID] IN (2,3,5,11,14,9, 99)) THEN 'CLOSED' WHEN S1.[STATE_ID] = 98 AND (S2.[STATE_ID] NOT IN (2,3,5,11,14,9)) THEN 'COMPANY-SIDE' END AS [STATUS] FROM (SELECT S1.* FROM [Thesis_Source].[dbo].[OTRS_STATE] S1 UNION ALL SELECT 98 , 'Register OTRS work hours' , 'OTRS hours registration' , 'User is registing the hours used to solve the ticket' , 1 , NULL , NULL) s1 , (SELECT S2.* FROM [Thesis_Source].[dbo].[OTRS_STATE] s2 UNION ALL SELECT 99 , NULL , NULL , NULL , NULL , NULL , NULL UNION ALL SELECT 98 , 'Register OTRS work hours' , 'OTRS hours registration' , 'User is registing the hours used to solve the ticket' , 1 , NULL , NULL) s2) V

Table A6 – SA STG_DIM_TICKET_TYPE Data Flow

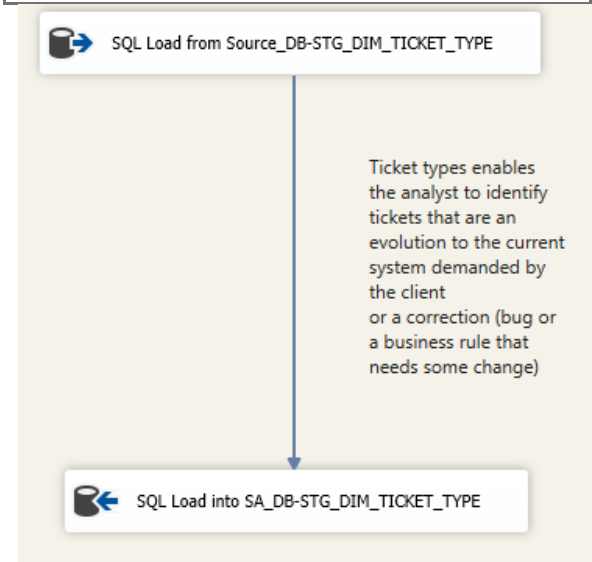
Data Flow Name: STG_DIM_TICKET_TYPE	
Schema	**SQL Query**
	SELECT [TYPE_ID] AS [BK_TYPE] ,[TYPE_NAME] ,[TYPE_ISVALID] ,[TYPE_CREATION_DATE] ,[TYPE_CHANGE_DATE] FROM [Thesis_Source].[dbo].[OTRS_TICKET_TYPE]

Table A7 – SA STG_DIM_CLIENTS Data Flow

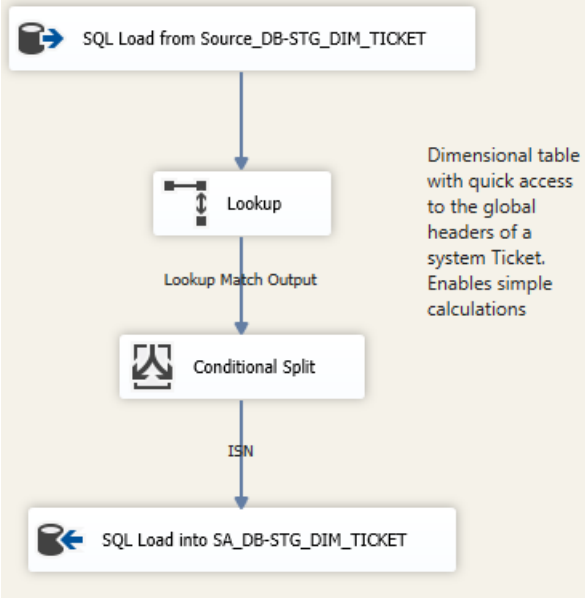
Data Flow Name: STG_DIM_TICKET	
Schema	**SQL Query**
	SELECT V.* , ROUND(DBO.UFN_CALC_TIME(V.[CREATE_TIME],[TICKET_CLOSE_DATE])/ 3600.00,2) AS [TICKET_CLOSE_TIME] , CASE WHEN Q.[DATE_REGISTERED] IS NULL AND V.[TICKET_CLOSE_DATE] IS NOT NULL THEN V.[TICKET_CLOSE_DATE] ELSE Q.[DATE_REGISTERED] END AS [TICKET_FIRST_RESPONSE_DATE] , ROUND(DBO.UFN_CALC_TIME(V.[CREATE_TIME],CASE WHEN Q.[DATE_REGISTERED] IS NULL AND V.[TICKET_CLOSE_DATE] IS NOT NULL THEN V.[TICKET_CLOSE_DATE] ELSE Q.[DATE_REGISTERED] END)/ 3600.00,2) AS [TICKET_FIRST_RESPONSE_TIME] FROM (SELECT t.[TICKET_ID] AS [BK_TICKET] , t.[TN] AS [TICKET_NUMBER] , dbo.INITCAP(REPLACE(U.USER_LOGIN,'.',' ')) AS [EMPLOYEE_RESPONSIBLE] , S.STATE_NAME AS [CURRENT_STATE] , q.[QUEUE_CLIENT_NAME] AS [CLIENT] , q.[QUEUE_NAME] , REPLACE(dbo.INITCAP(REPLACE(SUBSTRING(t.[TICKET_SOLICITOR],1,dbo.INSTR(t.[TICKET_SOLICITOR],'@',1,1)) ,'.',' ')) ,'@','') AS [SOLICITOR] , t.[CREATE_TIME] , t.[CHANGE_TIME] AS [LAST_CHANGE_TIME] , CASE WHEN S.STATE_NAME IN('canceled','closed successful','closed unsuccessful','rejected') THEN t.[CHANGE_TIME] ELSE NULL END AS [TICKET_CLOSE_DATE] FROM [Thesis_Source].[dbo].[OTRS_TICKET] T , [Thesis_Source].[dbo].[OTRS_QUEUE] Q , [Thesis_Source].[dbo].[OTRS_USERS] U , [Thesis_Source].[dbo].[OTRS_STATE] S WHERE 1=1 AND t.QUEUE_ID = Q.QUEUE_ID AND T.USER_ID = U.USER_ID AND T.TICKET_STATE_ID = S.STATE_ID) V LEFT JOIN (SELECT V.[FK_TICKET] , MIN(V.[DATE_REGISTERED2]) AS [DATE_REGISTERED] FROM (SELECT V1.[FK_TICKET] , V1.[FK_STATE] AS [FK_STATE1] , V2.[FK_STATE] AS [FK_STATE2] , V1.[DATE_REGISTERED] AS [DATE_REGISTERED1] , V2.[DATE_REGISTERED] AS [DATE_REGISTERED2] FROM (SELECT ROW_NUMBER() OVER (ORDER BY tH.TICKET_ID, TH.TICKET_HIST_ID AS [ROW_HIST] , TH.TICKET_ID AS [FK_TICKET] , TH.STATE_ID AS [FK_STATE] , T.TYPE_ID AS [FK_TICKET_TYPE] , TH.OWNER_ID AS [FK_USER] , MIN(TH.CREATE_TIME) AS [DATE_REGISTERED] , TH.QUEUE_ID AS [FK_QUEUE] FROM [Thesis_Source].[dbo].[OTRS_TICKET_HISTORY] TH , [Thesis_Source].[dbo].[OTRS_TICKET] T WHERE 1 = 1 AND T.TICKET_ID = TH.TICKET_ID AND TH.CREATE_TIME IS NOT NULL GROUP BY TH.TICKET_ID , TH.STATE_ID , T.TYPE_ID , TH.OWNER_ID , TH.TICKET_HIST_ID , TH.QUEUE_ID) V1 LEFT JOIN (SELECT ROW_NUMBER() over (order by tH.TICKET_ID, TH.TICKET_HIST_ID) AS [ROW_HIST] , TH.TICKET_ID AS [FK_TICKET] , TH.STATE_ID AS [FK_STATE] , T.TYPE_ID AS [FK_TICKET_TYPE] , TH.OWNER_ID AS [FK_USER] , MIN(TH.CREATE_TIME) AS [DATE_REGISTERED] , TH.QUEUE_ID AS [FK_QUEUE] FROM [Thesis_Source].[dbo].[OTRS_TICKET_HISTORY] TH , [Thesis_Source].[dbo].[OTRS_TICKET] T WHERE 1 = 1 AND T.TICKET_ID = TH.TICKET_ID AND TH.CREATE_TIME IS NOT NULL GROUP BY TH.TICKET_ID , TH.STATE_ID , T.TYPE_ID , TH.OWNER_ID , TH.TICKET_HIST_ID , TH.QUEUE_ID) V2 ON (V1.[FK_TICKET] = V2.[FK_TICKET] AND V2.[ROW_HIST] = V1.[ROW_HIST]+1)) V WHERE 1 = 1 AND (V.FK_STATE1 IN (1,4) AND V.FK_STATE2 IN (2, 3, 5, 10, 11, 12, 13, 14, 15, 16, 17)) GROUP BY V.[FK_TICKET]) Q ON (Q.FK_TICKET = V.[BK_TICKET])

Table A8 – SA STG_FACT_OTRS Data Flow

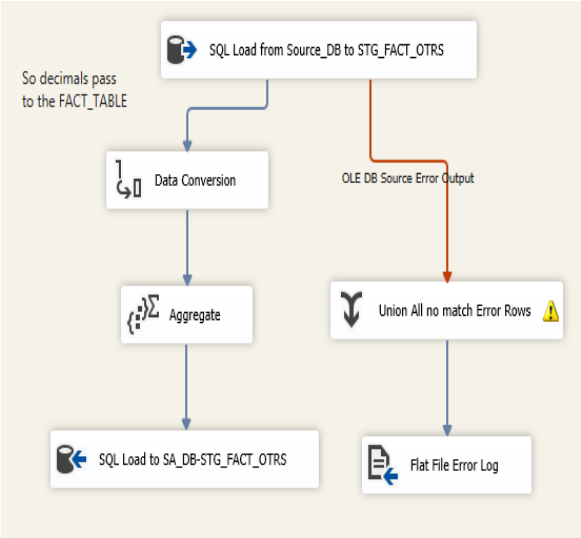
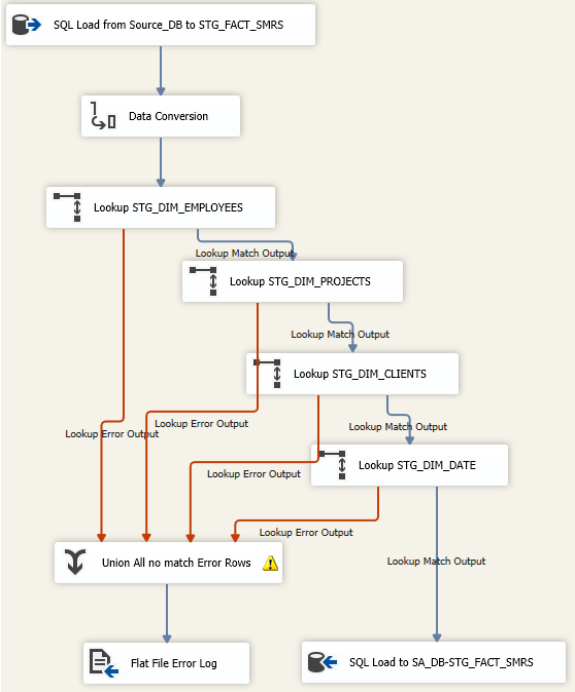
Data Flow Name: STG_FACT_OTRS	
Schema	**SQL Query**
	SELECT T.[SK_TICKET] AS [FK_TICKET] , ST.SK_STATUS AS [FK_STATE] , TT.[SK_TYPE] AS [FK_TICKET_TYPE] , Q.[SK_QUEUE] AS [FK_QUEUE] , LC.[SK_CLIENT] AS [FK_CLIENT], EMP.SK_EMPLOYEE AS [FK_EMPLOYEE] , V.[FK_DATE] AS [FK_DATE] , V.[DATE_REGISTERED1] AS [DATE_REGISTERED] , ROUND(V.[HOURS_PASSED] , 2) AS [HOURS_PASSED] , CASE WHEN V.[HOURS_REGISTERED] <= ISNULL(Q.[CONTRACT_ROUNDHOUR], V.[HOURS_REGISTERED]) AND V.[HOURS_REGISTERED] > 0 THEN ISNULL(Q.[CONTRACT_ROUNDHOUR], V.[HOURS_REGISTERED]) ELSE V.[HOURS_REGISTERED] END AS [HOURS_REGISTERED] , CASE WHEN V.[HOURS_REGISTERED] <= ISNULL(Q.[CONTRACT_ROUNDHOUR], V.[HOURS_REGISTERED]) AND V.[HOURS_REGISTERED] > 0 THEN ISNULL(Q.[CONTRACT_ROUNDHOUR], V.[HOURS_REGISTERED]) ELSE V.[HOURS_REGISTERED] END * Q.[CONTRACT_COSTPERHOUR] AS [HOURS_BILLED_OTRS] FROM (SELECT V1.[FK_TICKET] , V1.[FK_STATE] AS [FK_STATE1] , V2.[FK_STATE] AS [FK_STATE2] , V1.[FK_TICKET_TYPE] , V1.[FK_QUEUE] , V1.[FK_USER] , CAST(FORMAT(V1.[DATE_REGISTERED],'yyyyMMdd') as int) AS [FK_DATE] , ISNULL(DBO.UFN_CALC_TIME(V1.[DATE_REGISTERED],ISNULL(V2.[DATE_REGISTERED],V1.[DATE_REGISTERED])),0) / 3600.00 AS [HOURS_PASSED] , V1.[DATE_REGISTERED] AS [DATE_REGISTERED1] , V2.[DATE_REGISTERED] AS [DATE_REGISTERED2] , 0 AS [HOURS_REGISTERED] FROM (SELECT ROW_NUMBER() OVER (ORDER BY tH.TICKET_ID, TH.TICKET_HIST_ID) AS [ROW_HIST] , TH.TICKET_ID AS [FK_TICKET] , TH.STATE_ID AS [FK_STATE] , T.TYPE_ID AS [FK_TICKET_TYPE] , TH.OWNER_ID AS [FK_USER] , MIN(TH.CREATE_TIME) AS [DATE_REGISTERED] , TH.QUEUE_ID AS [FK_QUEUE] FROM [Thesis_Source].[dbo].[OTRS_TICKET_HISTORY] TH , [Thesis_Source].[dbo].[OTRS_TICKET] T WHERE 1 = 1 AND T.TICKET_ID = TH.TICKET_ID AND TH.CREATE_TIME IS NOT NULL GROUP BY TH.TICKET_ID , TH.STATE_ID , T.TYPE_ID , TH.OWNER_ID , TH.TICKET_HIST_ID , TH.QUEUE_ID) V1 LEFT JOIN (SELECT ROW_NUMBER() over (order by tH.TICKET_ID, TH.TICKET_HIST_ID) AS [ROW_HIST] , TH.TICKET_ID AS [FK_TICKET] , TH.STATE_ID AS [FK_STATE] , T.TYPE_ID AS [FK_TICKET_TYPE] , TH.OWNER_ID AS [FK_USER] , MIN(TH.CREATE_TIME) AS [DATE_REGISTERED] , TH.QUEUE_ID AS [FK_QUEUE] FROM [Thesis_Source].[dbo].[OTRS_TICKET_HISTORY] TH , [Thesis_Source].[dbo].[OTRS_TICKET] T WHERE 1 = 1 AND T.TICKET_ID = TH.TICKET_ID AND TH.CREATE_TIME IS NOT NULL GROUP BY TH.TICKET_ID , TH.STATE_ID , T.TYPE_ID , TH.OWNER_ID , TH.TICKET_HIST_ID , TH.QUEUE_ID) V2 ON (V1.[FK_TICKET] = V2.[FK_TICKET] AND V2.[ROW_HIST] = V1.[ROW_HIST]+1) WHERE 1=1 UNION ALL -- REGISTER THE HOURS IN EACH TICKET AT OTRS SELECT TA.ticket_id AS [FK_TICKET] , 98 AS [FK_STATE] , 98 , T.TYPE_ID , t.queue_id AS [FK_QUEUE] , TA.create_by AS [FK_USER] , CAST(FORMAT(ta.[create_time],'yyyyMMdd') as int) AS [FK_DATE] , 0 AS [HOURS_PASSED] , TA.create_time , NULL , ta.time_unit AS [HOURS_REGISTERED] FROM [Thesis_Source].[dbo].[OTRS_TICKET_TIME_ACCOUNTING] TA, [Thesis_Source].[dbo].[OTRS_TICKET] t WHERE 1=1 AND TA.ticket_id = t.ticket_id) V JOIN [Thesis_SA].[dbo].[STG_DIM_QUEUES] Q ON (V.[FK_QUEUE] = Q.BK_QUEUE AND CAST(V.[DATE_REGISTERED1] AS DATE) BETWEEN CAST(Q.REPORT_START_DATE AS DATE) AND CAST(Q.REPORT_END_DATE AS DATE)) LEFT JOIN [Thesis_SA].[dbo].[STG_DIM_CLIENTS] LC ON (LC.NAME_OTRS = Q.QUEUE_CLIENT_NAME) -- IDENTIFY THE EMPLOYEE ASSOCIATED WITH THE TICKET SO WE CAN GET THE VALUE HOUR ASSOCIATED AT THE TIME THE COLLABORATOR REGISTERED THE HOURS IN OTRS LEFT JOIN [THESIS_SA].[DBO].[STG_DIM_EMPLOYEES] EMP ON (EMP.OTRS_ID = V.[FK_USER] AND CAST(V.[DATE_REGISTERED1] AS DATE) BETWEEN CAST(EMP.EMPLOYEE_START_DATE AS DATE) AND CAST(EMP.EMPLOYEE_END_DATE AS DATE)) LEFT JOIN [Thesis_SA].[dbo].[STG_DIM_TICKET_TYPE] TT ON (v.FK_TICKET_TYPE = TT.BK_TYPE) LEFT JOIN [Thesis_SA].[dbo].[STG_DIM_TICKET] T ON (V.[FK_TICKET] = t.[BK_TICKET]) LEFT JOIN [Thesis_SA].[dbo].[STG_DIM_STATUS] ST ON (ST.STATE_ID2 = ISNULL(V.[FK_STATE2],CAST('99' AS INT)) AND V.[FK_STATE1] =ST.STATE_ID1) WHERE 1=1 and EMP.SK_EMPLOYEE is not null ;

Table A9 – SA STG_FACT_SMRS Data Flow

Data Flow Name: STG_FACT_SMRS	
Schema	**SQL Query**
	SELECT C.[BK_CLIENT] AS [FK_CLIENT] , SP.BK_PROJECTCODE AS [FK_PROJECT] , E.[BK_EMPLOYEE] AS [FK_EMPLOYEE] , CAST(FORMAT(sa.[ENTRYDATE],'yyyyMMdd') as int) AS [FK_DATE] , SA.activity_id AS [FK_ACTIVITY] , sA.[PROJECTCODE] , sa.[TOTALHRS] , sa.[TOTALHRS]*SP.[CONTRACT_COSTPERHOUR] AS [BILLED] FROM [Thesis_Source].[dbo].[SMRS_ACTIVITY] sa JOIN [THESIS_SA].[DBO].[STG_DIM_EMPLOYEES] E ON (E.EMPLOYEE_LOGIN = SA.ANALISTNAME AND SA.ENTRYDATE BETWEEN E.EMPLOYEE_START_DATE AND ISNULL(E.EMPLOYEE_END_DATE, GETDATE())) JOIN [THESIS_SA].[DBO].[STG_DIM_PROJECTS] SP ON (SA.PROJECTCODE = SP.BK_PROJECTCODE AND CAST(FORMAT(sa.[ENTRYDATE],'yyyyMMdd') as int) = SP.PROJECT_DATE_ID AND SP.[PROJECT_VALUE_TYPE] = 'Hours Register') JOIN [Thesis_SA].[dbo].[STG_DIM_CLIENTS] C ON (C.[NAME_PROJECT] = SP.CLIENT_NAME) WHERE 1=1

DW ETL

Control Flow

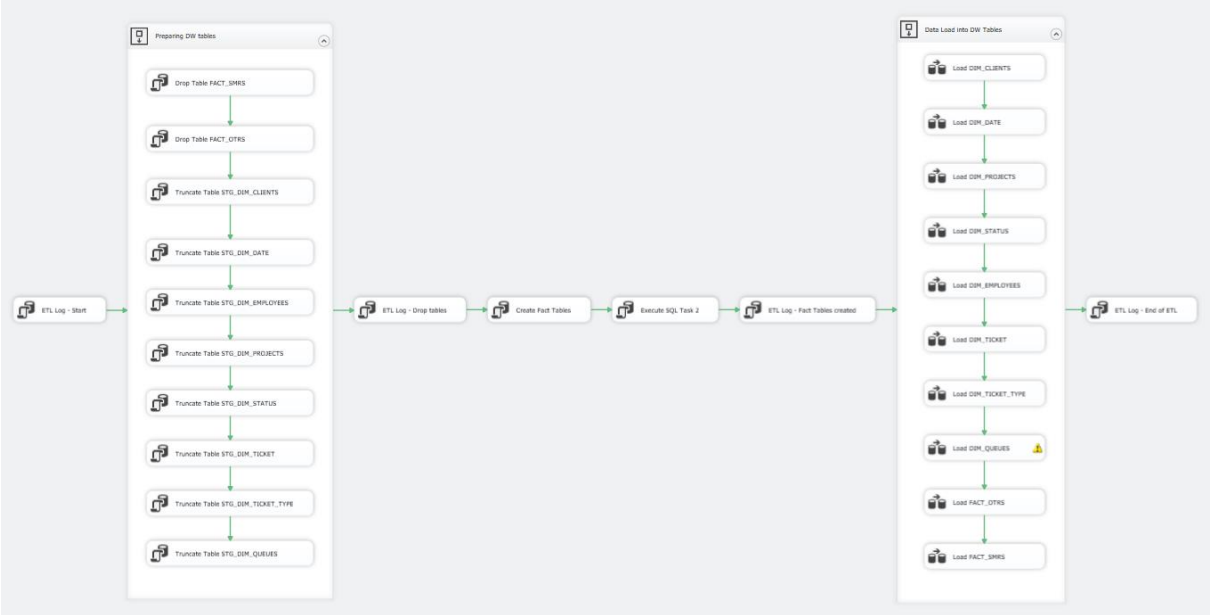


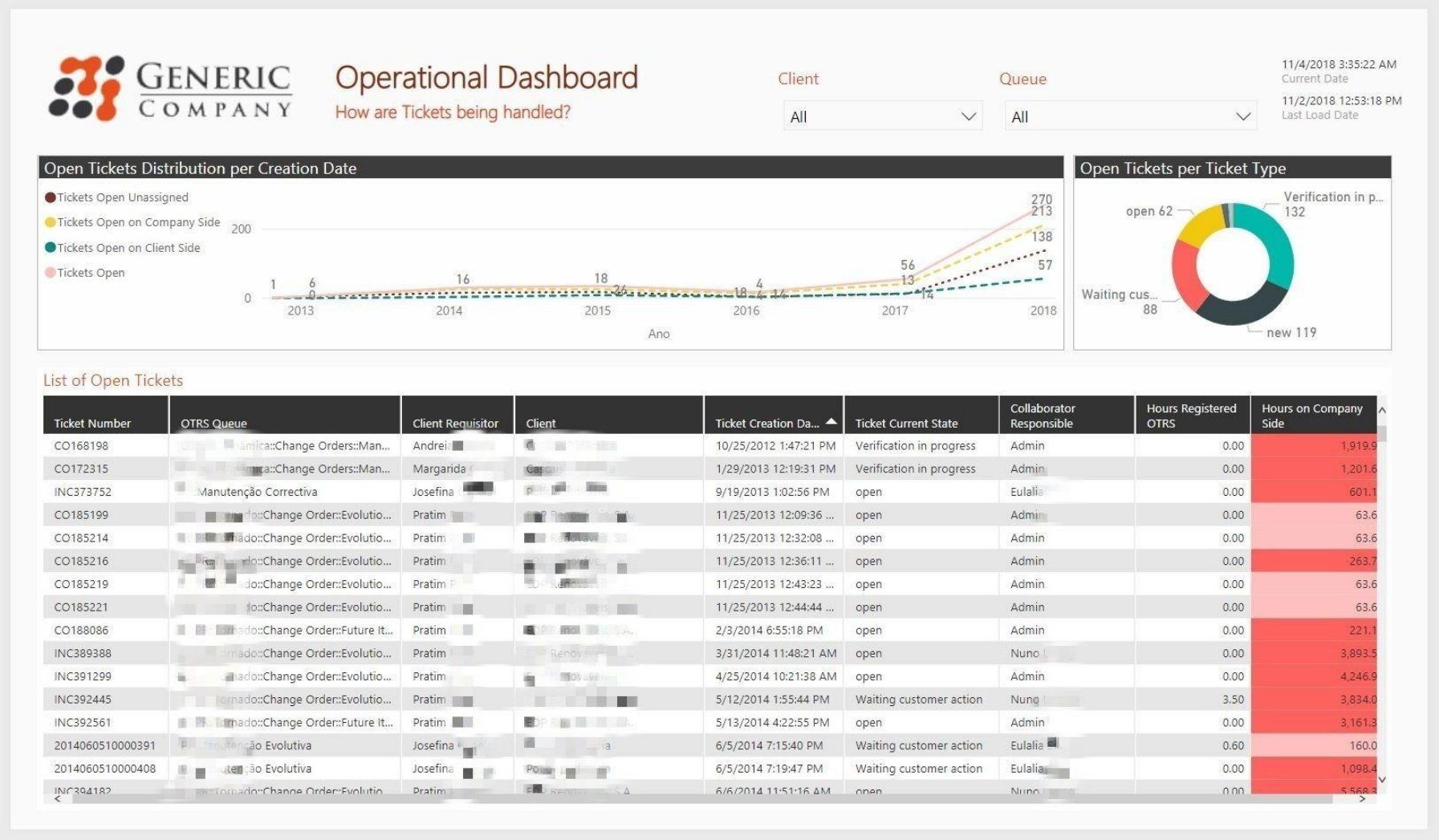
Figure A3 – Source ETL Control Flow

Data Flows

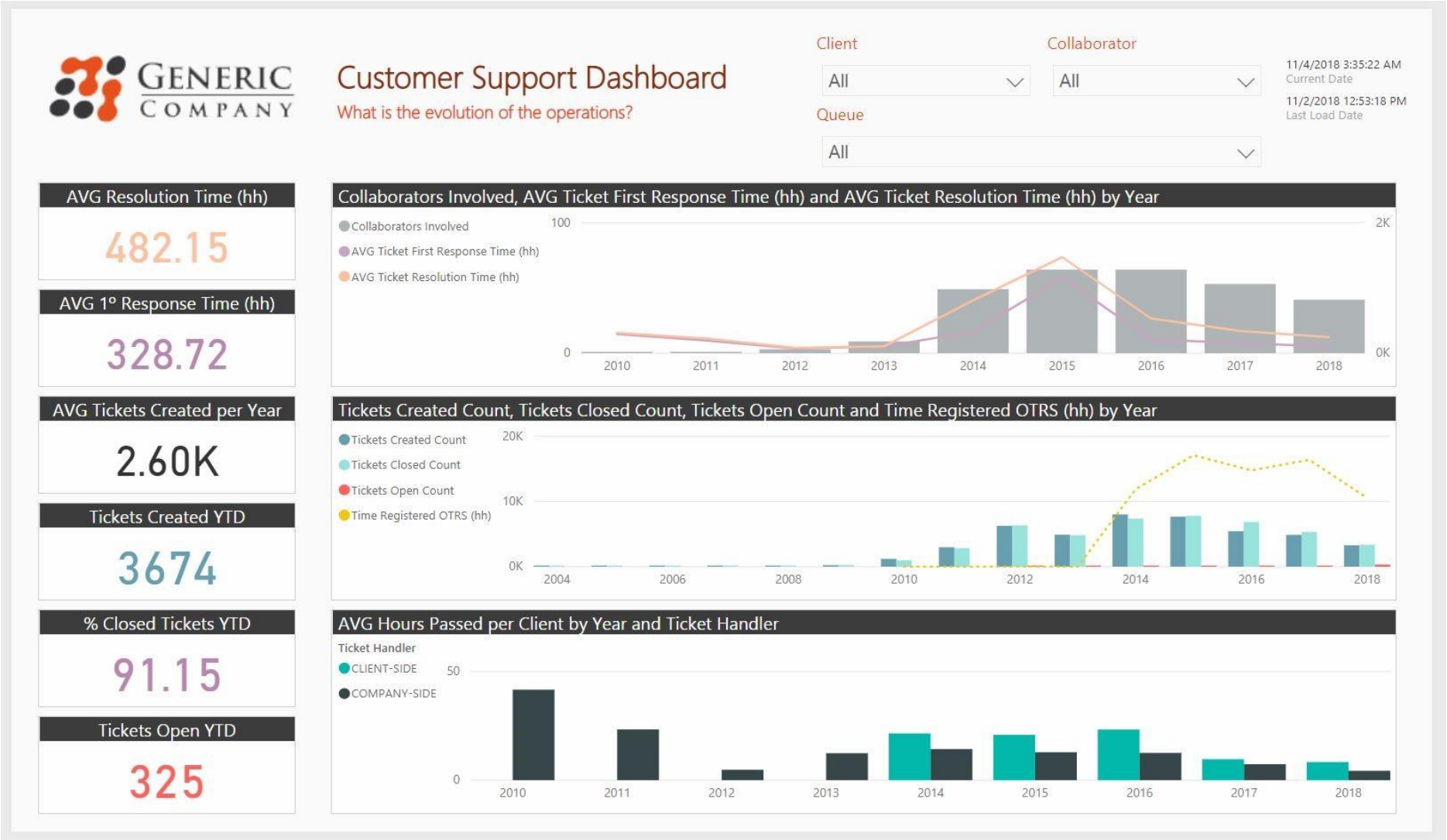
All the loadings are done directly from the Staging Area database to the Data Warehouse database without any transformations.

APPENDIX B

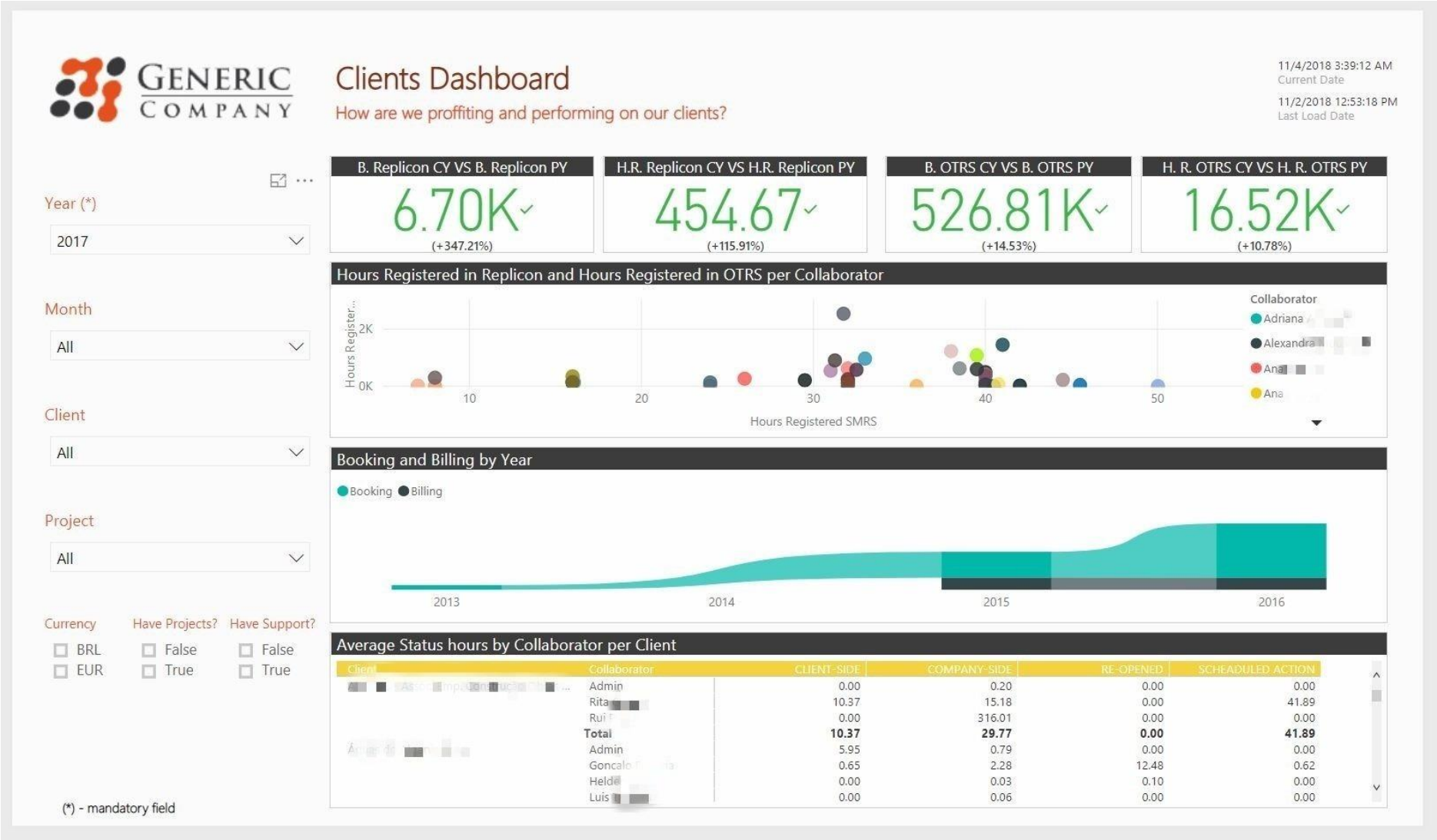
B.1 - Operational Dashboard



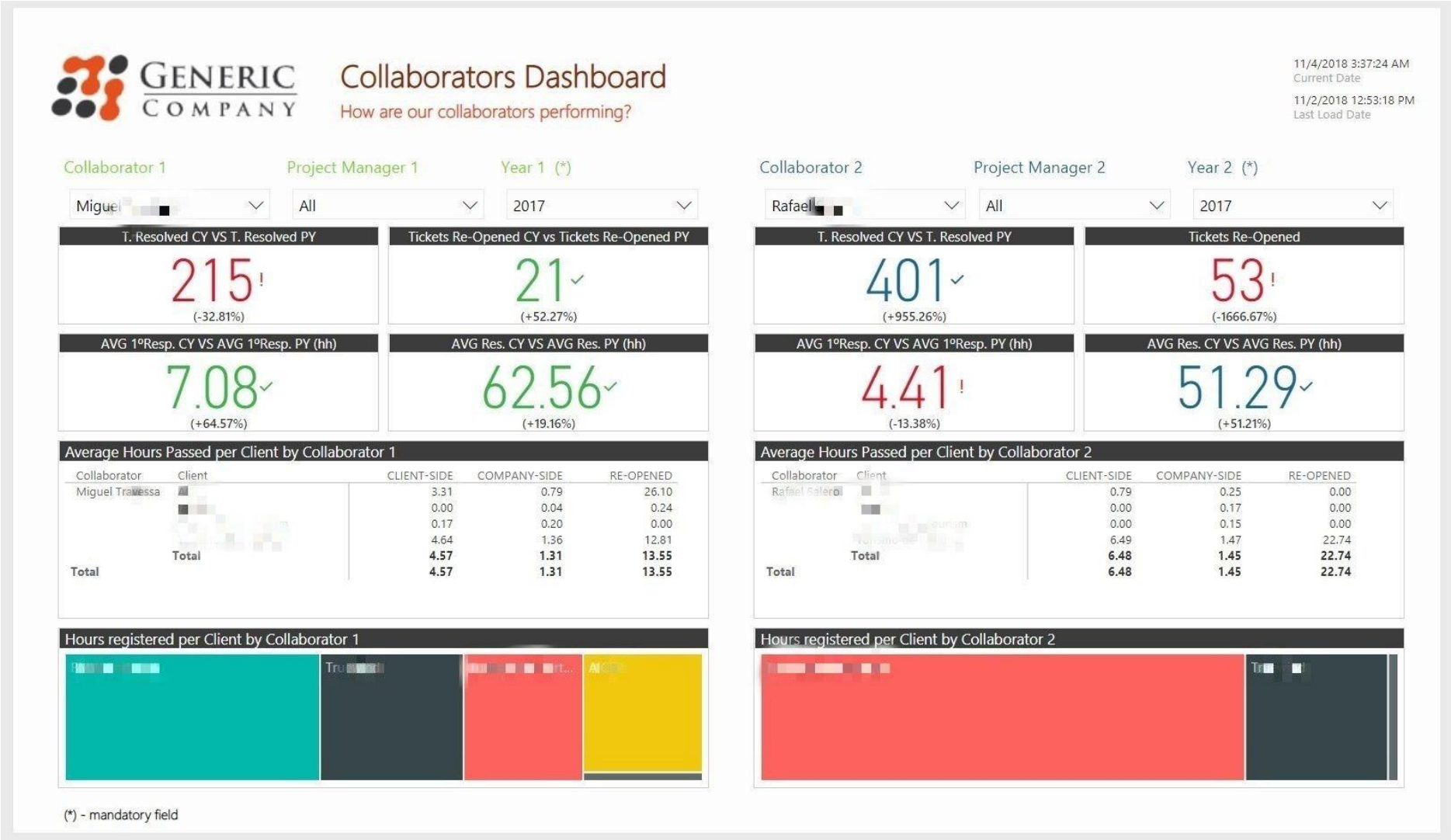
B.2 - Customer Support Dashboard



B.3 - Clients Dashboard



B.4 - Collaborators Dashboard



B.5 - Projects & Contracts Dashboard