



Case Report

# Business Intelligence's Self-Service Tools Evaluation

Jordina Orcajo Hernández <sup>1</sup> and Pau Fonseca i Casas <sup>2,\*</sup>

<sup>1</sup> BarcelonaTech School of Mathematics and Statistics, Universitat Politècnica de Catalunya, C. Pau Gargallo 5, 08028 Barcelona, Spain

<sup>2</sup> BarcelonaTech Statistics and Operations Research Department, Universitat Politècnica de Catalunya, Jordi Girona 3-1, 08034 Barcelona, Spain

\* Correspondence: pau@fib.upc.edu; Tel.: +34-934-01-7735

**Abstract:** The software selection process in the context of a big company is not an easy task. In the Business Intelligence area, this decision is critical, since the resources needed to implement the tool are huge and imply the participation of all organization actors. We propose to adopt the systemic quality model to perform a neutral comparison between four business intelligence self-service tools. To assess the quality, we consider eight characteristics and eighty-two metrics. We built a methodology to evaluate self-service BI tools, adapting the systemic quality model. As an example, we evaluated four tools that were selected from all business intelligence platforms, following a rigorous methodology. Through the assessment, we obtained two tools with the maximum quality level. To obtain the differences between them, we were more restrictive increasing the level of satisfaction. Finally, we got a unique tool with the maximum quality level, while the other one was rejected according to the rules established in the methodology. The methodology works well for this type of software, helping in the detailed analysis and neutral selection of the final software to be used for the implementation.

**Keywords:** business intelligence; self-service tools; systemic quality model; software selection



**Citation:** Orcajo Hernández, J.; Fonseca i Casas, P. Business Intelligence's Self-Service Tools Evaluation. *Technologies* **2022**, *10*, 92. <https://doi.org/10.3390/technologies10040092>

Academic Editor: Mohammed Mahmoud

Received: 11 July 2022

Accepted: 4 August 2022

Published: 10 August 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Business Intelligence (BI) is associated with a set of tools and techniques related to the transformation of raw data into meaningful and useful information for business analysis purposes [1,2]. BI technologies are capable of handling large amounts of unstructured data to help identify, develop and otherwise create new strategic business opportunities. One of the principal objectives of BI is to allow an easy interpretation of these large volumes of data. Specifically, self-service BI aims to improve the company's useful information use from their data. Self-service BI wants to allow workers to understand and analyze data without specialized expertise. In that sense, workers can make, faster and better decisions because the information is available and is not needed to wait for a specific reporting. Technical teams will be freed from the burden of satisfying end-user report requests, so they can focus their efforts on more strategic IT initiatives. There are many self-service BI tools in the market, and before recommending a particular one, an in-depth analysis of the available tools on the market should be conducted. The automation and systematization of the selection process of critical enterprise software such as enterprise resource planning (ERP) was studied by several authors (e.g., see [3]). Researchers attempted to rank several techniques and ERP alternatives in the process [4,5], as well as adapt existing methodologies using artificial neural networks to improve the decision process [6], use hybrid methodologies [7] or specify different scopes, for instance in the application to the management information system of a power plant [8] or supply chains [9]. In line with the definition of the BI tools, in [10] an in-depth analysis of existing challenges of business intelligence (BI) and a proposal for the new generation of tools are presented, with a focus on new data sources (e.g., social media) and including concepts like security and trust. In this context, social business intelligence requires integration with trusted external data [11].

Therefore, we need a method to be able to select or implement the appropriate information system that allows us to use this information in an appropriate manner in our organization.

In line with the implementation of an information system, an analysis of the problems related to its implementation and use can be reviewed in [12]. Moreover, some proposals for modeling information systems [13] and performing a functional safety assessment [14] can be useful for the modeling of the overall infrastructure. However, no attempts have been made to systematize the selection of BI tools in the context of a big corporation.

In software selection, the systemic quality model (SQMO) was proposed [15], providing successful implementation examples in other software areas (see [16,17]).

This work aims to build a comparative assessment of self-service BI tools, adapting a systemic quality model (SQMO) and applying the method to finally evaluate, in this case, four tools. Therefore, we focused on the development of a method that guides the selection process of BI tools. It must consider that the “best tool” concept is not applicable in this scope. For this reason, it is more usual to talk about an appropriate solution for a particular project.

## 2. BI Users

A rigorous evaluation should be conducted by several users to obtain trustworthy results. In particular, self-service BI tools, as data systems, usually have different user profiles and several users of each type should evaluate the tools from their particular point of view.

There are three different profiles of a user in data systems, according to [18].

**Farmers:** They access information predictably and repetitively. We could say that they have their parcel of information and they regularly cultivate and extract profit from this. They do not access a huge amount of data (because they do not leave the parcel) and they usually ask for aggregated data. These users usually use OLAP (online analytical processing) tools, which are focused on non-informatics users. They are simple and their main objective is data visualization. As farmers, there are employers, providers, and customers to whom the organization offers informational services. Currently, business intelligence, which promotes the use of these systems at all levels of the organization, allows business users to use data and information in business processes naturally, without having to leave their applications.

**Explorer:** Opposite to farmers, explorers have unpredictable and irregular access. They spend much time planning and preparing for their studies and when they have everything ready, they start to explore a lot of detailed information. They do not know exactly what they are looking for until they find it, and the results are not guaranteed in every case. However, sometimes they find something really interesting that improves the business. They are also known as power users. Thanks to big data, explorers have become data scientists. A data scientist must be able to extract information from large volumes of data according to a clear business objective and then present it in a simple way to non-expert users in the organization. Therefore, it consists of a cross profile with skills in computer science, mathematics, statistics, data mining, graphic design, data visualization, and usability.

**Tourists:** Typically, they entail a group of two or more people. On one side, there is a person with an overview of the company that comes up with the possibility of a study on a certain topic. On the other, there is a computer expert that knows the systems analysis of the company and is the manager who finds out if the study is feasible with the available data and tools. This team will access data without following any pattern and will rarely observe the same data twice. Therefore, their requirements cannot be known a priori. Tools used by tourists are browsers or search engines (to search both data and metadata) and the result of their work will be the projects carried out by *farmers* or *explorers*. In short, a tourist is a casual user of the information.

This project aims to develop a method of evaluation that should be applicable taking into consideration the different profiles of the tool. For example, if the tool will be used

by *farmers* and *explorers*, some *farmer* and *explorer* users should evaluate the tool. After this evaluation, a mean is done with the results. In this paper, to illustrate the methodology used, the evaluation by an *explorer* user is shown.

To carry out an assessment, several steps should be followed. First of all, the evaluator responsible for preparing the assessment has to know the subject and propose a methodology adapted to the specific scope. The adaption implies choosing a set of interesting metrics that will be used in the evaluation. Users can advise the evaluator about interesting metrics and the evaluator has to design a questionnaire to include them, the questionnaire is presented on Appendix D. Next, the evaluator must send a questionnaire to the users to collect the opinions from experts in the area. Moreover, the evaluator has to provide every item required to perform the evaluation (questionnaires, data, applications, etc.). Finally, the questionnaires are collected and the assessment proceeds in line with the chosen methodology to evaluate the results.

### 3. Methodology, the Systemic Quality Model (SQMO)

The systemic quality model (SQMO) was proposed in 2001 by [15]. The application of the SQMO for software evaluations provided successful implementation examples, see Figure 1.

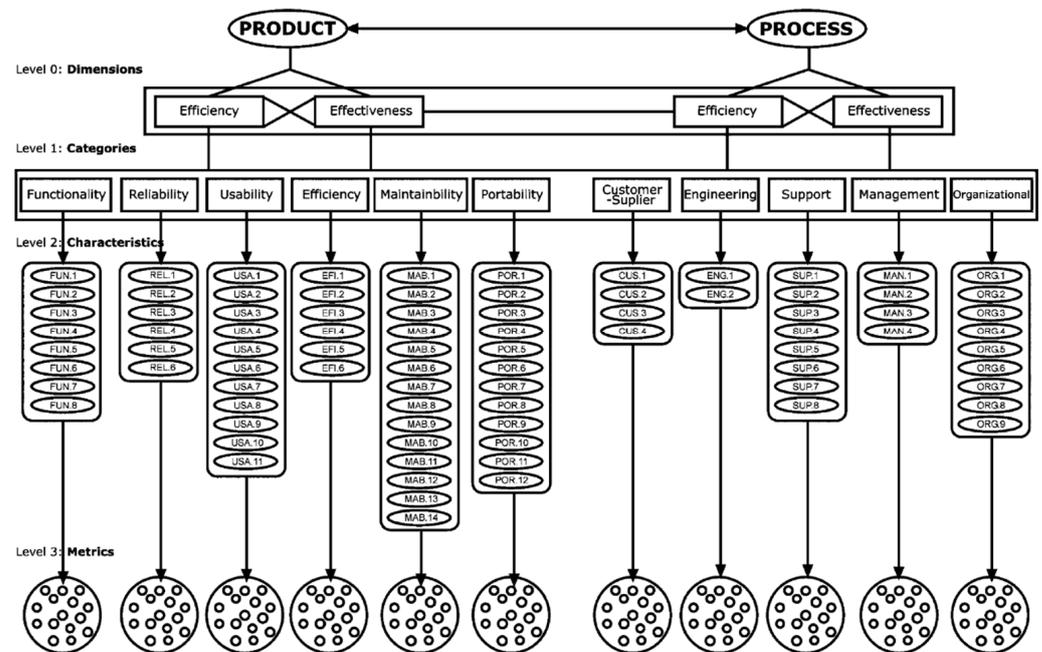


Figure 1. Diagram of the systemic quality model (SQMO) [16].

Until then, several models existed to evaluate product software and others to evaluate process software, but none with the capability to evaluate both aspects accurately. The SQMO can use either the *product* or the *process* sub-model or both. The first sub-model is designed to evaluate the developed software, while the second is designed to evaluate the development process of the software. From [15], the SQMO sub-models have different levels to assess software, see [16,17].

#### 3.1. Level 0: Dimensions

There are two dimensions for each sub-model: *efficiency* and *effectiveness* for the *product* and *efficiency* and *effectiveness* for the *process*. *Effectiveness* is the capability of producing the required result, while *efficiency* is the capability to produce a specific result effectively with a minimum amount or quantity of waste, expense, or unnecessary effort.

### 3.2. Level 1: Categories

There are six elements corresponding to *product* and five corresponding to *process*. The categories for the *product* sub-model are presented in Table 1.

**Table 1.** SQMO product sub-model categories [16].

Category	Definition
Functionality (FUN)	Functionality is the capacity of the software product to provide functions that meet specific and implicit needs when software is used under specific conditions
Reliability (FIA)	Reliability is the capacity of a software product to maintain a specified level of performance when used under specific conditions
Usability (USA)	Usability is the capacity of the software product to be attractive, understood, learned, and used by the user under certain specific conditions
Efficiency (EFI)	Efficiency is the capacity of a software product to provide appropriate performance, relative to the number of resources used, under stated conditions
Maintainability (MAB)	Maintainability is the capacity of the software to be modified. Modifications can include corrections, improvements, or adaptations of the software to adjust to changes in the environment, in terms of the functional requirements and specifications
Portability (POR)	Portability is the capacity of the software product to be transferred from one environment to another

The categories for the *process* sub-model are presented in Table 2.

**Table 2.** SQMO process sub-model categories [16].

Category	Definition
Client-supplier (CUS)	Is made up of processes that have an impact on the client, support the development and transition of the software to the client, and give the correct operation and use of the software product or service
Engineering (ENG)	Consists of processes that directly specify, implement, or maintain the software product, its relation to the system, and documentation on it
Support (SUP)	Consists of processes that can be used by any of the processes (including support ones) at several levels of the acquisition life cycle
Management (MAN)	Consists of processes that contain practices of a generic nature that can be used by anyone managing any kind of project or process, within a primary life cycle
Organizational (ORG)	Contain processes that establish the organization's commercial goals and develop process, product, and resource goods (value) that will help the organization attain the goals set in the projects

### 3.3. Level 2: Characteristics

SQMO specifies that each category consists of a set of characteristics, which define the more important characteristics and features that must be satisfied to assure the software *product* and/or *process* quality. *Product* characteristics are specified in Table 3. and process characteristics in Table 4. They are defined more accurately in [17].

**Table 3.** Characteristics for product sub-model.

Category	Characteristics	
	Product Effectiveness	Product Efficiency
Functionality	Fit to purpose	Correctness
	Precision	Structured
	Interoperability	Encapsulated
	Security	Specified
Reliability	Maturity	Correctness
	Fault tolerance	Structured
	Recovery	Encapsulated

Table 3. Cont.

Category	Characteristics	
	Product Effectiveness	Product Efficiency
Usability	Ease of understanding	Complete
	Ease of learning	Consistent
	Graphical Interface	Effective
	Operability	Specified
	Conformity of standards	Documented
		Auto-descriptive
Efficiency	Execution performance	Effective
	Resource utilization	No redundant
		Direct
		Used
Maintainability	Analysis Capability	Attachment
	Ease of changing	Cohesion
	Stability	Encapsulated
	Testability	Software maturity
		Structure information
		Descriptive
		Correctness
		Structural
		Modularity
Portability	Adaptability	Consistent
	Installation capability	Parameterized
	Co-existence	Encapsulated
	Replacement capability	Cohesive
		Specified
		Documented
		Auto-descriptive
		No redundant
	Auditing	
		Quality management
Data Quality -both dimensions-		

Table 4. Characteristics for the process sub-model.

Category	Characteristics	
	Process Effectiveness	Process Efficiency
Customer–Supplier	Acquisition system or software product	Supply
	Requirement determination	Operation
Engineering	Development	Maintenance of software and systems Principio del formulario

Table 4. Cont.

Category	Characteristics	
	Process Effectiveness	Process Efficiency
Support	Quality assurance	Documentation
	Joint review	Configuration management
	Auditing	Verification
	Solving problems	Validation
		Joint review
Management		Auditing
		Solving problems
	Management	Management
	Quality management	Project management
	Risk management	Quality management
Organizational		Risk management
	Organizational alignment	Establishment of the process
	Management of change	Process evaluation
	Process improvement	Process improvement
	Measurement	HHRR management
	Reuse	Infrastructure

### 3.4. Level 3: Metrics

Each characteristic consists of a group of metrics to be evaluated. They are the evaluable attributes of the product and the process, and they are not agreed upon because they vary depending on each study case. Metrics are detailed in Appendix “Appendix A. The Metrics Used in the Selection Process”.

### 3.5. Algorithm

The algorithm to measure the systematic quality by the SQMO, referenced in [15] is the following explained. First of all, the product software is measured, and then the development process.

### 3.6. Product Software

The first measured category must be always *functionality*. If the product does not meet the *functionality* category, the evaluation is ended. It is because the functional category identifies the software capability to fit the purpose for what it was built.

After that, a sub-model is adapted depending on the requirements. The algorithm suggests working with a maximum of three characteristics of the product (including *functionality*) because if more than three product features are selected, some might conflict. In this sense, [19] indicates that the satisfaction of quality attributes can have an effect, sometimes positive and sometimes negative, on meeting other quality attributes. The definition of satisfaction can vary depending on the case of use and it is not fixed by the methodology. In Section 6, this issue is discussed.

Finally, to measure the quality product of the software, Table 5 shows the quality levels related to the satisfied categories.

Once the evaluation of the product software has ended, recalling that only if the quality level is at least *basic*, the *development process* evaluation may start.

**Table 5.** Quality levels for the product software.

Functionality	Second Category	Third Category	Quality Level
Satisfied	No satisfied	No satisfied	Basic
Satisfied	Satisfied	No satisfied	Medium
Satisfied	No satisfied	Satisfied	Medium
Satisfied	Satisfied	Satisfied	Advanced

### 3.7. Development Process

To evaluate the *development process* there are four steps to follow. The algorithm used in the *development process* evaluation is fixed, unlike the *product software* evaluation. The steps are as follows: (i) determining the percentage of N/A (not applying) answers in the questionnaire for each category. If this percentage is greater than 11%, the application of the measuring instrument must be analyzed, and the algorithm stops. Otherwise, we continue with step 2; (ii) determining the percentage of N/K (not knowing) answers in the questionnaire for each category. If this percentage is greater than 15%, it shows that there is a high level of ignorance of the activities of the particular category. If the percentage is lower, we continue with step 3; (iii) determining the satisfaction level for each category (the definition of satisfaction can vary depending on the case of use and it is not fixed by the methodology; in Section 6, this issue is discussed); (iv) measuring the quality level of the process. The quality levels related to the satisfied categories in Table 6 are:

- **Basic level:** It is the minimum required level. Categories *customer-supplier* and *engineering* are satisfied.
- **Medium level:** In addition to the basic level categories satisfied, categories *support* and *management* are satisfied.
- **Advanced level:** All categories are satisfied.

**Table 6.** Quality levels for development process.

Quality Levels		Category Satisfied
Advanced	Medium	Basic
		Customer–supplier
		Engineering
		Support
		Management
		Organizational

Finally, there must be a joint between the product quality measuring and the process quality measuring, to obtain systematic quality measuring. The systemic quality levels are proposed in Table 7.

**Table 7.** Systemic quality levels.

Product Quality Level	Process Quality Level	Systemic Quality Level
Basic	-	Null
Basic	Basic	Basic
Medium	-	Null
Medium	Basic	Basic
Advanced	-	Null
Advanced	Basic	Medium
Basic	Medium	Basic
Medium	Medium	Medium
Advanced	Medium	Medium
Basic	Advanced	Medium
Medium	Advanced	Medium
Advanced	Advanced	Advanced

This method of measurement is responsible for maintaining a balance between the sub-models (when they are both included in the model).

#### 4. Adoption of the Systemic Quality Model (SQMO)

SQMO was selected as a reference because it is a complete approach influenced by many other models. First of all, it respects the concept of systemic total quality from [20]. It also considers the balance between the *process* and *product* sub-models proposed by [21]. These sub-models are based on the *product* and *process* quality models from [22] and [23], respectively. Moreover, the product quality categories are based on the work of [24] and the international standard ISO/IEC 9126 (JTC 1/SC 7, 1991). The process categories are extracted from the international standard ISO/IEC 15504 (ISO IEC/TR 15504-2, 1998).

Some authors [25] have pointed out that when characteristics are complex, they can be divided into a simpler set and a new level for sub-characteristics can be created. In this particular case, sub-characteristics were considered to gain clarity. To adapt the SQMO to each particular case, it should be decided which sub-model will be considered (*product*, *process*, or both), as well as which dimension (*efficiency* or /and *effectiveness*), which sub-characteristics, and which respective metrics. In the current evaluation, only the *product* sub-model of SQMO was considered. The *process* sub-model is excluded because we intend to evaluate the fully developed tools as future tools useful for the BI workforce. Moreover, only the *effectiveness* dimension is considered because special attention is focused on the evaluation of features observed during the execution. However, if one considers including the sub-model *process* or the *efficiency* dimension, there is an option to do so by following the steps explained above. Figure 2 reflects the adapted model used in the current evaluation.

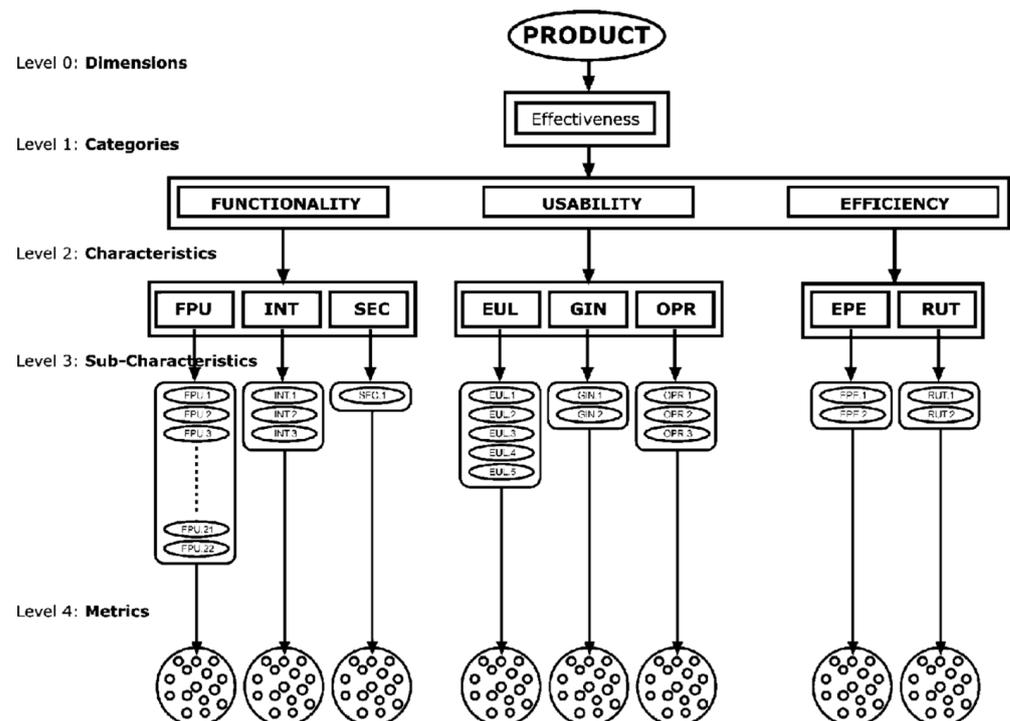


Figure 2. Diagram of the adapted systemic quality model [16].

Besides the *functionality* category, we choose *usability* because this type of tool (self-service BI tool) is focused on non-technical users and the difficulty of the product should be minimal. Moreover, it must be an attractive product because the success of the tool depends on the user's satisfaction. Finally, the *efficiency* category was chosen because the processor type, the hard disk space, and the minimum RAM required are all factors that determine the success of the tool's deployment. Self-service BI tools are popular thanks to their "working memory". Then, it is important to evaluate the minimum amount of memory required.

## 5. Scales of Measurement

In the current evaluation, all the evaluated metrics are ordinal variables because they have more than two categories and they can be ordered or ranked (see annex Appendix A. The Metrics Used in the Selection Process There are different types of scale measurement depending on the metric.

### *Type A of Scale Measurement*

The following scale measures the metrics with a scale from 0 to 4 as follows:

- **0:** The application does not have the feature.
- **1:** The application matches the feature poorly or it does not strictly match the feature but it can obtain similar results.
- **2:** The application has the feature and matches the expectations, although it needs an extra corporative complement. This mark should also be assigned when the feature implies a manual job (e.g., typing code, clicking a button) and the metric requires an automatic job.
- **3:** The application has the feature and matches the expectations successfully without a complement.
- **4:** The application has the feature and presents advantages over others.

Even so, other metrics need to be measured specifically. Sub-type A.1 of scale measurement is assigned to binary metrics: We assign 0 values if the application does not have the feature, and 4 values if the application has it. We chose these values to be consistent with the rest of the measurement scales. Sub-type A.2 of scale measurement is assigned when the metric is measurable; we assign 4 to the application with a better result and a lower score than the others. As there are 4 values, the scale is from 4 to 1. Although, if some applications have the same value for a metric, the same score has to be assigned to them. To clarify the current scale measurement, we present an example of the metric *compilation speed* (see annex Appendix A.3. Efficiency Category). The *compilation speed* is measured with a scale from 1 to 4. We assign 1 value to the tools that require more time to compile, and 4 to the tool that requires a shorter time.

The official SQMO method involves a balance between all the characteristics because they have the same level of importance. However, sometimes, the user wants to give more importance to certain characteristics depending on his interests, and for that, we provided the following alternative, also used as a variant of SQMO. This alternative consists of assigning weights to the metrics. Therefore, the importance level of the metrics varies. We remark that weights must depend on each evaluation. However, we tried to assign weights generalizing, and based on our own experience, the weights were assigned considering the stakeholders of the company and experts' knowledge and information. Recalling that if the methodology is implemented in another use case, it can be modified. The used weights scale is the following:

- **0:** Not applicable to the organization.
- **1:** Possible feature or wish list item.
- **2:** Desired feature.
- **3:** Required or must-have feature.

Finally, final scores for sub-characteristics are computed using the weights assigned to the metrics. The final score of a sub-characteristic corresponds to the following formula:

$$score_{sub-characteristic\ i} = \frac{\sum_1^n v_j \times w_j}{\sum_j^n w_j} \quad (1)$$

where  $v_j$  is the value for the score assigned to metric  $j$ , while  $w_j$  is the weight for the corresponding metric. Moreover,  $n$  corresponds to the number of metrics in the sub-characteristic  $i$ . This adaption is applied when the importance level of the metrics is not the same for all

metrics (see Table 8 as an example). In this way, we got a score for each sub-characteristic, considering the weights of metrics.

**Table 8.** Weights of metrics. The complete list is in the Appendix A.

Metric	Weight
Excel files	3
Plain text	3
Connecting to different data sources at the same time	2
Allow renaming fields	3
R connection	2
Geographic information	2
...	

## 6. The Concept of Satisfaction

The term satisfaction can vary depending on the case of use. The evaluator can assign a limit, for example, 50%, and a sentence that a feature is satisfied if its score is higher than 50% of the maximum score on the measuring scale. For example, as our metric measuring scale is from 0 to 4, a score is satisfactory if it is higher than 2. However, the evaluator can also sentence the limit to 3 and in this way, a score is satisfactory if it is higher than 3. Usually, assessments are done to determine which tools are better than others, supposing that all the evaluated tools satisfy the main parts of the features. When the evaluator is looking for a distinction between tools, this type of limit can be useful. This concept applies to our units of measurement, which are metrics, sub-characteristics, characteristics, and categories. Once the metrics are evaluated with their respective scales of measurement (A, A.1, A.2), the methodology used to determine the satisfaction score is as follows: metrics scores are normalized with a percentage. A metric is satisfied if its percentage score is higher or equal to the fixed limit (satisfaction limit). Sub-characteristics are measured by the number of metrics satisfied (satisfaction score). Then, a particular sub-characteristic is satisfied if the number of satisfied metrics is higher or equal to its fixed limit (satisfaction limit). As weights are added, the satisfaction score becomes as Equation (1), where

$$v_j = \begin{cases} 1, & \text{if the metric } j \text{ is satisfied,} \\ 0, & \text{if the metric } j \text{ is not satisfied} \end{cases}$$

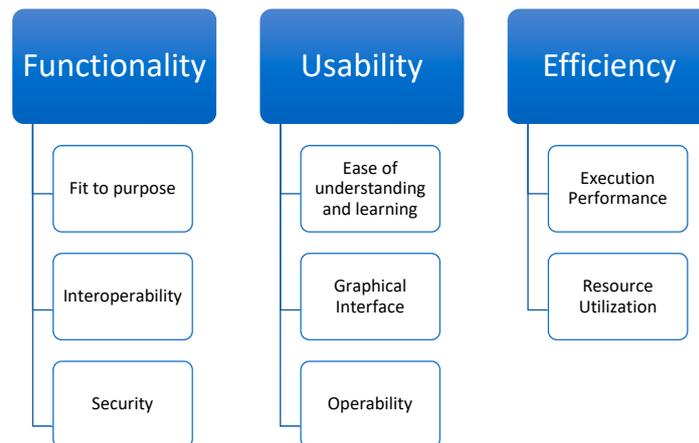
and characteristics are measured by the number of satisfied sub-characteristics (satisfaction score). Then, a particular characteristic is satisfied if the amount of satisfied sub-characteristics is higher or equal to its fixed limit (satisfaction limit). Categories are measured by the number of satisfied characteristics (satisfaction score). Then, a particular category is satisfied if the number of satisfied characteristics is higher or equal than its fixed limit (satisfaction limit). In the current evaluation, we decide to use the following limits, to get distinctions between tools, see Table 9. The evaluator can decide to modify the levels, to find distinctions between tools, or to be more restrictive or unrestrictive.

**Table 9.** Satisfaction limits.

Limit for metric	50%
Limit for sub-characteristic	50%
Limit for characteristic	75%
Limit for category	75%

### *Sub-Characteristics and Metrics for Self-Service BI Tools Evaluation*

In an evaluation, the most key step is to decide which characteristics must be evaluated. According to the SQMO schema, these characteristics are already agreed upon, but we have to establish the metrics related to each characteristic, see Figure 3.



**Figure 3.** Characteristic schema for each category, according to [16].

With our experience in the BI department and after working with these types of tools, we feel confident to decide which particular topics should be checked from self-service BI software. For each of the three evaluable characteristics, the sub-characteristics are listed in Tables 10–12, their respective metrics can be found in the appendix “Appendix A. The Metrics Used in the Selection Process”.

**Table 10.** Sub-characteristics for the functionality category, according to [17].

Functionality Category		
Fit for purpose	Interoperability	Security
Data loading	Languages	Security devices
Data model	Use project by third parts	
Fields relations	Languages	
Analysis	Data exchange	
Dashboard		
Reporting		

**Table 11.** Sub-characteristics for the usability category, according to [17].

Usability Category		
Ease of understanding and learning	Graphical interface	Operability
Learning time	Windows and mouse interface	Versatility
Browsing facilities	Display	
Terminology		
Help and documentation		
Support and training		

**Table 12.** Sub-characteristics for the efficiency category, according to [17].

Efficiency Category	
Execution performance	Resource utilization
Compilation speed	Hardware requirements
	Software requirements

## 7. Software Selection for the Evaluation

Before an evaluation, there must be a detailed selection of software that can be evaluated with the current evaluation model. Firstly, the area of application and the expected use of the software should be pre-established. The selection of software depends on this aspect because not every software is appropriate for every area. If the area of application

is pre-established, the selected software will be according to it. Secondly, a new level of depth should be considered with more specifications about the tool functionality. It should consider the features that make the tool useful for what we want to do.

Finally, it is needed to perform the identification of the required attributes based on the particular aims of the organization that will use the tool. Some of these attributes must be mandatory and others must be non-mandatory. Mandatory attributes are those that must be met by the selected software, while non-mandatory attributes are those that will be evaluated, which are the metrics. This aspect takes a key role in the selection and in the evaluation.

### Algorithm

Nowadays, there are many applications in the market related to business intelligence and because of that, deciding which applications should be included in an evaluation is a laborious task. Here we follow the methodology for selecting software proposed by Le Blanc [26]. In the first place, a long list of BI tools is elaborated (area of application). The next step is to reduce this to a medium list containing only tools that accomplish critical capabilities for business intelligence and analytics (the features that make the tool useful). Finally, a short list provided with the particular aims of the organization is built (required attributes).

The particular area of application is business intelligence. There are many platforms specialized in this area in the market. In this first step, we use Gartner as a data source for all business intelligence and analytics platforms in the market. Each year it edits and updates the report and inclusion criteria change depending on how the market changes, so it is a reference company. Therefore, we focus on those that have been mentioned in the report from Gartner Magic Quadrant for Business Intelligence and Analytics Platforms [27]. In this way, all the tools mentioned in the Magic Quadrant report of February 2015 (although Gartner, finally, has not evaluated them) compose the long list of sixty-three different platforms, which is shown in Table 13.

**Table 13.** The long list.

Adaptive Insights	Birst	DataRPM	FICO	Jedox	Oracle	Salesforce
Advizor Solutions	Bitam	Datawatch	GoodData	Kofax(Altosoft)	Palantir Technologies	Salient Management Company
AFS Technologies	Board International	Decisyon	IBM Cognos	L-3	Panorama	SAP
Alteryx	Centrifuge Systems	Dimensional Insight	iDashboards	LavaStorm Analytics	Pentaho	SAS (SAS Business Analytics)
Antivia	Chartio	Domo	Incorta	Logi Analytics	Platfora	Sisense
Arcplan	ClearStory Data	Dundas Data Visualization	InetSoft	Microsoft BI	Prognoz	Splunk
Automated Insgihts	DataHero	Eligotech	Infor	MicroStrategy.	Pyramid Analytics	Strategy Comapnio
BeyondCore	Datameer	eQ Technologic	Information Builder	Open Text (Actuate)	Qlik	SynerScope
Tableau	Targit	ThoughtSpot	Tibco Software	Yellowfin	Zoomdata	Zucche

To build the medium list we also base our selection on Gartner, in the Magic Quadrant report, where they choose the platforms to be evaluated if they satisfied particular capabilities that Gartner deems are critical to every business intelligence and analytics platform. In the Magic Quadrant report, Gartner chooses the platforms that satisfy 13 technique features and 3 non-techniques and they were classified into three categories: *enable*, *produce* and *consume*.

For *enable*, these features include:

- **Functionality and modeling:** Diverse source combination and analytical models' creation of user-defined measures, sets, groups, and hierarchies. Advanced capabilities can include semantic auto discovery, intelligent profiling, intelligent joins, data lineage, hierarchy generation, and data blending from varied data sources, including multi-structured data.
- **Internal platform integration:** To achieve a common look and feel, and install, query engine, shared metadata, and promo ability across all the components of the platform.
- **BI platform administration:** Capabilities that enable securing and administering users, scaling the platform, optimizing performance, and ensuring high availability and disaster recovery.
- **Metadata management:** Tools for enabling users to control the same systems-of-record semantic model and metadata. They should provide a robust and centralized way for administrators to search, capture, store, reuse, and publish metadata objects, such as dimensions, hierarchies, measures, performance metrics/KPIs, and report layout objects.
- **Cloud deployment:** Platform as a service and analytic application as service capabilities for building, deploying, and managing analytics in the cloud.
- **Development and integration:** The platform should provide a set of visual tools, programmatic and a development workbench for building dashboards, reports and also queries, and analysis.

For *produce*, these features include:

- **Free-form interactive exploration:** Enables the exploration of data through the manipulation of chart images; it must allow changing the color, brightness, size, and shape, and allow to include the motion of visual objects representing aspects of the dataset being analyzed.
- **Analytic dashboards and content:** The ability to create highly interactive dashboards and content with possibilities for visual exploration. Moreover, the inclusion of geospatial analytics to be consumed by others.
- **IT-developed reporting and dashboards:** Provides the capability to create highly formatted, print-ready, and interactive reports, with or without a previous parametrization. This includes the ability to publish multi objects, linked reports, and parameters with intuitive and interactive displays.
- **Traditional styles of analysis:** Ad hoc query that allows users to build their data queries, without relying on IT, to create a report. Specifically, the tools must have a reusable semantic layer that enables users to navigate available data sources, predefined metrics, hierarchies, and so on.

For *consume*, these features include:

- **Mobile:** Enables organizations in the development of mobile content and delivers it in a publishing and/or interactive mode.
- **Collaboration and social integration:** Enables users to share information, analysis, analytic content, and decisions via discussion threads, chat annotations, and storytelling.
- **Embedded BI:** Resources for modifying and creating analytic content, visualizations, and applications. Resources for embedding this analytic content into a business process and/or an application or portal.

Moreover, platforms had met other non-technical criteria. Generating at least \$20 million in total BI-related software license revenue annually, or at least \$17 million in total BI-related software license revenue annually, plus 15% year-over-year in new license growth. For vendors that also supply more transactional applications, it is necessary to analyze if its BI platform is used regularly by organizations that do not use its other transactional applications. Had a minimum of 35 customer survey responses from companies that use the vendor's BI platform in production.

With these added non-technical features, Gartner guarantees that at least 35 companies use each one of the tools. Moreover, it guarantees that companies that are growing

year-over-year use these tools. The medium list obtained was composed of 24 platforms (see Table 14). Notice that this can change depending on the time of the analysis and the specific needs of the company.

**Table 14.** Medium list.

Alteryx	Information Builder	Panorama	SAP (SAP Lumira)
Birst	Logi Analytics	Pentaho	SAS (SAS Business Analytics)
Board International	Microsoft BI	Prognoz	Tableau
Datawatch	MicroStrategy. (MicroStrategy Visual Insight)	Pyramid Analytics	Targit
GoodData	Open Text (Actuate)	Qlik (QlikView)	Tibco Software
IBM Cognos	Oracle	Salient Management Company	Yellowfin

Finally, to build the short list we focus on the particular aims of our organization. The particular tools that we want to evaluate are self-service BI tools and which means that the business user should be able to analyze the information he wants and build his reports. In traditional tools, the user asks a technical team for the information he needs, and he orders how information has to be displayed the technical team prepares data and built the ordered reports. Against that, self-service tools are being imposed on others because the working methodology is changing from being driven by the business model to being driven by the data model. There are six [6] features that characterize the particular aims of the organization: *ease of use, ability to incorporate data sources, “intelligence” to interpret data models, analysis functions, integration with corporative systems, and support.*

*Ease of use:* These tools are designed to be used by non-technical people. It means that users do not need to spend much time learning how the tool works before doing basic analysis.

*Ability to incorporate data sources,* both corporative databases (Oracle, SAP, etc.) local information (basically Microsoft Excel<sup>®</sup>), and external databases (Twitter, etc.).

*“Intelligence” to interpret correctly data models.* As they are auto-service tools and they face many types of data models, without previous modeling by a technical team, the interpretation of the model from the tool must be the correct one. If it is not the correct one, it can be misleading. How easy is to discover that the data model is wrong and how easy is to arrange the data model, are also important points to consider.

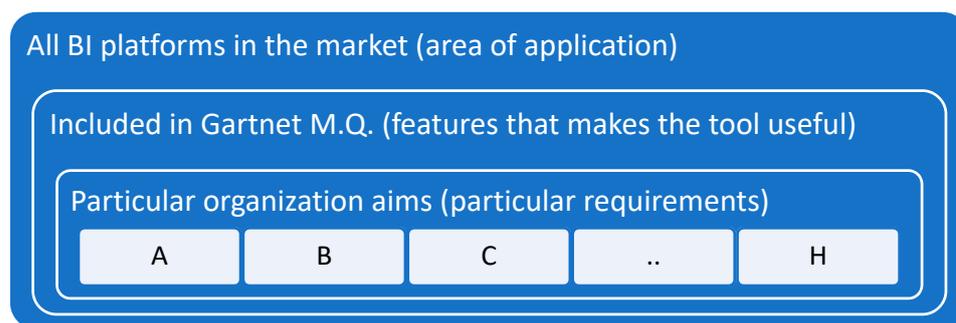
*Analysis functions:* Besides the typical pie and bar graphs, they must incorporate other tools to get advanced analysis (integration in R, statistic routines . . . ) always remembering the easy use.

*Possible integration with corporative systems and efficiency:* Usually, the user will work with a huge volume of data and therefore the analysis cannot be on a local PC. Tools should have the option of a central server that accesses data and process them. Big companies need security when the server is incorporated into the corporative environment. Then, the role of an administrator in managing the user’s access is key for big companies.

*Support:* In the case of an open-source tool being included in the larger list, it will not be considered in the medium list if it cannot offer instant customer support.

## 8. The Evaluated Software, the Short List

Finally, the short list is composed of eight platforms that can be evaluated with the adapted SQMO and they are nicked as software A, B, C, D, E, F, G, and H. See Figure 4 for a description of the process of list creation.



**Figure 4.** Schema for the selection process.

We describe next the evaluation of the four first tools from the short list, A, B, C, and D. For confidentiality reasons we are not going to provide the name of the short list, however, this does not have any impact on the description of the methodology used.

#### *Data Used*

To use and evaluate the applications, we needed a set of data and we decided to simulate it. The data set was simulated using R language and it was constructed by doing an emulation of a car insurance company database and using a relational structure. The structure of the dataset used can be consulted in Appendix C. The use of simulated data helps us in the testing of extreme cases.

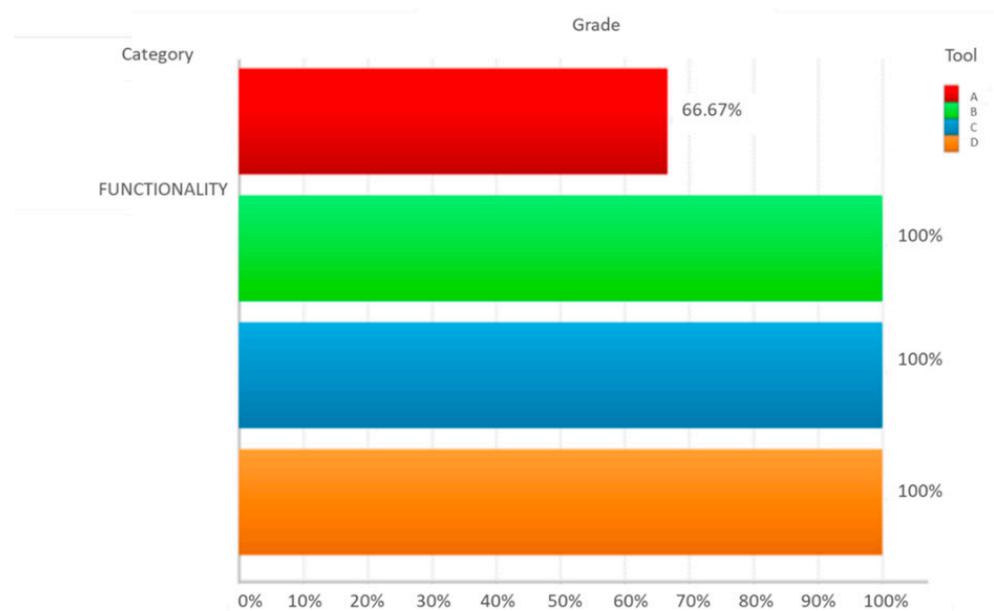
### 9. Evaluation Results

Once metrics are chosen, weights are assigned to each metric, applications are selected and data are available, it is time to carry out the evaluation. The evaluation shown here is done only by one *explorer* user. However, an evaluation should be done by several users, representing all the different types of users. In particular, self-service BI tools, such as data systems, usually have different user profiles [28]. The same amount of each type of user should evaluate the tools, from their particular point of view. From the operative point of view, to store the scores, an excel sheet with the 82 metrics is built. It is where users complete the cells with the score for each one of the metrics. The sheet is built considering the weights, see annex Appendix B. Metrics Weights and the satisfaction scores (see Section 6). The sheet is replicated identically assigning a sheet to each application. Therefore, a total of four excel sheets are filled by users, see Appendix C. 0141220\_Initial\_Test. With the evaluation sheets, the user must score the metrics for the selected applications. Scoring the metrics is the key step to getting results about each of the applications in each of the three categories: *functionality*, *usability*, and *efficiency*. The four sheets, one for each application, and the same database must be offered to each of the users.

#### *Results*

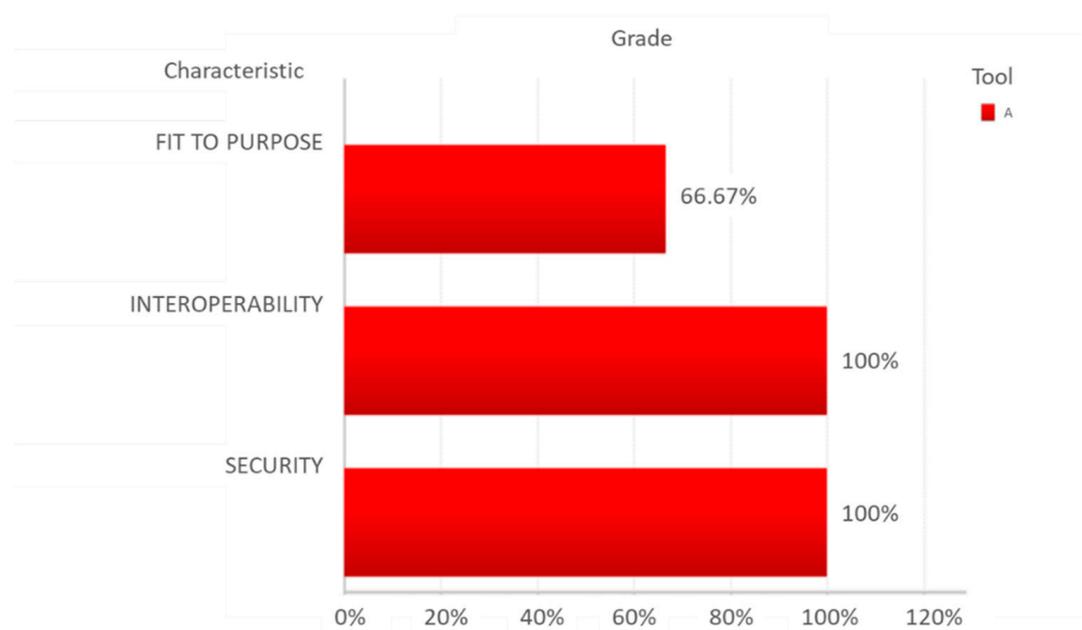
Once time every metric has been evaluated it is time to get the results of the assessment. In the usual case that more than one user is being implied in the evaluation of the metrics, we recommend calculating a mean score for each metric. On the other hand, one of the bases of the methodology [17] is that if the *functionality* category is not satisfied, the evaluation is aborted and other categories are not evaluated. Because of that, the analysis starts with the satisfaction score of the *functionality* category. In the current evaluation, using the satisfaction limits mentioned in Table 9, the obtained satisfaction scores for *functionality* are shown in Figure 5.

With the adaption of the methodology, we sentence that a category is satisfied if 75% of its characteristics are satisfied. Applying that software, A does not satisfy the *functionality* category because it only satisfies 66.67% of the *functionality* characteristics. Then, the evaluation of software A is aborted.



**Figure 5.** Results for the functionality category.

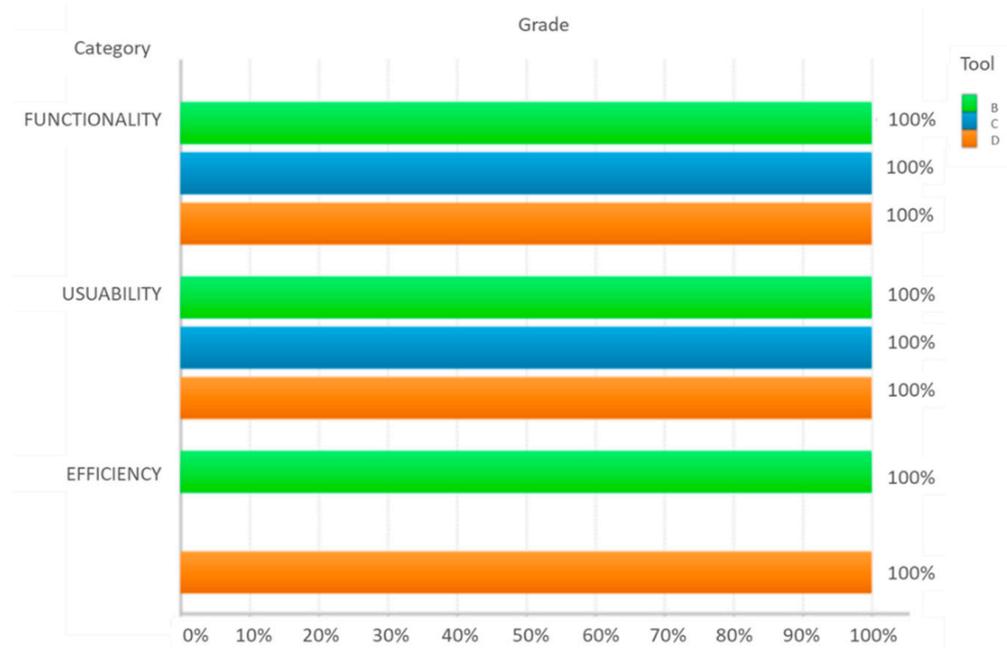
To know the reason software A does not satisfy the *functionality* category, an analysis of a deeper level helped us to know what the scores for each functional characteristic are. Functional characteristics are fit for purpose, interoperability, and security, and Figure 6 shows their respective satisfaction scores. We could see that the characteristic fit for purpose is not satisfied because only 66.67% of its sub-characteristics are satisfied. Particularly, the non-satisfied sub-characteristics are *field relations* and *reporting*.



**Figure 6.** Functionality characteristics results, for tool A.

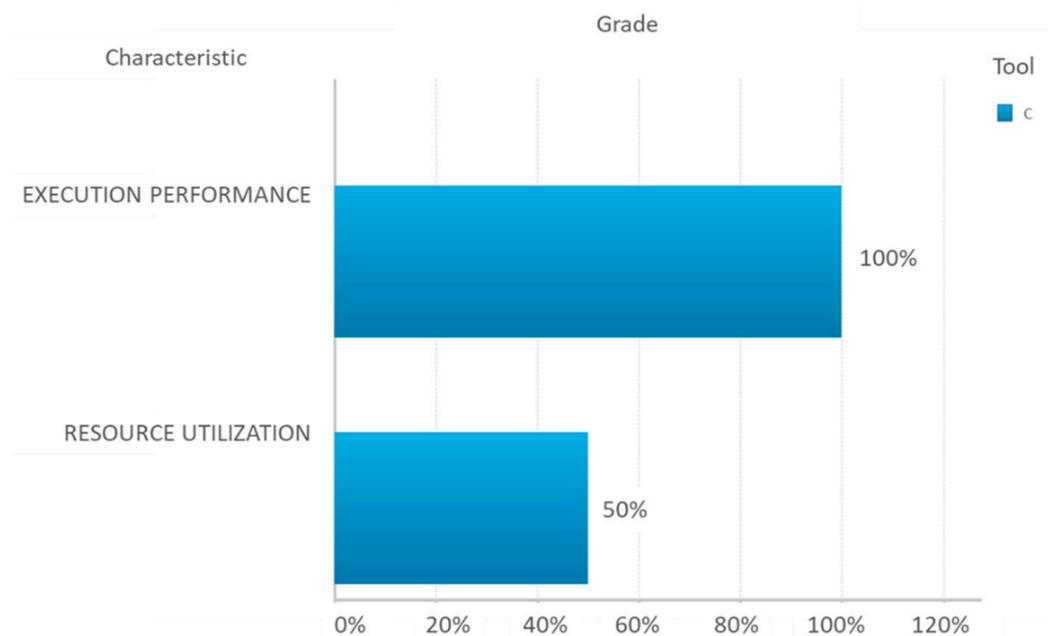
Software A does not satisfy the sub-characteristic Fields relations because it is not capable to alert about the presence of circular references (FFF1), and in fact, it does not skip them (FFF2). Moreover, it cannot directly relate a table to more than one table (FFF3). On the other hand, reporting sub-characteristics is not satisfied because software A does not have an option to build reports (FFR1), (FFR2), and (FFR3). Then, software A evaluation is aborted, and the evaluation continues with the three other tools. The other three tools

satisfy the *usability* category in addition to functionality. Moreover, software B and D also satisfy the *efficiency* category, but C does not. Figure 7 shows the satisfaction score in each category.



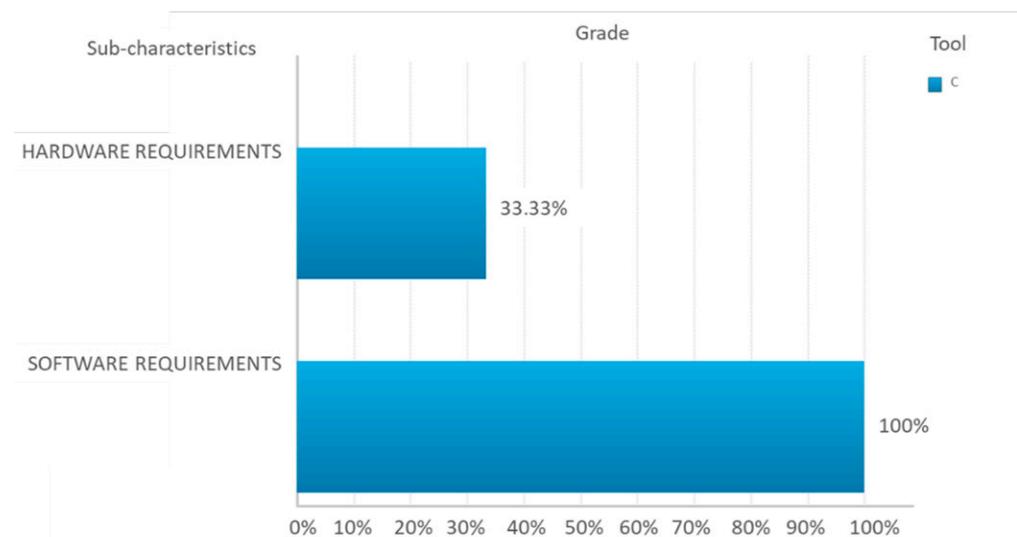
**Figure 7.** Category results.

Software C does not satisfy the efficiency category. It does not satisfy the characteristic *resource utilization*, as it is shown in Figure 8.



**Figure 8.** Resource utilization sub-characteristics results, for tool C.

Resource utilization characteristic has a satisfaction score of 50%, lower than the fixed limit of 75% hence it is considered as not satisfied. Only 50% of the *resource utilization* sub-characteristics are satisfied. In particular, Figure 9 shows the satisfaction scores for the corresponding sub-characteristic.



**Figure 9.** Efficiency characteristics results, for tool C.

*Hardware* requirements sub-characteristic is not satisfied with a 33.33% of satisfaction score because it is the tool that requires more disk space (*ERH3*) and additionally, software C cannot be installed in processors of 32 bits (*ERH1*).

Finally, according to Table 5, the product quality levels of software B, C, and D are those defined in Table 15.

**Table 15.** Quality levels depend on satisfied categories.

Tool	Functionality	Usability	Efficiency	Quality Level
Software B	Satisfied	Satisfied	Satisfied	Advanced
Software C	Satisfied	Satisfied	No satisfied	Medium
Software D	Satisfied	Satisfied	Satisfied	Advanced

Then, software B and D offer an advanced quality level while software C has a medium quality level. To get differences between software B and software D, the fixed levels for satisfaction are increased, being more restrictive. Particularly, we use the following levels defined in Table 16.

**Table 16.** Satisfaction limits, for a second evaluation.

Limit for metric	50%
Limit for sub-characteristic	50%
Limit for characteristic	80%
Limit for category	75%

In this way, a characteristic becomes satisfied if only 80% of its sub-characteristics are satisfied. As it can be seen in Figure 10, only software D satisfies the *functionality* category, unlike software B, which does not, because only 66.67% of its functional characteristics are satisfied.

As is seen in Figure 11, software B does not satisfy the *functionality* category in this second evaluation because the *interoperability* characteristic is not satisfied it has a score of 75%, meaning that only the 75% of the *interoperability* sub-characteristics are satisfied.

It is because the *Portability* sub-characteristic is not satisfied, as a consequence of software B working only on one specific operating system (*F1P1*), and it does not offer an available SaaS (software as a service) edition (*F1P2*).

Then, software D reached an advanced quality level. It can be considered the most appropriate tool for the established requirements.

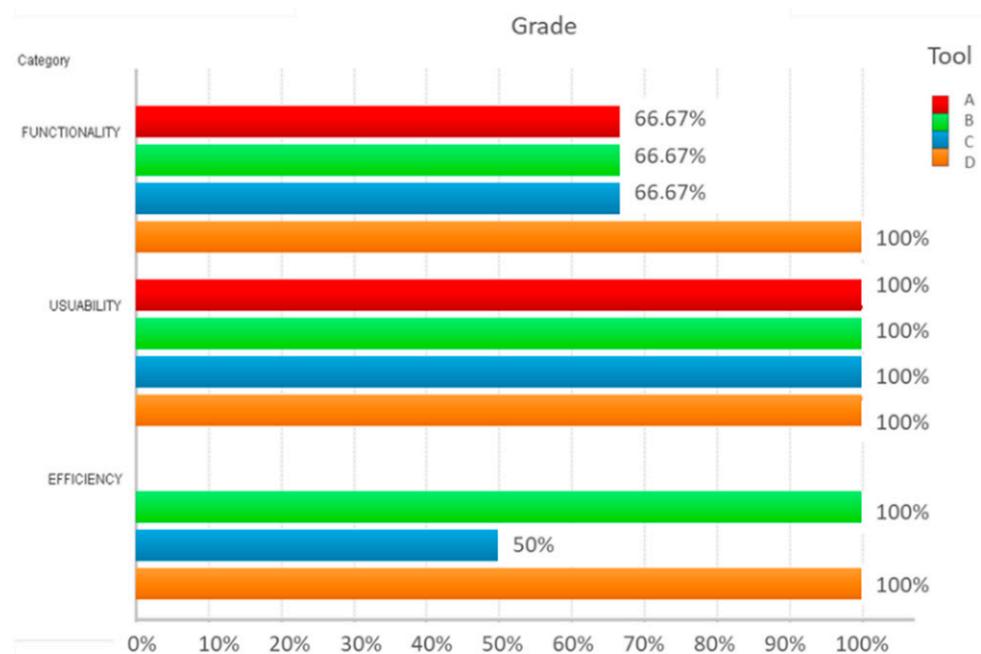


Figure 10. Category results, second evaluation.

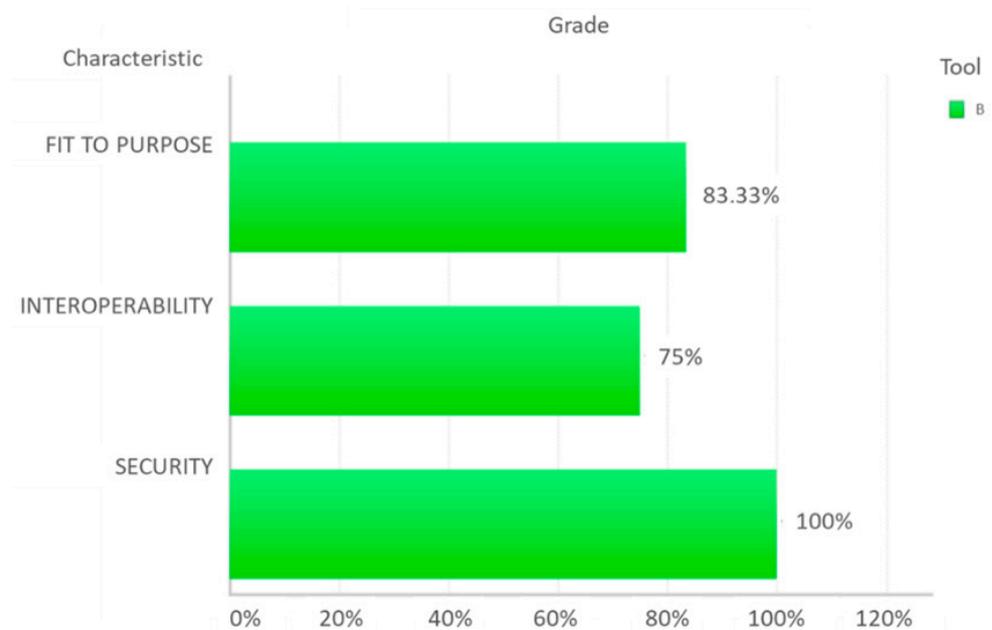


Figure 11. Characteristic results, for a second evaluation for tool B.

## 10. Conclusions

This project has the purpose of building an assessment of self-service BI tools, and evaluating, in particular, four tools (formerly named A, B, C, and D for confidentiality reasons). To build the assessment, an existing quality model is taken as a reference, the systemic quality model (SQMO) developed by the Universidad Simón Bolívar (Venezuela). We adapt it to our aims and then we establish the metrics.

While we are deciding how to measure the metrics, we realize that the cutoff of satisfaction might be subjective. That is why we evaluate with two different satisfaction limits. The first one established that a feature is satisfied if 75% of its sub-characteristics are met. The second one establishes that it is satisfied if 80% of its sub-characteristics are met. In both cases, the rest of the satisfaction limits keep constant. In the first scenario, we observe that tools B and D get an *advanced quality level*, unlike tool C, which gets a *medium*

*quality level*. Tool C is rejected according to the rules established by SQMO. To obtain the differences between tools B and D, we perform the second evaluation being more restrictive in the satisfaction limit. The results are that tool D got an *advanced quality level* and tool C is rejected according to the rules established by SQMO.

The current limitations of the proposal lie in the limitations of the SQMO approach and the metrics selections. In its original form, it does not consider aspects like social or financial, being the original SQMO proposal strongly focused only on software technical specifications [29].

Therefore, depending on the organization, the forms must be adapted to include all those metrics needed to provide a good selection, being this a key aspect to consider depending on the area and the organization. The proposed adaptation of SQMO, with the metrics presented in this paper, can be used as a tool to perform a neutral evaluation of the different BI tools that currently exist in the market. This evaluation mitigates the existing risks in a critical implementation due to the time, resources, and personnel involved in these kinds of projects.

**Author Contributions:** Conceptualization, J.O.H. and P.F.i.C.; methodology, J.O.H. and P.F.i.C.; software, J.O.H.; validation, J.O.H. and P.F.i.C.; formal analysis, J.O.H. and P.F.i.C.; investigation, J.O.H. and P.F.i.C.; resources, J.O.H. and P.F.i.C.; data curation, J.O.H.; writing—original draft preparation, J.O.H. and P.F.i.C.; writing—review and editing, J.O.H. and P.F.i.C.; visualization, J.O.H. and P.F.i.C.; supervision, P.F.i.C.; project administration, J.O.H. and P.F.i.C.; funding acquisition, J.O.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. The Metrics Used in the Selection Process

This appendix includes the metrics used for the selection process for the different categories analyzed.

### Appendix A.1. Functionality Category

#### Appendix A.1.1. Fit to Purpose Characteristic

This characteristic includes different metrics classified into five sub-characteristics: data loading, data model, field relations, analysis, dashboards, and reporting.

**Data loading:** This sub-characteristic includes various metrics to evaluate the loading process.

**Direct connection to a data source (FFI1):** It measures the possibility of a direct connection to data sources. There are some applications with integrated connector drivers (e.g., ODBC, JDBC, etc.) compatible with some databases, and the user does not need to install it to connect the application to the data source.

**Big Data sources (FFI2):** It measures the capability to connect to any Big Data source different from Hadoop.

**Apache Hadoop (FFI3):** It refers to the ability to connect to Hadoop infrastructure. This technology is used to manage large volumes of structured or non-structured data allowing fast access to data. Hadoop simply becomes one more data source and it is the most common way of storing big data.

**Microsoft Access (FFI4):** It evaluates the capability to connect to the Microsoft Access database.

**Excel files (FFI5):** It evaluates the capability to load data from Excel files.

**From an Excel file, load data from all sheets at the same time (FFI6):** It evaluates the capability to load data from all sheets at the same time. In some applications, the user must

do the same data loading process for each one of the sheets, while other tools let the user choose which sheets he wishes to load and import them at the same time.

**Cross-tabs (FFI7):** It measures the capability of loading data from cross-tabs in Excel files. Usually, applications need cross-tabs in a specific format and some of them have an excel complement to normalize the cross-tabs before importing them.

**Plain text (FFI8):** It evaluates the capability of loading data from plain text files (.txt, .inf, .80, .dat, .tmp, .prv, .hlp, .htm., etc.).

**Connecting to the different data sources at the same time (FFI9):** It evaluates the capability to connect the application to several data sources at the same time and to do cross-analysis between data from them.

**Easy integration of many data sources (FFI10):** It evaluates how easy is for the user to integrate many data sources in the data analysis.

**Showing data before the data loading (FFI11):** It evaluates the capability to show data before the data loading. Showing data can be useful for the user to understand how data are before loading them.

**Determining data format (FFI12):** It evaluates the capability to show data formats (*integer, double, date, string...*) of the fields before the data loading. Some applications assign formats to fields automatically while some others let the user assign them before the loading. Determining data formats before the loading is the best choice but, in some applications, it can be done after the loading, and it is equally evaluated.

**Determining data type (FFI13):** It evaluates the capability to show data types (*dimension, measure*) of the fields before the data loading. Some applications assign types to fields automatically, while some others let the user assign them before the loading. Depending on the application's terminology, data types can be attributes or dimensions and measured. Determining data types before the loading is the best choice but, in some applications, it can be done after the loading, and it is equally evaluated.

**Allowing column filtering before the loading (FFI14):** It evaluates the capability to load only the columns that the user wants.

**Allowing row filtering before the loading (FFI5):** It evaluates the capability to filter registers before loading them. Sometimes, the user does not want to analyze the whole dataset, and data filtering can be useful before loading them.

**Automatic measures creation (FFI16):** The ability of the tool to automatically create some measures, possibly useful, from the already loaded data.

**Allow renaming datasets (FFI17):** It evaluates the capability to assign a name to datasets that should be loaded in the application.

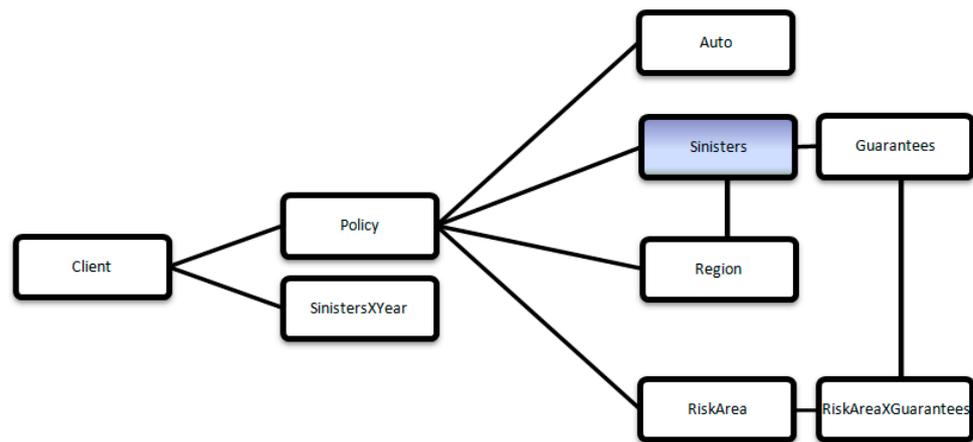
**Allow renaming fields (FFI18):** It evaluates the capability to rename fields. It can be useful when the user has not named the fields in the database by himself and prefers to rename them with more appropriate names for the analysis. Renaming fields before the loading is the best choice but, in some applications, it can be done after the loading, and it is equally evaluated.

**Data cleansing (FFI19):** It evaluates the capability of the applications to allow the user to clean data. For example, drop registers with null values or substitute particular values.

**Data model:** This sub-characteristic includes various sub-metrics to evaluate the modeling process for each tool.

**The data model is done automatically (FFD1):** It refers to the capability of the applications to relate automatically tables. Some applications relate two tables if they have fields with the same name and structure, therefore, these applications model data automatically.

**The done data model is the correct one (FFD2):** This metric evaluates the capability of applications to get relations between tables as the user wants it. In our particular case of 20141220\_Initial\_test data, the model is shown in Figure A1. If the user builds the data model manually, getting the desired model should be easy. While, if the model is done automatically, it can be more difficult depending on if the automatic model is the right one, or if there exists the possibility to modify the model by the user.



**Figure A1.** The correct data model for 20141220\_Initial\_test data.

**The data model can be visualized (FFD3):** This metric evaluates if a tool allows seeing the data model during the analysis. Visualizing the model during the analysis always lets the user check the relations between fields.

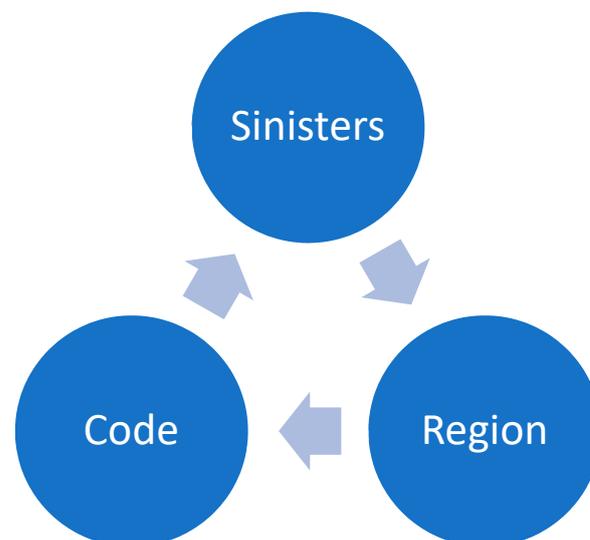
**Field relations:** This sub-characteristic includes several metrics related to the connections between fields when the data source is relational. To clarify some of the proposed metrics, the database 20141220\_Initial\_test is used with examples.

**Alerting about circular references (FFF1):** A circular reference exists when there are, at least, 3 tables related between them.

Figure A2 synthesizes the concept. For example, the user can desire to visualize Table A1; it represents particular policies and the regions where the policies have had an accident. *The policy table* is related to the *region table* by the field *code*, which refers to the identification code for the region where the policy is registered. *Region table* has other fields, additionally to *code*, as the name of the region. On the other hand, the *sinisters table* is also related to the *region table* by the field *code*, which refers to the code identification for the region where accidents occur.

**Table A1.** Circular reference.

Policy_id	Code of the Region	Region
-----------	--------------------	--------



**Figure A2.** Circular reference.

In that particular case, some applications could show non-correct values for the *region* because of the ambiguity about which way to take to reach the *region table*. If it passes by the *policy table* then it shows regions where the policy is registered, but if it passes by the *sinister table*, it shows regions where accidents occur. This metric evaluates the capability of a tool to realize a circular reference and alert the user about it.

**Skipping circular references (FFF2):** This sub-characteristic evaluates the capability of the software to omit circular references.

**The same table can be used several times (FFF3):** It evaluates the capability of the application to use a table directly related to more than one table. For example, if there is a table with coordinates, it can be related to more than one table, for example, with two tables where in the first table there is a place of birth and in the second one there is a place of death. Some tools allow the to load just once the table and use it as many times as the user needs. Other tools require loading the table as many times as relations it will have.

**Analysis:** This sub-characteristic includes several metrics about the capabilities of the analysis.

**Creating new measures based on previous measures (FFA1):** All the applications analyzed must be able to create a measure based on already loaded measures. This sub-characteristic evaluates how easy is to build new measures based on loaded measures.

**The creation of new measures based on dimensions (FFA2):** This sub-characteristic evaluates how easy is to build new measures based on loaded dimensions.

**Variety of functions (FFA3):** It measures the diversity of functions offered by the application to build a new field. Applications can offer functions related to statistics, economics, mathematics, and also with strings and logic functions.

**Descriptive statistics (FFA4):** It refers to the possibility to analyze data statistically from a descriptive point of view. All the applications analyzed in that project can do descriptive statistics. Therefore, this metric evaluates the complexity of the descriptive statistic allowed in each program.

**Predictive statistics (FFA5):** It measures the ability to get indicators by predictive functions. It is not a common feature in self-service BI tools and because of that, the presence of few predictive methods will be positively evaluated.

**R connection (FFA6):** It evaluates the capability of applications to connect to R to get advanced analytical functions.

**Geographic information (FFA7):** This sub-characteristic measures the capability of displaying data on maps.

**Time hierarchy (FFA8):** It evaluates the capability of the application to create *time intelligence*. It consists in, from a particular date, creating other fields like a month, quarter, or year. These sets of fields are grouped in a hierarchy. Particularly, a time hierarchy. This metric evaluates the capability of the tool to create automatic time hierarchies.

**Creating sets of data (FFA9):** It evaluates the capability of a tool to create sets of data. During the analysis, the user can be interested in a deeper analysis of a set of registers. Some tools let to save these datasets and work with them.

**Filtering data by an expression (FFA10):** It evaluates the capability of a tool to filter data during the analysis by expression values.

**Filtering data by a dimension (FFA11):** It evaluates the capability of a tool to filter data during the analysis by dimension values.

**Visual perspective linking (FFA12):** It evaluates the capability to link multiple images, so a selection on one image shows related and relevant data in other images.

**No null data specifications (FFA13):** This metric evaluates if the applications have any requirements to the null values, for example, that null values must be noted as NULL, or just with a space or by contrary that the user can define how are the null values represented in the data source.

**Considering nulls (FFA14):** This metric measures if applications consider null values as another value. Considering null as another value might be useful because the user can

visualize the behavior of null data and then detect a pattern for them. This metric also evaluates if null values are skipped from a calculated expression.

**Variety of graphs (FFA15):** It measures the diversity of graphs offered by the application.

**Modifying graphs (FFA16):** It measures the capability to modify the default setting of graphs. For example, if there is the possibility to change levels of a legend, change colors, change the shapes of markers... It is an important characteristic because sometimes it is the key to understanding a data pattern.

**Huge amount of data (FFA17):** It measures the capability to display a huge amount of data. Particularly, it measures the capability to display datasets without any data problems because of their size.

**Data refresh (FFA18):** It measures the capability to update data automatically. For example, if data are modified in the original file, some applications update automatically the data while in others the user must do it, manually.

**Dashboard:** This sub-characteristic includes several metrics to measure the capabilities of a tool relating to dashboards.

**Dashboard exportation (FFD1):** It evaluates the capability of the tool to export the dashboard to share with other people to visualize and interact with the results.

**Templates (FFD2):** It evaluates the capability to fix a schema dashboard or access templates to use it several times with different types of data. It is a useful feature to homogenize projects.

**Free design (FFD3):** It measures the ability to let the user build dashboards with total freedom. Some tools have limited options for building dashboards, while others let the user insert text, format it, insert images, etc.

**Reporting:** This sub-characteristic includes several metrics to measure reporting capabilities of a tool.

**Reports exportation (FFR1):** It evaluates the diversity of formats to export reports. Some formats are Excel spreadsheets, PDF files, HTML files, Flash files, the own tool format, etc.

**Templates (FFR2):** It evaluates the capability to fix a schema report or access templates to use it several times with different data. It is a useful feature to improve consistency when the user builds the same type of report periodically.

**Free design (FFR3):** It measures the ability to let the user build reports with total freedom. Some tools have limited options for building dashboards, while others let the user insert text, format it, insert images, etc.

#### Appendix A.1.2. Interoperability Characteristic

This characteristic includes several sub-characteristics to evaluate the capability of an application to work with other organizations and systems.

**Languages:** This sub-characteristic is composed of a metric, which evaluates the variety of languages displayable in the tool.

**Languages displayed (FIL1):** It evaluates the variety of displayed languages offered by the tool. In particular, it evaluates if the tool can be displayed in more than two languages or not.

**Portability:** This sub-characteristic is composed of three metrics, which evaluate the ability of a tool to be executed in different environments.

**Operating systems (FIP1):** This metric measures the variety of different operating systems compatible with the tool. In particular, it evaluates if the tool can work, at least, in two different operating systems.

**SaaS/Web (FIP2):** The acronym SaaS means Software as a Service. This metric evaluates if a tool offers access to projects via a web browser for hosting their deployments in the cloud.

**Mobile (FIP3):** It evaluates the possibility to have reports and dashboards available on the mobile device via a mobile app.

**Use project by third parts:** This sub-characteristic is composed of a unique metric, and it measures the capability of sharing and modifying projects by other people.

**Using the project by a third party (FIU1):** It evaluates the capability to share projects and modify them with other users.

**Data exchange:** This sub-characteristic is composed of metrics, which evaluate the data exportation when they have already been manipulated in the tool.

**Exportation in .txt (FID1):** It evaluates the capability of a tool to export data .txt.

**Exportation in CSV (FID2):** It evaluates the capability of a tool to export data in CSV format.

**Exportation in HTML (FID3):** It evaluates the capability of a tool to export data in HTML format.

**Exportation in Excel file (FID4):** It evaluates the capability of a tool to export data in Excel files.

### Appendix A.1.3. Security Characteristic

This characteristic is composed of a unique sub-characteristic, which groups metrics about the security process.

**Security devices:** This sub-characteristic is composed of two metrics related to the protection of data.

**Password protection (FSS1):** It evaluates the capability to protect projects with a password.

**Permissions (FSS2):** It evaluates the capability to assign different permissions to different users.

### Appendix A.2. Usability Category

#### Appendix A.2.1. Ease of Understanding and Learning Characteristic

This characteristic includes different sub-characteristics.

**Learning time:** This sub-characteristic includes only one metric.

**Average learning time (UEL1):** This metric measures the time spent by the user in learning the functionality of the tool.

**Browsing facilities:** This sub-characteristic evaluates how the user can browse inside the tool.

**Consistency between icons in the toolbars and their actions (UEB1):** This metric measures the capability of the tool to be consistent with its icons.

**Displaying right-click menus (UEB2):** This metric measures if the tool offers a displaying menu by right-clicking.

**Terminology:** This sub-characteristic evaluates if the terminology is consistent with the global business intelligence terminology.

**Ease of understanding the terminology (UET1):** This metric measure how easy is for the user to understand the terminology.

**Help and documentation:** This sub-characteristic is composed of metrics, which measure the help offered by the tool to a user when he has doubts about the functionality or management of the tool.

**User guide quality (UEH1):** This metric evaluates if the user guide is understandable. Highlighting that self-service tools are also offered for non-technical users.

**User guide acquisition (UEH2):** This metric measures the process to get to the user manual. For example, if it is free if it is difficult to find on the web, etc.

**On-line help (UEH3):** It measures the offering of online help.

**Support and training:** These sub-characteristics measure the quality and variety of the support offered by the tool.

**Availability of tailor-made training courses (UES1):** It measures if the tool offers training courses adapted to organizations, and it is positively measured if the course can be done in the organization.

**Phone technical support (UES2):** It measures if the tool offers a phone for technical support and the timetable of it.

**Online support (UES3):** It measures if the tool offers online support and if it is in life or not.

**Availability of consulting services (UES4):** It measures if the company offers consulting services.

**Free formation (UES5):** It evaluates if the platform offers free formation for users.

**Community (UES6):** It evaluates if there exists a community to ask for doubts or to share knowledge with other users.

#### Appendix A.2.2. Graphical Interface Characteristic

This characteristic evaluates the graphical interface of the tool.

**Windows and mouse interface:** This sub-characteristic evaluates the windows interface and the mouse functions.

**Editing elements by double-clicking (UGW1):** It measures if the tool offers editing elements by double-clicking.

**Dragging and dropping elements (UGW2):** It measures the capability of the tool in dragging and dropping elements.

**Display:** This sub-characteristic refers to a unique metric about the capability of editing the screen layout.

**Editing the screen layout (UGD1):** It measures the capability of a tool to edit the screen layout.

#### Appendix A.2.3. Operability Characteristic

This characteristic evaluates the ability of the tool to keep the system and the tool in reliable functioning conditions.

**Versatility:** This sub-characteristic evaluates the versatility of the tool.

**Automatic update (UOV1):** It measures if the tool is automatically updated when new versions appear.

### Appendix A.3. Efficiency Category

#### Appendix A.3.1. Execution Performance Characteristic

This characteristic is composed of a sub-characteristic, which evaluates the execution performance of the tool.

**Compilation speed:** This sub-characteristic measures the compilation speed, and how fast the software builds a particular chart.

**Compilation speed (EEC1):** It measures the compilation speed. It is a very subjective measure because it depends on the machine where it is installed.

#### Appendix A.3.2. Resource Utilization Characteristic

This characteristic is composed of two (2) sub-characteristics, which evaluate the extra hardware and software requirements.

**Hardware requirements:** This sub-characteristic is composed of three metrics, which measure the vital hardware to run the tool.

**CPU (processor type) (ERH1):** This metric evaluates if the tool can be installed as much to  $\times 86$  processors as to  $\times 64$  processors.

**Minimum RAM (ERH2):** It measures the RAM needed in the way that maximum punctuation means it requires low memory while minimum punctuation means it needs much memory.

**Hard disk space required (ERH3):** It measures the hard disk space needed in the way that maximum punctuation means it requires low space while minimum punctuation means it needs much memory.

**Software requirements:** This sub-characteristic is composed of a unique metric, which measures if extra software is required to execute the tool.

**Additional software requirements (ERS1):** this metric evaluates if extra software is required to run the tool.

## Appendix B. Metrics Weights

**Table A2.** Metric weights.

Metric	Weight
Direct connection to data sources	2
Bigdata sources	1
Apache Hadoop	1
Microsoft Access	2
Excel files	3
From an excel file, import all sheets at the same time	2
Cross-tabs	2
Plain text	3
Connecting to different data sources at the same time	2
Easy integration of many data sources	2
Visualizing data before the loading	2
Determining data format	2
Determining data type	2
Allowing column filtering before the loading	2
Allowing row filtering before the loading	2
Automatic measures creation	3
Allow renaming datasets	2
Allow renaming fields	3
Data cleansing	2
The data model is done automatically	2
The done data model is the correct one	2
The data model can be visualized	3
Alerting about circular references	3
Skipping with circular references	3
The same table can be used several times	2
Creating new measures based on previous measures	3
Creating new measures based on dimensions	3
Variety of functions	3
Descriptive statistics	2
Prediction functions	2
R connection	2
Geographic information	2
Time hierarchy	3
Creating sets of data	2
Filtering data by expression	3
Filtering data by dimension	3
Visual perspective linking	2
No null data specifications	2
Considering nulls	3

## Appendix C. 0141220\_Initial\_Test

The created database, used to evaluate the applications, is called **20141220\_Initial\_test**. It is composed of nine tables forming a relational database, particularly a snowflake schema.

In Figure A3, it is showed the relational data model structure, where two tables are related by a common field (*foreign key*), which appears in both tables and which is shown in the figure, next to the type of relationship. The *fact table* is called *sinisters*, and the *dimension tables* are *client*, *policy*, *auto*, *region*, *SinistersXYear*, *risk area*, *guarantees*, and *GuaranteesXRiskArea*.

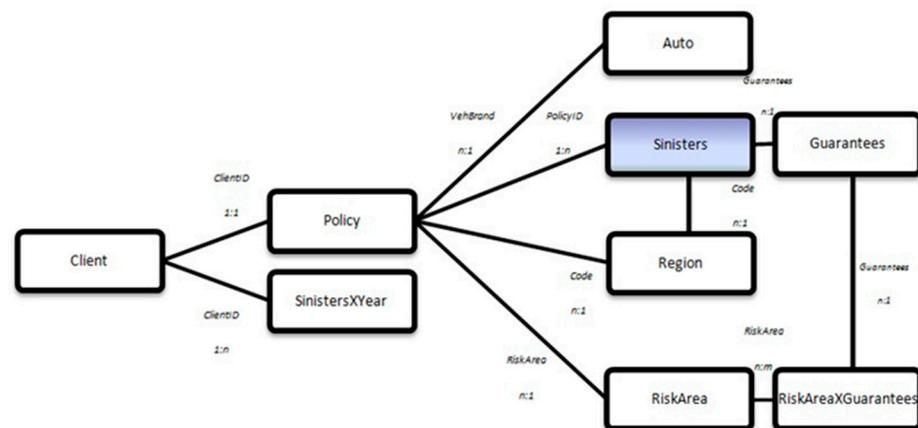
We decided to propose a relational database to realize how the evaluated self-service BI tools managed the relations between tables. Some of these tools built automatically the data model (the relations between tables), that is that the user loads tables, and the tool, by itself, relates tables. Hence, we wanted to know if this automatic modeling worked well or not.

Moreover, we wanted to evaluate if applications were capable to understand both types of relationships. The most common relationship is 1: n, and we were almost certain that applications support them. However, we doubted the support of n:m relationships. There was one of the evaluated tools, could not relate two tables by an n:m relationship.

Additionally, our model has a particularity. There are two circular references in *region* and *guarantees* fields. A circular reference exists when there are, at least, 3 tables related between them. For example, the *Region* table has information about the regions, and it has the name of all regions and their population. The *policy* table is connected to the *region* table, by the field *code*. This field corresponds to the code of the region where the policy is registered. On the other hand, the *sinisters* table is also connected to the *region* table, by the field *code*. However, this time, it corresponds to the code of the region where the accident had happened. Both relations have different meanings, but they are related to the same table. We added these circular references to know how the self-service applications managed them.

Skipping circular references can be done easily, by duplicating tables. We have loaded two tables identically equal to the *region* table, one is related to the *policy* table and the other one to the *sinisters* table. However, this action implies the use of more memory, and it is not recommended.

The fact that we decided to simulate a car's insurance company database is due to it is a common case of use in consultancy. Moreover, we were lucky to know an actuarial expert who offered us some information about the car's insurance area.



**Figure A3.** The data model for 20141220\_Initial\_test.

The main point of the work was not to do an accurate analysis of data. For this reason, the simulation was just a way of getting data and they cannot be considered real data, because the process to get them is just a rough approximation.

We obtained some data from two existing datasets of R. To not have to invent all data, although some fields were invented by us because they were not in the existing datasets. Some data were extracted from the *CASdatasets* package of R. It is composed of several actuarial datasets (originally for the “Computational Actuarial Science” book). Particularly, we extracted some data from *freMPL6* and *freMTPL2freq* datasets.

Moreover, to evaluate the analysis capabilities of the applications, data were simulated forcing patterns. In particular, geographical, and stationary patterns were imposed. In **20141220\_Initial\_test**, the amount of occurred car accidents in a region is proportional to the amount of population in it. However, in the months of July, August, and September, in the region of Granada, we force to have more accidents. Additionally, some people are

forced to have more probability to have accidents than the main part of the population. They are a woman with ages between 40 and 45, a man with ages between 50 and 65, young people under 24, and beginners.

The database consists of 26000 policies. Each policy is identified by one client. There are 22 different variables, classified in tables.

#### Appendix C.1. Client Table

It consists of client characteristics. As a consequence, there are 26,000 clients, and this table has 26,000 rows. The variables inside are the followings:

- **ClientID:** Discrete variable with 26,000 different values from 1 to 26,000.
- **Gender:** Qualitative variable that can take the answers "Male" and "Female".
- **Maristat:** Qualitative variable that can take the answers "Others" and "Alone".
- **CSP:** Qualitative variable referenced to the social category known as CSP in France. It can take values as "CSP50". The classification of socio-professional categories (CSP) was conceived by INSEE in 1954. The objective was to categorize individuals according to their professional situation, taking account of several criteria: their profession, economic activity, qualification, hierarchical position and status.
- It included 9 main groups subdivided into 30 socio-professional categories.
- **LicBeg:** Date variable with 26,000 values between "5 September 1941" and "11 January 2009". It refers to the date when the client got the driver's license.
- **DrivBeg:** Date variable with 26,000 values between "25 January 1916" and "6 January 1991". It refers to the birth date client.

#### Appendix C.2. Auto Table

It consists of vehicle characteristics depending on its brand. As a consequence there are 11 different brands of vehicles, this table has 11 rows. The variables inside are the followings:

- **VehBrand:** Categorized numeric variable that refers to the vehicle brand. It takes 11 different values from {1, 2, 3, 4, 5, 6, 10, 11, 12, 13, 14}.
- **VehPow:** Categorized numeric variable that refers to the vehicle power. It takes 4 different values from {6, 7, 8, 9}.
- **VehType:** Qualitative variable that refers to the vehicle type. There are 4 possible answers: "familiar", "compact", "sport" and "terrain".

#### Appendix C.3. Region Table

It shows the information about regions. There are 52 regions and therefore this table has 52 rows. Its variables are the followings:

- **Code:** Numeric variable which refers to a code for each region. For example, Zaragoza has the code 52, because it is the last region if they are ordered alphabetically.
- **Region:** Qualitative variable that refers to the Spain province(region) where the sinister happens. The answers can be: "Alava", "Albacete", "Alicante", "Almeria", "Asturias", "Avila", "Badajoz", "Barcelona", "Burgos", "Caceres", "Cadiz", "Cantabria", "Castellon", "Ciudad Real", "Cordoba", "La Coruna", "Cuenca", "Gerona", "Granada", "Guadalajara", "Guipuzcoa", "Huelva", "Huesca", "Islas Baleares", "Jaen", "Leon", "Lerida", "Lugo", "Madrid", "Malaga", "Murcia", "Navarra", "Orense", "Palencia", "Palmas", "Pontevedra", "La Rioja", "Salamanca", "Segovia", "Sevilla", "Soria", "Tarragona", "Teruel", "Tenerife", "Toledo", "Valencia", "Valladolid", "Vizacaia", "Zamora", "Zaragoza", "Melilla" and "Ceuta". is a the whole name of the region, in last example region should be "Zaragoza".
- **Population:** Discrete variable that refers to the population of the region where the car is declared.

#### Appendix C.4. Risk Area Table

It has the name of each type of the 5 Risk Area types. As there are 5 different Risk Areas there are 5 rows in this table. The variables inside are the following:

- *RiskArea*: Categorized variable meaning the type of insurance included in each policy, also known as the product. The categories are {1,2,3,4,5}.
- *RiskAreadesc*: Qualitative variable referring to the corresponding Risk Area name. They are “gold”, “silver”, “master”, “plus” and “regular”.

#### Appendix C.5. Guarantees Table

This table shows two variables. The name of different guarantees covered by the insurance company and their base cost. There are 8 different guarantees and therefore the table has 8 rows.

- *Guarantees*: Qualitative variable referring to guarantees with the following possible answers: “windows”, “travelling”, “driver insurance”, “claims”, “fire”, “theft”, “total loss”, and “health assistant”.
- *Base*: Discrete Variable refers to the cost covered by the insurance company. It has 8 different values: 50, 500, 100, 25, 1000, 20,000, 3000 and 300.

#### Appendix C.6. Risk X Guarantees Table

This table joins the Risk Area variable with the guarantees that it provides. This table has 29 rows because the Risk Area categorized by 1 has 8 guarantees, the 2 has 7 guarantees, the 3 has 6 guarantees, the 4 has 5 guarantees and the 5 has 3 guarantees.

- *RiskArea*: Categorized variable meaning the type of insurance included in each policy, also known as the product. The categories are {1,2,3,4,5}.
- *Guarantees*: Qualitative variable referring to guarantees provided by the Risk Area corresponding. The different answers are “windows”, “travelling”, “driver insurance”, “claims”, “fire”, “theft”, “total loss” and “health assistance”.

#### Appendix C.7. Policy Table

It consists of policy characteristics. As a consequence, there are 26,000 different policies, and this table has 26,000 rows. Its variables are the followings:

- *PolicyID*: Discrete variable with 26,000 different values from 1 to 26,000.
- *ClientID*: Discrete variable with 26,000 different values from 1 to 26,000.
- *RecordBeg*: Date variable with 26,000 values between “1 January 2000” and “31 December 2010”. It refers to the date when the policy is begun.
- *RecordEnd*: Date variable with 26,000 values between “1 January 2000” and “31 December 2010”. It refers to the date when the policy is finished. In most cases, it has the value “1 January 9999”, which means that the policy is still active.
- *VehBeg*: Date variable with 26,000 values between “26 January 1911” and “1 January 2011”. It refers to the date when the vehicle was built.
- *VehBrand*: Categorized numeric variable that refers to the brand of the secured vehicle, by the policy. It takes 11 different values from {1, 2, 3, 4, 5, 6, 10, 11, 12, 13, 14}.
- *BonusMalus*: Discrete variable between 50 and 350: below 100 means bonus, above 100 means malus in France.
- *RiskArea*: Categorized variable meaning the type of insurance included in each policy, also known as the product. The categories are {1, 2, 3, 4, 5}.
- *Code*: Numeric variable which refers to the code of the region where the policy is registered.

#### Appendix C.8. SinistersXYear Table

It shows the number of sinister for each policy in each year, from 2000 to 2010. As there are 26,000 different policies, the table has 26,000 different rows. About the columns,

it has one for each year, therefore there are 11 columns. The unique variable is *sinister*. It is evaluated for each policy and each year. Three is the maximum of sinister in one year, imposed in the simulation process.

#### Appendix C.9. Sinisters Table

This table shows information about the sinister of each policy. It has 196,235 rows. It is because in total, for all the policies there are 196,235 sinister. The variables inside are the followings:

- *PolicyID*: Discrete variable with 26,000 different values from 1 to 26,000. However, in the table, every policy is repeated as times as sinister has it in the total of 11 years.
- *RiskArea*: Categorized variable meaning the type of insurance included in each policy, also known as the product. The categories are {1,2,3,4,5}. As the *PolicyID*, it is repeated as times as the policy uses the risk area that has been included.
- *Guarantees*: Qualitative variable referring to a guarantee provided by the Risk Area corresponding and used in the sinister. The different answers are “windows”, “travelling”, “driver insurance”, “claims”, “fire”, “theft”, “total loss” and “health assistance”.
- *Sinisterdate*: Date variable with values between “1 January 2000” and “31 December 2010”. It refers to the date when the sinister happens.
- *Code*: Numeric variable which refers to the code of the region where the sinister happens.

The names of the variables explained above, Table A3, are the final names. During the simulation process, some variables were called different. In the simulation, some variables have been built two times with different lengths. To keep the consistency in the code, all different fields are called different. However, after the simulation, the names have been changed to let the user build connections between tables by a common field.

**Table A3.** Variables names in the data simulation.

Variable Name in Simulation	Table	Variable Name
VehBrand_N	<i>Policy</i>	VehBrand
RiskArea_N	<i>Policy</i>	RiskArea
Code_O	<i>Policy</i>	Code
V1, . . . , V11	<i>SinistersXYear</i>	2000, . . . , 2010
PolicyID_S	<i>Sinisters</i>	PolicyID
RiskArea_s	<i>Sinisters</i>	RiskArea
Guarantees_s	<i>Sinisters</i>	Guarantees
Code_S	<i>Sinisters</i>	Code

#### Appendix D. Questionnaires

This annex attaches one of the questionnaires, Table A4, that must be filled by the users implied in the evaluation. In our case, there are four questionnaires, one for each evaluated tool. They show the scores, according to the scale of measurement established in Section 5, for each metric. Moreover, according to the satisfaction limits established in Section 6 for the second evaluation, the satisfaction score is shown.

The column **M.S** refers to the scale of measurement established for each metric. The column **WEIGHTS** refers to the weights established for each metric in Section 5. The **COMPENSATED VALUE** refers to the product of the weight and the metric’s score.

The **NORMALIZED VALUE** is the satisfaction score for the metric. While for sub-characteristics/characteristics/categories they are called simply **TOTAL**.

The columns called **INDICATOR** take values 1 or 0, depending on if the Metric/Sub-characteristic/Characteristic/Category is satisfied according to the satisfaction limit established in Section 6.

Table A4. Evaluation questionnaire for B tool.

Category	Characteristic	Subcharacteristic-Desc	Metric-Code	Metric	M.S	Value	Weight	Compensated Value	Normal. Value	Indicator_Metric	Total Sub-Characteristic	Indicator_Sub-Charac.	Total Characteristic	Indicator_Charac	Total Category	Indicator_Categ.
Functionality	Fit to Purpose	Data loading	FFI1	Direct connection to data sources	A	2	2	4	50.00%	1	82.50%	1	83.33%	1	66.67%	0
			FFI2	BigData sources	A	2	1	2	50.00%	1						
			FFI3	Apache Hadoop	A	2	1	2	50.00%	1						
			FFI4	Microsoft Access	A	2	2	4	50.00%	1						
			FFI5	Excel files	A	3	3	9	75.00%	1						
			FFI6	From an excel file, import all sheets at the same time	A	1	2	2	25.00%	0						
			FFI7	Cross-tabs	A	4	2	8	100.00%	1						
			FFI8	Plain text	A	3	3	9	75.00%	1						
			FFI9	Connecting to different data sources at the same time	A	3	2	6	75.00%	1						
			FFI10	Easy integration of many data sources	A	3	2	6	75.00%	1						
			FFI11	Visualizing data before the loading	A	1	2	2	25.00%	0						
			FFI12	Determining data format	A	3	2	6	75.00%	1						
			FFI13	Determining data type	A	4	2	8	100.00%	1						
			FFI14	Allowing column filtering before the loading	A	3	2	6	75.00%	1						
			FFI15	Allowing row filtering before the loading	A	2	2	4	50.00%	1						
			FFI16	Automatic measures creation	A	0	3	0	0.00%	0						
			FFI17	Allow renaming datasets	A	3	2	6	75.00%	1						
			FFI18	Allow renaming fields	A	3	3	9	75.00%	1						
			FFI19	Data cleansing	A	2	2	4	50.00%	1						

Table A4. Cont.

Category	Characteristic	Subcharacteristic-Desc	Metric-Code	Metric	M.S	Value	Weight	Compensated Value	Normal. Value	Indicator_Metric	Total Sub-Characteristic	Indicator_Sub-Charac.	Total Characteristic	Indicator_Charac	Total Category	Indicator_Categ.	
		Data model	FFD1	The data model is done automatically	A	3	2	6	75.00%	1	100.00%	1					
			FFD2	The done data model is the correct one	A	2	2	4	50.00%	1							
			FFD3	The data model can be visualized	A	4	3	12	100.00%	1							
		Field relations	FFF1	Alerting about circular references	A	3	3	9	75.00%	1	75.00%	1					
			FFF2	Skipping with circular references	A	3	3	9	75.00%	1							
			FFF3	The same table can be used several times	A	0	2	0	0.00%	0							
		Analysis	FFA1	Creating new measures based on previous measures	A	3	3	9	75.00%	1	73.33%	1					
			FFA2	Creating new measures based on dimensions	A	3	3	9	75.00%	1							
			FFA3	Variety of functions	A	3	3	9	75.00%	1							
			FFA4	Descriptive statistics	A	3	2	6	75.00%	1							
FFA5	Prediction functions		A	0	2	0	0.00%	0									
FFA6	R connection		A	3	2	6	75.00%	1									
FFA7	Geographic information		A	2	2	4	50.00%	1									
FFA8	Time hierarchy	A	2	3	6	50.00%	1										
			FFA9	Creating sets of data	A	1	2	2	25.00%	0							
			FFA10	Filtering data by expression	A	1	3	3	25.00%	0							

Table A4. Cont.

Category	Characteristic	Subcharacteristic-Desc	Metric-Code	Metric	M.S	Value	Weight	Compensated Value	Normal. Value	Indicator_Metric	Total Sub-Characteristic	Indicator_Sub-Charac.	Total Characteristic	Indicator_Charac	Total Category	Indicator_Categ.
			FFA11	Filtering data by dimension	A	1	3	3	25.00%	0						
			FFA12	Visual Perspective Linking	A	4	2	8	100.00%	1						
			FFA13	No Null data specifications	A.1	0	2	0	0.00%	0						
			FFA14	Considering nulls	A	4	3	12	100.00%	1						
			FFA15	Variety of graphs	A	3	3	9	75.00%	1						
			FFA16	Modify graphs	A	4	3	12	100.00%	1						
			FFA17	No limitations to displaying large amounts of data	A.1	4	2	8	100.00%	1						
			FFA18	Data refresh	A	2	2	4	50.00%	1						
			FFD1	Dashboards Exportation	A	3	3	9	75.00%	1						
		Dashboards	FFD2	Templates	A	0	2	0	0.00%	0	71.43%	1				
			FFD3	Free design	A	4	2	8	100.00%	1						
			FFR1	Reports Exportation	A	3	3	9	75.00%	1						
		Reporting	FFR2	Templates	A	0	2	0	0.00%	0	42.86%	0				
			FFR3	Free design	A	1	2	2	25.00%	0						
		Languages	FIL1	Languages displayed	A.1	4	2	8	100.00%	1	100.00%	1				
			FIP1	Operating Systems	A.1	0	2	0	0.00%	0						
		Portability	FIP2	SaaS/Web	A	1	1	1	25.00%	0	40.00%	0				
			FIP3	Mobile	A	3	2	6	75.00%	1						
		Using the project in third parts	FIU1	Using the project by third parts	A	3	2	6	75.00%	1	75.00%	1				
			FID1	Exportation in txt	A	3	2	6	75.00%	1			75.00%	0		
			FID2	Exportation in CSV	A	3	2	6	75.00%	1						
		Data exchange	FID3	Exportation in HTML	A	3	2	6	75.00%	1	100.00%	1				
			FID4	Exportation in Excel file	A	3	3	9	75.00%	1						

Table A4. Cont.

Category	Characteristic	Subcharacteristic-Desc	Metric-Code	Metric	M.S	Value	Weight	Compensated Value	Normal Value	Indicator_Metric	Total Sub-Characteristic	Indicator_Sub-Charac.	Total Characteristic	Indicator_Charac	Total Category	Indicator_Categ.	
	Security	Security devices	FSS1	Password protection	A	3	3	9	75.00%	1	100.00%	1	100.00%	1			
			FSS2	Permissions	A	3	3	9	75.00%	1							
		Learning time	UEL1	Average learning time	A.2	4	3	12	100.00%	1	100.00%	1					
		Browsing facilities	UEB1	Consistency between icons in the toolbars and their actions	A	3	3	9	75.00%	1	100.00%	1					
Usability	Ease of Understanding and Learning			UEB2	Displaying right-click menus	A	3	3	9	75.00%	1			100.00%	1	100.00%	1
		Terminology		UET1	Ease of understanding the terminology	A	4	3	12	100.00%	1						
		Help and documentation	UEH1	User guide quality	A.2	3	2	6	75.00%	1	100.00%	1					
			UEH2	User guide acquisition	A	3	2	6	75.00%	1							
		Support training		UEH3	On-line documentation	A	3	2	6	75.00%	1	100.00%	1				
				UES1	Availability of tailor-made training courses	A	3	2	6	75.00%	1						
UES2	Phone technical support				A	3	2	6	75.00%	1							
UES3	Online support			A	3	2	6	75.00%	1	100.00%	1						
			UES4	Availability of consulting services	A	3	2	6	75.00%	1							
			UES5	Free formation	A	3	2	6	75.00%	1							
	Graphical Interface Characteristic	Windows and mouse interface	UGW1	Editing elements by double-clicking	A	0	2	0	0.00%	0	50.00%	1	100.00%	1			
			UGW2	Dragging and dropping elements	A	3	2	6	75.00%	1							
		Display	UGD1	Editing the screen layout	A	3	2	6	75.00%	1	75.00%	1					
	Operability	Versatility	UOV1	Automatic update	A	2	2	4	50.00%	1	50.00%	1	100.00%	1			

Table A4. Cont.

Category	Characteristic	Subcharacteristic-Desc	Metric-Code	Metric	M.S	Value	Weight	Compensated Value	Normal. Value	Indicator_Metric	Total Sub-Characteristic	Indicator_Sub-Charac.	Total Characteristic	Indicator_Charac	Total Category	Indicator_Categ.
Efficiency	Execution Performance	Compilation speed	EEC1	Compilation Speed	A.2	4	2	8	100.00%	1	100.00%	1	100.00%	1		
			ERH1	CPU(processor type)	A.1	4	2	8	100.00%	1						
	Resource Utilization	Hardware requirements	ERH2	Minimum RAM	A.2	3	2	6	75.00%	1	100.00%	1	100.00%	1	100.00%	1
			ERH3	Hard disk space required	A.2	4	2	8	100.00%	1						
		Software requirements	ERS1	Additional software requirements	A	4	2	8	100.00%	1	100.00%	1				

## References

1. Hass, K.B.; Lindbergh, L.; Vanderhorst, R.; Kiemski, K. *From Analyst to Leader: Elevating the Role of the Business Analyst Management Concepts*; Berrett-Koehler Publishers: Oakland, CA, USA, 2007; p. 152.
2. International Institute of Business Analysis. *The Guide to the Business Analysis Body of Knowledge TM*; International Institute of Business Analysis: London UK, 2012.
3. Pollock, N.; Williams, R. Technology choice and its performance: Towards a sociology of software package procurement. *Inf. Organ.* **2007**, *17*, 131–161. Available online: <http://www.sciencedirect.com/science/article/pii/S147177270700022X> (accessed on 9 July 2022). [CrossRef]
4. Zaidan, A.A.; Zaidan, B.B.; Hussain, M.; Haiqi, A.; Mat Kiah, M.L.; Abdulnabi, M. Multi-criteria analysis for OS-EMR software selection problem: A comparative study. *Decis. Support Syst.* **2015**, *78*, 15–27. Available online: <http://linkinghub.elsevier.com/retrieve/pii/S0167923615001347> (accessed on 9 July 2022). [CrossRef]
5. Sudhaman, P.; Thangavel, C. Efficiency analysis of ERP projects—Software quality perspective. *Int. J. Proj. Manag.* **2015**, *33*, 961–970. Available online: <http://linkinghub.elsevier.com/retrieve/pii/S0263786314001689> (accessed on 9 July 2022). [CrossRef]
6. Yazgan, H.R.; Boran, S.; Goztepe, K. An ERP software selection process with using artificial neural network based on analytic network process approach. *Expert. Syst. Appl.* **2009**, *36*, 9214–9222. Available online: <http://www.sciencedirect.com/science/article/pii/S0957417408008877> (accessed on 9 July 2022). [CrossRef]
7. Gürbüz, T.; Alptekin, S.E.; Işıklar Alptekin, G. A hybrid MCDM methodology for ERP selection problem with interacting criteria. *Decis. Support Syst.* **2012**, *54*, 206–214. Available online: <http://linkinghub.elsevier.com/retrieve/pii/S0167923612001170> (accessed on 9 July 2022). [CrossRef]
8. Bao, T.; Liu, S. Quality evaluation and analysis for domain software: Application to management information system of power plant. *Inf. Softw. Technol.* **2016**, *78*, 53–65. Available online: <http://linkinghub.elsevier.com/retrieve/pii/S0950584916300933> (accessed on 9 July 2022). [CrossRef]
9. Koh, S.C.L.; Gunasekaran, A.; Goodman, T. Drivers, barriers and critical success factors for ERP implementation in supply chains: A critical analysis. *J. Strateg. Inf. Syst.* **2011**, *20*, 385–402. Available online: <http://linkinghub.elsevier.com/retrieve/pii/S0963868711000400> (accessed on 9 July 2022). [CrossRef]
10. Abu Salih, B.; Wongthongtham, P.; Beheshti, S.M.R.; Zajabbari, B. Towards a Methodology for Social Business Intelligence in the Era of Big Social Data Incorporating Trust and Semantic Analysis. *Lect. Notes Electr. Eng.* **2019**, *520*, 519–527.
11. Wongthongtham, P.; Salih, B.A. Ontology and Trust based Data Warehouse in New Generation of Business Intelligence. *IEEE 13th Int. Conf. Ind. Inform.* **2015**, *Idc*, 476–483.
12. Arvidsson, V.; Holmström, J.; Lyytinen, K. Information systems use as strategy practice: A multi-dimensional view of strategic information system implementation and use. *J. Strateg. Inf. Syst.* **2014**, *23*, 45–61. Available online: <http://linkinghub.elsevier.com/retrieve/pii/S0963868714000055> (accessed on 9 July 2022). [CrossRef]
13. Kovtun, V.; Izonin, I.; Gregus, M. The functional safety assessment of cyber-physical system operation process described by Markov chain. *Sci. Rep.* **2022**, *12*, 7089. Available online: <https://doi.org/10.1038/s41598-022-11193-w> (accessed on 9 July 2022). [CrossRef] [PubMed]
14. Kovtun, V.; Izonin, I.; Gregus, M. Model of Information System Communication in Aggressive Cyberspace: Reliability, Functional Safety, Economics. *IEEE Access* **2022**, *10*, 31494–31502. [CrossRef]
15. Mendoza, L.E.; Grimán, A.C.; Rojas, T. Algoritmo para la Evaluación de la Calidad Sistémica del Software. In Proceedings of the 2nd Ibero-American Symposium on Software Engineering and Knowledge Engineering, Salvador, Brasil, October 2002.
16. Rincon, G.; Alvarez, M.; Perez, M.; Hernandez, S. A discrete-event simulation and continuous software evaluation on a systemic quality model: An oil industry case. *Inf. Manag.* **2005**, *42*, 1051–1066. [CrossRef]
17. Mendoza, L.E.; Pérez, M.A.; Grimán, A.C. Prototipo de Modelo Sistémico de Calidad (MOSCA) del Software 2 Matriz de Calidad Global Sistémica. *Comput. Y Sist.* **2005**, *8*, 196–217.
18. Kirkwood, H.P. *Corporate Information Factory*; John Wiley & Sons: Hoboken, NJ, USA, 1998; Volume 22, pp. 94–95. Available online: <http://search.proquest.com.bibl.proxy.hj.se/docview/199904146/abstract?accountid=11754> (accessed on 9 July 2022).
19. Bass, L.; Clements, P.; Kazman, R. *Software Architecture in Practice*; Addison-Wesley Professional: Boston, MA, USA, 2003; p. 528.
20. Callaos, N.; Callaos, B. Designing with Systemic Total Quality. *Educ. Technol.* **1993**, *34*, 12.
21. Humphrey, W.S. *Introduction to the Personal Software Process*; Addison-Wesley Professional: Boston, MA, USA, 1997.
22. Ortega, M.; Pérez, M.; Rojas, T. A Model for Software Product Quality with a Systemic Focus. *Int. Inst. Inform. Syst.* **2001**, 395–401.
23. Pérez, M.A.; Rojas, T.; Mendoza, L.E.; Grimán, A.C.; Procesos, D.; Bolívar, U.S. Systemic Quality Model for System Development Process: Case Study. In Proceedings of the AMCIS 2001, Boston, MA, USA, 3–5 August 2001; pp. 1–7.
24. Dromey, G. Cornering to Chimera. *IEEE Softw.* **1996**, *13*, 33–43. [CrossRef]
25. Kitchenman, B.A.; Jones, L. Evaluating software engineering methods and tool part 5: The influence of human factors. *ACM SIGSOFT Softw. Eng. Notes* **1997**, *22*, 13–15. [CrossRef]
26. Jelassi, T.; Le Blanc, L.A. DSS Software Selection: A multiple Criteria Decision Methodology. *Inf. Manag.* **1989**, *17*, 49–65.
27. Hostmann, B.; Oestreich, T.; Parenteau, J.; Sallam, R.; Schlegel, K.; Tapadinhas, J. Magic Quadrant for Business Intelligence and Analytics Platforms MicroStrategy Positioned in ‘Leaders’ Quadrant. 2013. Available online: <http://www.microstrategy.com/about-us/analyst-reviews/gartner-magic-quadrant>. (accessed on 9 July 2022).

28. Abelló Gamazo, A.; Samos Jiménez, J.; Curto Díaz, J. *La Factoría De Información Corporativa*; Universitat Oberta de Catalunya: Barcelona, Spain, 2014.
29. Rincon, G.; Perez, M. Discrete-event Simulation Software Decision Support in the Venezuelan Oil Industry. In Proceedings of the AMCIS 2004 Proceedings, New York, NY, USA, 22 August 2004. Available online: <http://aisel.aisnet.org/amcis2004/7> (accessed on 9 July 2022).