Master's degree thesis

LOG950 Logistics

"Green" supplier selection problem solving in information systems on the SAP-basis

Artsiom Latyshau

Number of pages including this page: 143

Molde, 22.05.2023

Molde University College
Specialized University in Logistics

Mandatory statement

Each student is responsible for complying with rules and regulations that relate to examinations and to academic work in general. The purpose of the mandatory statement is to make students aware of their responsibility and the consequences of cheating. Failure to complete the statement does not excuse students from their responsibility.

*Please complete the mandatory statement by placing a mark **in each box** for statements 1-6 below.*		
1.	**I hereby declare that my paper is my own work, and that I have not used other sources or received other help than mentioned in the paper.**	☒
2.	**I/we hereby declare that this paper** 1. Has not been used in any other exam at another department/university/university college 2. Is not referring to the work of others without acknowledgement 3. Is not referring to my previous work without acknowledgement 4. Has acknowledged all sources of literature in the text and in the list of references 5. Is not a copy, duplicate or transcript of other work	Mark each box: 1. ☒ 2. ☒ 3. ☒ 4. ☒ 5. ☒
3.	**I am aware that any breach of the above will be considered as cheating, and may result in annulment of the examination and exclusion from all universities and university colleges in Norway for up to one year, according to the Act relating to Norwegian Universities and University Colleges, section 4-7 and 4-8 and Examination regulations section 14 and 15.**	☒
4.	**I am aware that all papers/assignments may be checked for plagiarism by a software assisted plagiarism check**	☒
5.	**I am aware that Molde University College will handle all cases of suspected cheating according to prevailing guidelines.**	☒
6.	**I am aware of the University College's rules and regulation for using sources**	☒

Personal protection

Personal Data Act

Research projects that processes personal data according to Personal Data Act, should be notified to Data Protection Services (NSD) for consideration.

Have the research project been considered by NSD? □**yes** ☒**no**

- If yes:

Reference number:

- If no:

I hereby declare that the thesis does not contain personal data according to Personal Data Act.: ☒

Act on Medical and Health Research

If the research project is effected by the regulations decided in Act on Medical and Health Research (the Health Research Act), it must be approved in advance by the Regional Committee for Medical and Health Research Ethic (REK) in your region.

Has the research project been considered by REK? □**yes** ☒**no**

- If yes:

Reference number:

Publication agreement

ECTS credits: 30

Supervisor: Terje Lauvaas Andersen

Agreement on electronic publication of the master thesis

Author has copyright to the thesis, including the exclusive right to publish the document (The Copyright Act §2).

All theses fulfilling the requirements will be registered and published in Brage HiM, with the approval of the author.

Theses with a confidentiality agreement will not be published.

I hereby give Molde University College the right to, free of charge, make the thesis available for electronic publication: ☒**yes** ☐**no**

Is there an agreement of confidentiality? ☐**yes** ☒**no**

(A supplementary confidentiality agreement must be filled in)

- If yes:

Can the thesis be online published when the period of confidentiality is expired? ☒**yes** ☐**no**

Date: 22.05.2023

Preface

This thesis is the result of a Master's program education in Logistic Analytics at Molde University College – Specialized University in Logistics. The financial support was provided by the Diku Eurasia program project CPEA-2015/10061 "Development of a multidisciplinary Norwegian-Belarusian study program in Logistics Analytics promoting Bologna process reforms".

The purpose of the study is to provide a solution, that allows to choose the most appropriate supplier taking into account their greenness and ability to follow environmental standards. Despite the fact, that the solution is based on the SAP information system, it is supposed to make it flexible so that it could be easily replicated in other information systems.

Acknowledgment

I would like to thank my supervisor Terje Lauvaas Andersen for his guidance, support, advice, time, and patience provided during the work. I would like to thank Professor Bjørn Jæger for the organizing necessary communications and providing access to the training SAP system. I am very grateful to Professor Irina Vladimirovna Gribkovskaia and Professor Markov Sergey Viktorovich for their contribution to the development of the student exchange program. They have been supporting me in all education in Molde. I also want to thank Professor Arild Hoff for his kindness and responsiveness to manage master students' questions. I appreciate the help of Seniorrådgivere Anette Kristin Myrstad and Ragnhild Oddrun Ekren Brakstad with solving all administrative issues during education. Thanks to housing consultant Helene Gjerde for her goodwill and immediate reaction to any students' appeals. Many thanks to the administration of the university for the modern, comfortable, and innovative university.

Especial gratitude for financial support to the Diku Eurasia program project CPEA-2015/10061 "Development of a multidisciplinary Norwegian-Belarusian study program in Logistics Analytics promoting Bologna process reforms".

Many thanks to my parents for their support and faith in me. I hope to live up to their expectations.

I would like to express special gratitude to my Belarusian friends who have spent all this time in Molde with me. We have passed this path together supporting and helping each other. I'm really glad to be a part of the ESN during my studies. Thanks for the experience and friends. Thanks to all those people with whom we met during the training, I hope to continue communication after.

Artsiom Latyshau.

Abstract

The supplier selection problem is an essential part of the purchasing process. An appropriate decision influences the business's efficiency and competitiveness. The question of environmental security is sharp nowadays. Today's society and government are curious about following environmental and green standards. Moreover, the rapid development of technologies and information systems particularly enables to solve a variety of business problems more efficiently than it was previously.

The common purchasing process has its general steps and more or less may be represented in any IS. However, using information systems to solve supplier selection problems requires choosing appropriate criteria taking into account today's greenness requirements. The data structure, model architecture, SS methods, and criteria must be explored to solve the problem in IS effective. This study tries to figure out those questions.

Contents

List of abbreviations

IS – Information system

SS – supplier selection problem

ERP – Enterprise resource planning

CRM – Customer relationship management

SCM – Supply chain management

WMS – Warehouse Management System

BI – Business intelligence

ESG – Environmental, Social, and Governance

MCDM – Multi-criteria decision making

DEA – Data envelopment analysis

AHP – Analytic Hierarchy Process

FAHP – Fuzzy analytic hierarchy process

TOPSIS – Technique for others method Reference by Similarity to Ideal Solution

PROMETHEE – Preference Ranking Organization Methods for Enrichment Evaluations

MD – master data

FM – functional module

API – Application Programming Interface

EDI – Electronic Data Interchange

List of tables

List of figures

List of attachments

1 Introduction

Effective sourcing and management of supplier information is essential to successful supplier selection (SS). The right choice of suppliers is the best way to positively influence the cost, quality, and efficiency of the supply chain and one of the fastest ways to reduce costs and increase revenue. High attention is paid to the supplying of materials and services in logistic systems planning. Supplier management is an extremely important problem influencing the performance of the business. To provide materials and services on an appropriate level companies are usually choosing between several alternatives, or in other words, suppliers [1, pp. 193-194]. There are several factors could be taken into account to evaluate and choose an appropriate supplier [2, pp. 10411-10415]:

- Economic criteria,
- Environmental criteria,
- Social criteria,
- Resiliency criteria.

New tendencies lead to new problems as well, which makes the list of usual SS criteria not enough to suit modern conditions. The greatest example is the climate change problem. This problem and green economy approaches are becoming more popular nowadays. In order to evaluate the suppliers in that terms, factors such as recyclability, reusability, and green design are being used along with economic and social parameters [3, p. 1400].

European Union takes care of the environment by establishing environmental policies such as RoHS (Restriction of Hazardous Substances Directive) and WEEE (Waste Electrical and Electronic Equipment Directive). The RoHS directive restricts sellers, producers, and distributors of electronic facilities and hardware the placing on the EU market. Those directives forbid electronic equipment that exceeds the agreed levels of hazardous materials and components insight, such as lead, cadmium, mercury, hexavalent chromium, and so on [4].

According to Norway's strategy for developing a green and circular economy [5], the government plans to make efforts to support and improve Norway’s green industry sector. Particularly they plan to improve Norway's green competitiveness, work on non-toxic material cycles, and harness the potential of digitalization to make product information and market information more accessible. The key actions of the “*Action plan of increasing the* *proportion of green public procurements and green innovation*" [6] include promoting solutions for zero or low emissions, avoiding to use of chemicals, and increasing the proportion of green public procurement. Gathering the information from the website www.regjeringen.no about the budget for security and fair distribution [7], the government suggests procedures for emission-reducing and business improvements that do not make damage the climate. One of them is increasing the tax on CO2 emissions by 21%. Such state actions may lead suppliers and customers in Norway to follow the "green" direction, adopt their business, and choose partners more precisely.

The SS problem involves selecting the most appropriate supplier(s) from a range of potential suppliers based on different criteria. Cost, quality, delivery time, and environmental behavior may be considered as criteria for the evaluation. Basically, the SS problem is a complex and dynamic problem that requires a systematic and structured approach. It involves the evaluation of both tangible and intangible factors, such as supplier reputation, communication skills, and cultural fit. Often it's quite difficult to evaluate intangible factors. Moreover, it is important to consider the interdependencies among the criteria, as improving performance in one area may have unintended consequences in another. The problem of SS becomes more popular from 2000 to 2020 in the scientific society as well. The number of works written by the SS problem using the Data Environment Analysis (DEA) increased that time [3].

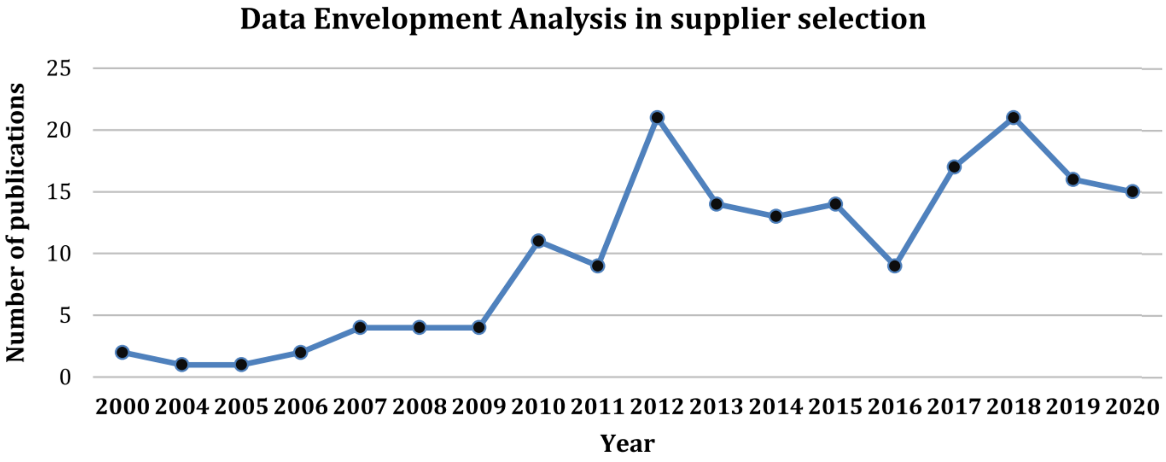
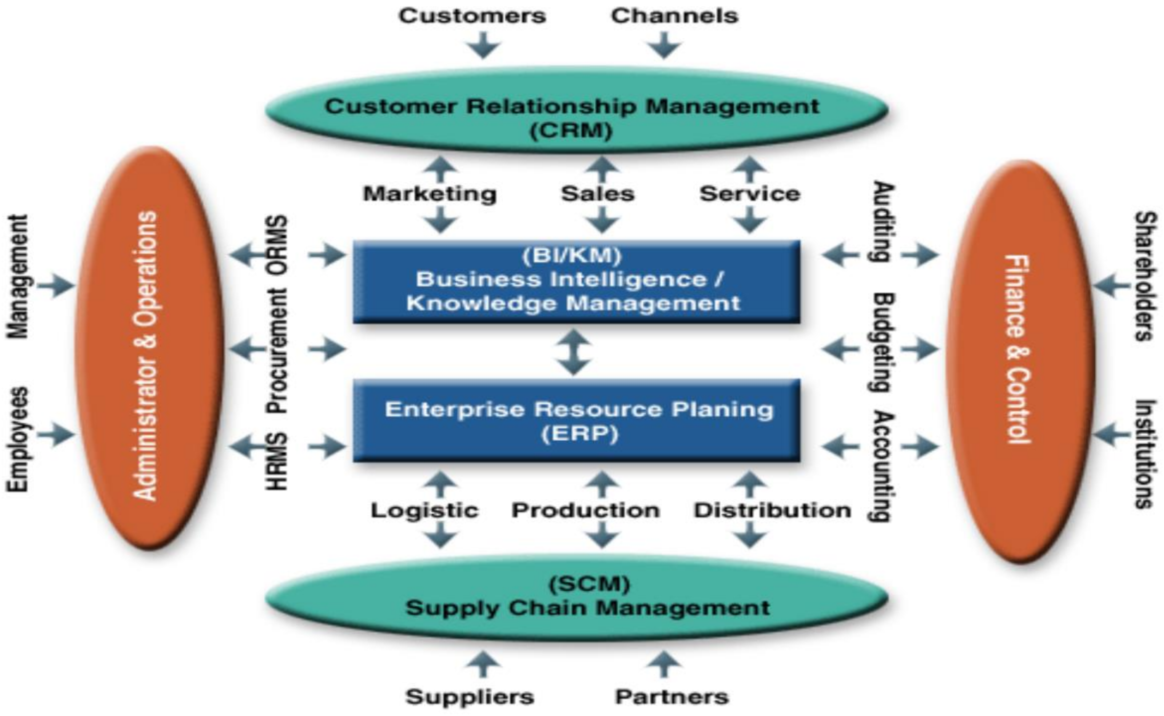


Figure 1: Dynamic of publications of DEA in SS, 2000 – 2020 [3, p. 1408]

The SS problem is described pretty well in the theoretical cases in the works [2] and [8]. But there is a lack of articles, that has been written about using and implementing approaches of SS in information systems.

Here it needs to clarify what is meant under the term *information systems* (IS). IS collects, processes, stores, analyzes, and distributes information for a specific purpose [9, p. 12]. It may help to make business decisions, and transform the data to a viewable and readable form, it could be connected with the other IS either inside of the company or outside between other companies (**Figure 2**). The goal of IS is to produce the right people with appropriate information at the right time, in the right amount and format. The main known ISs are:

- Enterprise resource planning (ERP)
- Customer relationship management (CRM)
- Supply chain management (SCM)
- Business intelligence (BI)
- Warehouse management
- etc.



Customers
Channels
Customer Relationship Management (CRM)
Marketing
Sales
Service
Management
Employees
Administrator & Operations
HRMS Procurement ORMS
(BI/KM) Business Intelligence / Knowledge Management
Enterprise Resource Planing (ERP)
Auditing Budgeting Accounting
Finance & Control
Shareholders
Institutions
Logistic
Production
Distribution
(SCM) Supply Chain Management
Suppliers
Partners

Figure 2: ERP, SCM, BI & CRM Integration [10]

Taking into account described increasing importance of the environmental factor and a lack of researches about SS models implementation, it could prove the relevance of the topic. The topic could be interesting for companies that use IS. Almost half of the million companies across the whole world are using ISs (ERP and CRM in particular) in their activities [11].

ERP and CRM are the most popular ISs. The market of ERP is competitive, and it consists of a huge product variety. Using the data from [11] it's possible to calculate the Herfindahl-Hirschman index [12] for the ERP market. Its value is 418 (less than 1500), which means that the market is unconcentrated and competitive.

SAP company takes the main share of the ERP market as almost 14,5% of the business use SAP's ERP (SAP ERP, SAP R/3; S/4 Hana **Figure 3**).

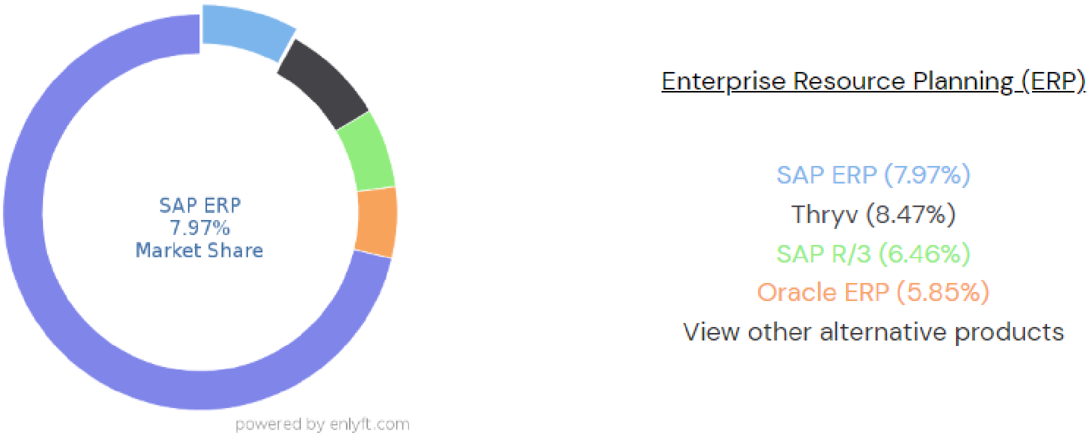


Figure 3: ERP Market Share [11]

The CRM market has the clear leader company "Salesforce.com", which has about 40% of the CRM market (**Figure 4**).

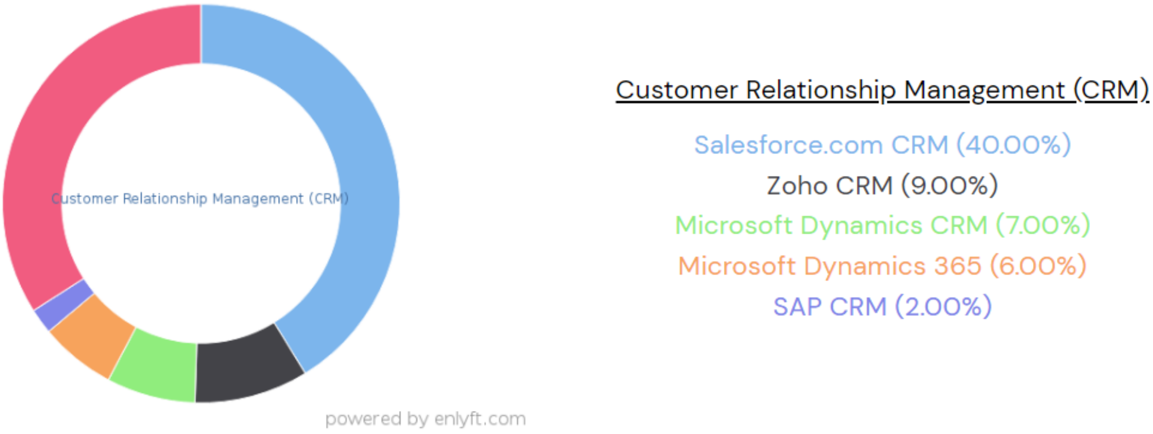


Figure 4: CRM Market Share [11]

enlyft.com is a business intelligence platform that provides insights and data on the technology stack used by companies around the world. The platform uses advanced data analytics and machine learning algorithms to gather information on the software applications, technologies, and vendors used by companies, as well as their IT budgets and spending trends. The platform offers both free and paid plans, with the paid plans providing additional features and data access. Only free data is used for this work. Even though enlyft.com primarily has information about huge and well-known companies and obviously can’t collect the whole data about all companies that used ISs, the amount of collected information seems enough to prove the prevalence of IS usage in modern businesses.

Therefore, this thesis explores the possibility of implementing a SS model into an IS taking into account environmental or, in other words, “green” components. The implementation is made on the SAP basis because of two reasons: SAP is one of the main representatives in the IS market and access to the system was provided by Molde University College.

2 Problem description

2.1 General formulation of the problem

A general problem of SS may be formulated as follows: select the best supplier, that can suit customers' requests as much as possible. Ideally, one supplier should be better than other ones in all aspects, but basically, it's impossible, so that variety of criteria and corresponding weights are used to make an appropriate choice. Those criteria may be different, for example, price and quality, financial reliability, lead time, and so on. Usually, businesses decide by themselves which criteria are most important, taking into account existing state rules and low regulations. Specific constraints may be established as well, such as budget limits or the need to meet particular environmental criteria.

The SS process goal is to answer 4 main questions [1]:

- what products or services to order,
- in what quantity should be ordered,
- from which suppliers to order,
- in what time periods to order,

SS problem may be described more specifically by dividing it into two aspects. The first aspect consists of collecting the suppliers to be dealt with. This aspect regulates the scale of the problem and may influence criteria weights. The second aspect is the selection process itself among the given alternatives [13, pp. 3-4]. Those two aspects may be related by the simple example. For instance, having two suppliers on the market usually means that the price and quality are almost the only criteria. But in the competitive market customers usually pay attention to other criteria, like delivery, finance reliability, service, supplier greenness, and so on.

In the current business environment, purchasing has become a determining factor in the creation of added value for products. Purchasing is also a vital factor in ensuring the profitability and survival of a company in the market. Research on SS is very numerous, and the scientific study of this problem began quite a long time ago: the date of first publication back to the 1960s. Weber, Current, and Benton (1991) and Ghodsypour and O'Brien (1998) provide a comprehensive review of past research [4].

Nowadays scientists and individuals who are dealing with the SS problem pay attention to Green aspects of the SS. Green aspects may be considered as environmental directives or awareness raising. Green SS has prompted the creation of numerous evaluating approaches [14, p. 3]. SS approaches will be overviewed in the following chapters.

This work tries to enlarge the basic model evaluating Price, Quality, and Service, by adding new elements to take into account the supplier's greenness. It allows to find out which parameters may be used to evaluate supplier greenness, and how those parameters may be stored and processed. A brief overview of elements changing is represented in **Figure 5**.

Some different tools or environments could be used to perform SS. Here tools and environments mean that SS can be made in IS (ERP, CRM), on Internet portals and supplier aggregators [15], cloud solutions [16], or even well-known Microsoft Excel [17]. The choice depends on the business scale and sphere, companies' recourses, and technology availability. Which tool should be used depends on its advantages and disadvantages as well.

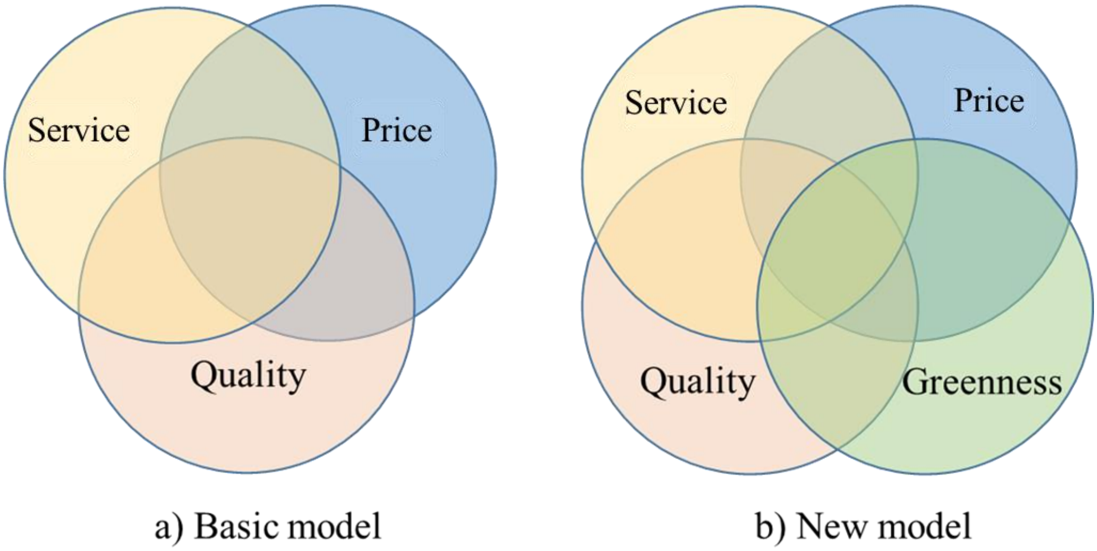


Figure 5: Key elements for an ERP model: a) basic model [18], b) new model

No less important to choose a relevant SS model, taking into account companies' needs and sphere of business. The market of ISs is too wide, each system has its restrictions and limitations in terms of database structure, new functionality implementation, program language peculiarities, and so on. All of this should be taken into account before choosing the approach of SS.

2.2 Limitations and simplifications

The work describes the approaches to solving the case when a company needs to choose a supplier during the purchasing process (**Figure 6**). **Figure 6** is made in BPMN[1] notation and illustrates generally the purchasing process. It starts from the needs definition actions, then continues with communication with suppliers, and ends by performing purchasing.

This work explores the situation when a company needs to choose the best supplier when there are no current contracts: either all existing contracts are finished, or it needs to purchase a new product that wasn't purchased previously (Figure 6, green box). Till this moment the exact needs are determined and the business knows what particular product it needs. Usually, companies organize tenders, which explain their needs: products, characteristics, an acceptance application date, and so on. During the thunder customers collect information about suppliers via requests for information, quotations, and proposals. More precisely this part of the process is explained in the reference [19, p. 3]. So the problem here is to choose an appropriate supplier according to some selection criteria, that suit companies' strategies, business needs, and wishes, state requirements, and so on [1, p. 194].

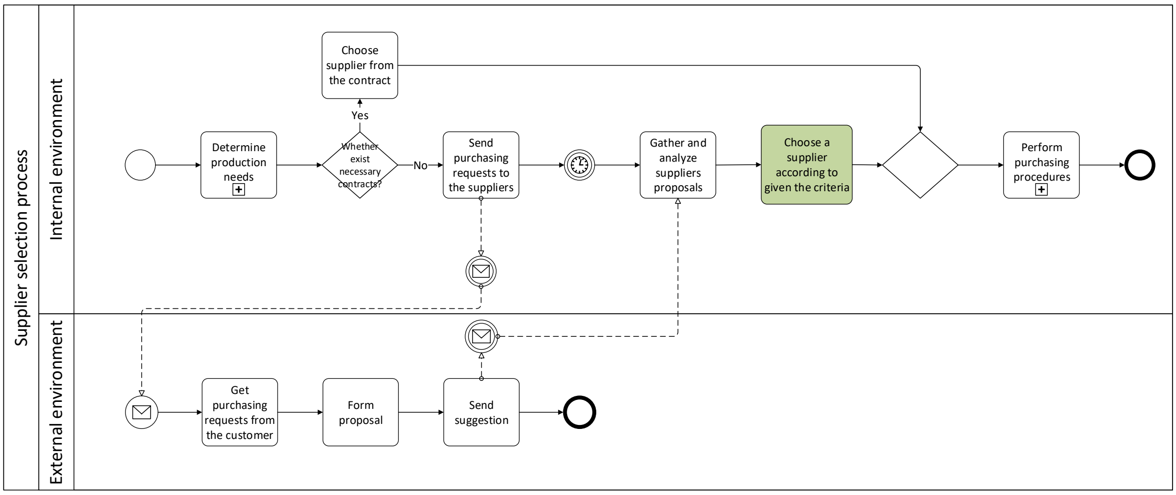

Supplier selection process
Internal environment
External environment
Choose supplier from the contract
Yes
Determine production needs
Whether exist necessary contracts?
No
Send purchasing requests to the suppliers
Gather and analyze suppliers proposals
Choose a supplier according to given the criteria
Perform purchasing procedures
Get purchasing requests from the customer
Form proposal
Send suggestion

Figure 6: Supplier selection process (BPMN notation)

To simplify the problem, it assumes that a company evaluates suppliers by two approaches. Firstly, the company takes into account general supplier information that is independent of the product, like finance reliability (credit rating, current ratio), quality-

[1] The business process model and notation is the standard for business process diagrams. It is intended to be used directly to design, manage, and implement business processes. At the same time, BPMN diagrams can be converted into software process components. BPMN has an easy-to-use flowchart-like notation that is independent of any particular implementation environment [65].

related certificates, management quality, ISO certification, and so on. Secondly, the company evaluates the quality of only one particular product, because the company knows its exact needs till the moment of purchase. This is the main simplification of this work.

Basically, evaluating a supplier based on one product may not take into account other factors such as the supplier's financial stability or ability to suit some international standards. That’s why general supplier information is supposed to be taken into the model as well in this work. Where such an approach may be used and which vector the model could be enlarged see **Chapter 7.1.**

This work is exploring the possibility of SS on the SAP system basis. The reasons for choosing SAP as the developing environment are described in **Chapter 1**. SAP applications are developed on the ABAP programming language [20]. This requires an exact data structure typing or creating extra functions of methods that allow to the definition of necessary data type.

The basic interface for performing supplier evaluation is produced by SAP. That means that the appearance of transactions is not supposed to be changed within the current thesis even if the tool looks not modern enough. The appearance may be changed by SAP configurations but this is not the main topic. Basically, all IS produce functionality to more or less maintain their appearance. The transactions view may be preconfigured and explained briefly where it needs during creating the algorithm. This is supposed, that the data required for the supplier evaluation is stored in only one IS (SAP Enterprise Resource Planning) [21]. Other systems like CRM, SCM, WMS, and so on are not considered in this work.

As all calculations are considered to be done by the IS, it requires numeric data. That’s why some criteria like companies’ capability of quality management, social responsibility, and internal control process are hard to be estimated. It’s quite complicated to collect that data from the suppliers. Some data may be closed by the privacy policy of the information disclosure; some suppliers may give the wrong data. Moreover, there are no common standards on how to collect and adjust such data on a numeric basis. That’s why the preferences have been given to the criteria that could be represented and stored numerically and could be ranked or explained as Boolean parameters (true/false) [22].

2.3 Research questions

Taking into account described problem definition, restrictions, and limitations, the following research questions should be answered to explore the topic properly:

Table 1: The main statements and outcomes of the research.

Research question	Motivations	Expected outcome
What criteria could be taken into account to evaluate suppliers' "green components" in an information system?	An attempt to fill the lack of research on the topic, and to expand on previous research on the topic. Provide a list of criteria that can be used in business from the information system.	The list of categories and criteria that could be taken into account.
What is the process of implementing a supplier selection method according to the available information system?	There are lots of articles written about the methods of supplier selection, but there is a lack of works that explain more deeply how to implement any existing approach to an information system.	Working interface in an information system that allows evaluating suppliers based on the determined criteria, including supplier greenness.
How necessary data may be stored in a given information system?	To be able to automate supplier evaluation in the information system, it is necessary to ensure the correct storage of the necessary data. You need to show how the data used in supplier evaluation can be stored in the system or ranked (for non-numeric values).	Explained the relationship between required data and information system business objects, tables, and views.

3 Literature review

This thesis has been started based on Research Design [23] task and Proposal [24] that were written during the master's program at Molde University College

3.1 Supplier selection problem

To find relevant sources of literature, articles, books, and other material, the tool «Publish or Perish» [25] is used. The requests to Google Scholar were made via this tool. It allows collecting articles by title, keywords, authors, publication name, year, and so on. This tool allows saving requests to have the possibility to analyze them. Some search results are given in the **Table 2**.

Table 2: Searching of papers for observation

Title words	Keywords	Period	Numbers of papers
Supplier selection	-	2010 – 2023	> 1000
Supplier selection	ERP, information system		> 200
Supplier selection	SAP		61
Supplier selection	Sustainable ERP		>200
Green supplier selection			>200
Green supplier selection	SAP		2
Green supplier selection	Sustainable ERP		24
Carbon footprint	Calculation	2000 – 2023	> 1000
Supplier selection implementation	Business		23
Supplier selection implementation	Information system		25
Supplier selection SAP	-	-	0
Supplier selection ERP	-	1995 – 2023	6
Sustainable ERP			60

According to search results represented in **Table 2**, the topic of SS is quite popular (more than 1000 works were written in the last 13 years). As was mentioned before, the interest in this topic becomes more and more popular (Figure 1). The topic related to sustainability and green parameters are popular as well (green SS and carbon footprint topics have more than 200 and 1000 works respectively). But there is a lack of works on the implementation and usage of theoretical methods in practice (no more than 100 in each category, related to IS, implementation, ERP, and SAP). So, from one point of view there is a lot of information that could be used to make good work about supplier selection, from the other point of view there are some uncovered topics about the practical use of existing models that could be explored.

By analyzing requests, the following articles were taken as the basis. One of the main articles considered is the work of Dutta, Jaikumar, and Arora [3]. It gives a review of the main-known works about SS during the period of the first two decades 21st century. It highlights the main approaches used in supplier selection and produces a dynamic of changes in the interest toward this topic. The article gives an understanding of the status of SS to the 2020 year.

The book "*Introduction to Logistics Systems Management*" [1] gives an explanation of the importance of the SS problem for the business. The book generally explains the main question of the supplier selection problem. It briefly describes selection criteria and produces some simple examples of calculating coefficients to select the best supplier.

Behzad Masoomi and his colleagues [14] give a greatly relevant literature review on the topic of the current thesis. They discover the articles about the general SS problem, give links for the articles about the green SS problem, and share the observation of approaches and models of the SS and the main criteria that could be used to evaluate suppliers.

There are some state sources were explored to find information about the tendency of the green economy and its accompanying trend. The websites electricenergyonline.com [5] and www.regjeringen.no [7] are among them. The Norwegian Agency for Public and Financial Management's action plan of increasing the number of green public procurements [6] was explored as well to estimate the relevance of green components for the SS problem in today's reality.

3.2 Supplier selection models

Jane Mbiatem, Atour Taghipour, and Beatrice Canel-Depitre [19] make a comparison of existing models for the SS. Authors run a study case through the main-popular multi-criteria decision-making methods (DEA, AHP/FAHP, Fuzzy, TOPSIS, and so on) rather than cost/price single criteria. Unfortunately, they didn’t produce enough information about the calculation. But still, it gives an overview of those methods.

The supplier selection models, such as DEA, AHP/FAHP, Fuzzy, and TOPSIS, offer plenty of approaches for solving the SS problem. The evaluation and prioritization of suppliers are based on various criteria. By summing up the information from Dutta’s, Jaikumar’s, and Arora’s article [3], it's possible to conclude, that DEA analyzes the suppliers' efficiency, AHP gives a hierarchical structure-based evaluation, FAHP considers the uncertainty, and TOPSIS evaluates candidates taking into account how far are they from the ideal solution. The chosen model depends on the business requirements and the data necessary availability. An overview of the existing SS models is given in **Chapter 4.1**.

Nitin Sachdeva, Avinash K. Shrivastava, and Ankur Chauhan [26] divide the methodologies of SS by relevant criteria. It has an overview of works from different authors collecting the criteria that were used in different models. The main focus of this work is on fuzzy methodologies. The weights determination methodology is described here as well.

The method PROMETHEE II was taken as the basis of the model of this work. The information about the formulas and approaches to calculation was taken from N. Agrawal’s article [27]. It describes how to implement data envelopment analysis (DEA) with numerical illustrations. There are 3 cases described in the work. A general comparison, advantages, and disadvantages are represented there (**Attachment 2**). The article explains the main objectives of SS approaches as well. Despite some misprints in the tables with calculations, the article gives a deep explanation of each PROMETHEE II step. On the basis of understandable cases, the authors describe all formulas, matrices, and calculations required to solve the problem. The article includes an overview of the main-known methods and briefly describes the advantages and disadvantages of each one. Exactly this article became the decisive factor to choose the PROMETEE II method.

The article of Vahid Balali, Banafsheh Zahraie, and Abbas Roozbahani [28] provides a comparison between the AHP method and the PROMETEE family of methods (I and II). Arguments from that article persuaded that PROMETEE II’s advantages are valid to use this method for the implementation in an IS.

3.3 Supplier selection criteria

Collection and observation of the main basic SS criteria were made in Lyes Benyoucef’s, Hongwei Ding’s, and Xiaolan Xie’s article [13]. The article started with the observation of Dickson's[1] SS criteria list, which was made in the 1960s. Then authors enlarge the list according to the market situation of the early 2000s. The article was written 20 years ago so that it doesn't have green supplier criteria but still it has an appropriate basis to form the list of selection criteria. Charles A. Weber [29] explains Dickson's SS criteria as well.

The most cited article about the green SS problem is the Amy H.I. Lee's, He-Yau Kang's, Chang-Fu Hsu's, and Hsiao-Chu Hung's one [4]. It was written in 2009, 6 years after the article [13]. The article from 2009 explains the importance to take into account the suppliers' greenness. That importance is explained via new low regulations, concerns about environmental issues, and customer preferences toward clean and environmental suppliers. It seems, that the science community became paying more attention to the ecology and environment in the first decade of the 21st century. This article suggests green categories and green parameters for those categories that could be collected and observed to evaluate suppliers' greenness. The idea to use categories and related parameters in the model for the current thesis was taken from this article. As the solution method, the article describes the model with fuzzy parameters. This implies gathering information and expert opinions about the significance of the parameters. The approach is well explained but seems not relevant to this thesis.

The work of Ghamari, Roya, Mohammad Mahdavi-Mazdeh, and Seyed Farid Ghannadpour [2] represents the types of criteria that are taken into account in the steel industry to choose a supplier. The article describes popular approaches for SS as well. It is worth paying attention to the year of the article: 2020. So, the criteria accumulated in the article are quite relevant to be used in the thesis model.

A huge observation of green SS criteria was made by A. Konys [30]. While lots of papers are written about improving existing models or discovering new ones, this article concentrates on identifying sets of criteria and capturing the domain knowledge. The article has well-formulated research questions, explained with motivations and outcomes. This was taken as a sample for the current research.

Srikant Gupta, Prasenjit Chatterjee, Morteza Yazdani, and Ernesto DR Santibanez Gonzalez [8] describe criteria that should be taken into account to satisfy the Green Selection

[1] Dickson v. Hausman is the researcher of XXth century. He was the first who started to explore supplier selection problem and formalized supplier selection criteria [author's note].

approach. The article tries to prove, that the best supplier is not one who delivers the product as cheaper and as faster as possible, but the one who produces its commodity with the smallest amount of pollution and energy consumption.

One of the ways to keep the balance between saving resources and collecting green data is to use product “carbon footprint”. A “carbon footprint” has lots of definitions. The book “Ecological Economics Research Trends” [31, p. 4] gives several definitions, that could be related to the different spheres of life. In terms of making products, a “carbon footprint” could be described as the total amount of CO_2 and other greenhouse gases, that were created during the whole life cycle of the product. Amit Kumar, and Vipul Jain [32] explain the approach of products carbon footprinting. The article presents the model that takes into account CO_2 emissions of product production during supplier evaluation. The guidance for smaller carbon footprinting impact [33] was observed to have an understanding of units of measure of carbon footprint.

Gathering the information about "carbon footprint" the idea to explore existing standards about its calculation had happened. The presence or absence of one or the other standard may be considered as a selection criterion. There are several standards how to calculate the carbon footprint according to P. J. Pandey [34, p. 143]:

- GHG protocol of World Resource Institute (WRI)
- ISO standards
- Publicly Available Specifications-2050 (PAS 2050) of British Standard Institution (BSI)
- 2006 IPCC Guidelines for National Greenhouse Gas Inventories

Among the others, ISO standards have the clearest structure, purpose, and documentation. There are lots of such standards, described rules, and regulated businesses’ behavior in various markets. The whole range of ISO standards is represented on its website [35]. For instance, the purpose of ISO 14001 is to provide organizations with rules and approaches according to which the environment protecting and responding to changing environmental conditions should be performed [36].

3.4 Information system documentation

The book “Introduction to Information Systems” [9] explains the general principles of ISs, it discovers both sides business and technical, providing examples with business cases. One of the most valuable chapters of the current work is Chapter 5, where data and knowledge management are explained. It introduces approaches for data management and explains database approaches in general and in an IS.

Baymou [10] explores the significance of ERP systems generally and in supply chain processes. It describes the impact of ERP on business, competition, and nature [10, pp. 4-5]. The integration between different IS, like CRM, SCM, BI, and so on, is explained understandably. This article is one of the few, that compare the real products of real companies (SAP and ORACLE).

To effectively integrate SS models in ISs an appropriate data model is required. According to Chin-Tsai Lin [18] ERP (here ERP is a part of the company's ISs) data model should have consisted of three data elements: time, cost, and quality. To gain this data it needs to have an excess for both master data[1] and transactional data[2].

The website enlyft.com [11] provided statistics and distribution about the market share of different ERP systems. It describes the ERP systems by the number of companies, by regions, and by companies' size. This data was one of the main factors to use the SAP environment for the model of the thesis.

There are some basic SAP resources were used to explore standard functionality for supplier evaluation and make corresponding APAB development in the system. SAP Help Portal [37] was used to explore the functionality of vendor evaluation in the SAP Material Management (MM) module. It explains standard approaches, functions, and models for supplier evaluation. It provides the transaction codes to perform evaluation, gain evaluated suppliers' reports, how to configure necessary criteria and sub-criteria. SAP Help Portal was used to figure out required ABAP function modules, syntaxes, rules, and the most common errors and approaches.

3.5 ESG business example

This work explores the potential of the implementation of well-known SS methods into existing IS. This means that the work is exploring this potential in a training system without direct cooperation with any businesses or companies. But some real data and the

[1] Master data is all the data that has an essential meaning to the running of companies and businesses. It includes the following objects: places, people, and things such as materials or products. Usually, master data is a small percentage of all business data, but at the same time, it is the most valuable and complicated one in a company [63].

[2] Transactional data is data that cannot be changed. Data is created as a result of a business user performing a business operation, such as creating a purchase order, creating a process order, creating a GL entry [64].

information about SS business process were collected. One unnamed company[1] follow ESG rules that help management make decisions.

ESG (Environmental, Social, and Governance) is a framework that allows businesses to understand how other organizations manage their activity related to environmental, social, and governance criteria (ESG factors). ESG takes the holistic view that sustainability extends beyond just environmental issues [38].

According to the process, each vendor must respond to the questionnaire, which is made according to the ESG methodology. The questionnaire consists of the following categories (the whole questionnaire is represented in the **Attachment 1**):

- General
- Environment
- Health, Safety, and Wellbeing
- Human and labor rights
- Non-discrimination
- Modern Slavery and Child labor
- Freedom of association & Collective bargaining
- Wages & Working Hours

On the basis of the venders' responses, the special employees assign corresponding scores to the answers. Then the calculations in Excel are made. Although the company has SAP licenses and corresponding ISs, the process of collecting and evaluating information is made in Excel with lots of employee attraction. The implementation that is made in this work might facilitate such processes.

Based on this information the vision of the process was made. Information about the usage of an ESG approach in practice was gathered. It helped to understand the real criteria of the suppliers' evaluation process and formulate the model for this work.

[1] A company that shared an ESG list of questions insisted to remain anonymous. The name of the company does not affect the current work.

4 Methods and working environment

4.1 Supplier selection method

The choice of the right suppliers and the correct distribution of orders are two of the main strategic decisions in the process of SS [3]. Some different complex methods and techniques exist to solve the problems. Those decision-making approaches are vital for businesses to match the necessary competition level and keep for the corporate and business strategy.

There are several methods for SS: a technique for order of preference by similarity to ideal solution (TOPSIS), analytic hierarchy process (AHP), data envelopment analysis (DEA), etc. [3, p. 1403]. Businesses in general and IS end-users, in particular, want algorithms and methods to work fast and give the result immediately. So it needs to evaluate algorithms in terms of their speed and efficiency.

There are modern approaches for SS considering the environmental or so-called “green” part of suppliers. The environmental impact, standards, and efficient product design become more popular; governments become more strict in checking and enforcing environmental criteria. So companies need a green supplier assessment approach to be competitive and to rapidly changed government requirements and standards.

Multi-criteria decision-making (MCDM) is a sequence of operations, that helps to choose an alternative from multiple and conflicting variants by evaluating multiple conflicting criteria. SS via MCDM collects information about companies’ preferences, which helps them to choose the best or to some extent optimal vendors. The methods, that could be used:

- Analytical hierarchical process (AHP): technique, that was developed to analyze complex decisions.
- Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS): simple, but efficient model, that measures similarity or differences with the perfect solution. The best solution is the one that is at as greatest as the possible geometric distance from the negative-ideal solution and at as shortest as possible geometric distance from the positive-ideal solution.
- Artificial neural network (ANN): considers examples, without being programmed according to the task-specific rules.

- Data envelopment analysis DEA: evaluates objects (in the case of the work suppliers or vendors) by finding the best alternatives by comparing each object with others.
- etc.

Basically, the AHP approach is used to solve a complex problem. The approach involves six general steps [4]:

1. Define the unstructured problem and state the objectives and outcomes.
2. Decompose the complex problem into a hierarchical structure with decision elements (criteria and alternatives).
3. Employ pairwise comparisons among decision elements and form comparison matrices.
4. Use the eigenvalue method to estimate the relative weights of decision elements.
5. Check the consistency property of matrices to ensure the judgments of decision-makers are consistent.
6. Aggregate the relative weights of decision elements to obtain an overall rating for the alternatives.

PROMETHEE method is a simple Multi-Criteria Decision Method that suggests a ranking approach compared to other multi-criteria methods [28]. To implement the PROMETHEE I method, there are two types of information required. First is the weights for the criteria, and the second is the decision-making preference function [39].

Comparing PROMETHEE I and PROMETHEE II methods, the first one provides a partial ranking of the decision alternatives, and the second one provides a full ranking of the alternatives. In this thesis, the method PROMETHEE II has been used to fit the following requests:

- simplicity (this is supposed, that the method is easy the implementation)
- robust (may work with data of different scales and types)
- speed (IS users are not going to wait while the algorithm is working)
- get the full ranking list (to have the possibility to create the supplier ranking list)

The main problem was to choose one of the following alternatives: use the difficult method that was developed to analyze complex decisions losing in speed and simplicity or use a simpler and faster method losing some quality. From a scientific point of view, the first option is better. But from the business users' point of view, it’s better to have a tool that allows get fast results and make difficult decisions between close alternatives on the administrative level.

PROMETHEE II method could be explained in five steps (more deeply all steps are described in the reference [27], chapter “PROMETHEE II method”):

- **Step 1**: normalize the data.

Basically, the evaluation data has a different scale. For instance, the price could be given in thousands or millions, while the lead time is in a couple of days. To compare such criteria normalization is required. As the model is multi-criteria, supposed that the behavior of different criteria influences the supplier’s score differently. Supplier’s score may rise while non-beneficial parameters like price are decreasing and beneficial parameters like finance rating are increasing. So, there are opposite formulas used to normalize various types of data.

Beneficial criteria:

$$R_{ij} = \frac{[x_{ij} - min(x_{ij})]}{[max(x_{ij}) - min(x_{ij})]} (i = 1, 2, 3 \ldots n; j = 1,2,3 \ldots m) \qquad (1)$$

Non-beneficial criteria:

$$R_{ij} = \frac{[max(x_{ij}) - x_{ij}]}{[max(x_{ij}) - min(x_{ij})]} (i = 1, 2, 3 \ldots n; j = 1,2,3 \ldots m) \qquad (2)$$

- **Step 2**: Evaluate the relative differences between the i-th alternative and the other alternatives. This is required to determine which alternative is superior.
- **Step 3**: Use the preference function to create a preference matrix.

Several types of preference functions may exist: linear preference, level criterion, quasi-criterion, Gaussian criterion, and so on. To simplify the model, the following preference function is used in the work:

$$F_j(a,b) = \begin{cases} 0, if\ R_{aj} \le R_{bj} \\ R_{aj} - R_{bj}, if\ R_{aj} \ge R_{bj} \end{cases} \qquad (3)$$

a, b – the alternatives. In case of SS problem – suppliers.

- **Step 4**: Include weights in the calculation.

The multi-criteria preference index for each pair of the supplier is calculated by taking a weighted average of the preference functions, with the aggregated preference function serving as the weighting mechanism:

$$\prod(\boldsymbol{a}, \boldsymbol{b}) = \frac{[\sum_{j=1}^{n} \boldsymbol{W_j} \times \boldsymbol{F_j(a,b)}]}{\sum_{j=1}^{n} \boldsymbol{W_j}} \qquad (4)$$

- **Step 5**: Identity the positive "leaving" and negative "entering" outranking flows.

  Positive:

  $$\Phi^{+}(a) = \frac{1}{m-1}\sum_{b=1}^{m}\pi(a,b)\ \ (a \neq b) \tag{5}$$

  Negative

  $$\Phi^{-}(\mathbf{a}) = \frac{\mathbf{1}}{\mathbf{m-1}}\sum_{\mathbf{b=1}}^{\mathbf{m}}\boldsymbol{\pi}(\mathbf{b},\mathbf{a})\ \ (\mathbf{a} \neq \mathbf{b}) \tag{6}$$

- **Step 6**: calculate the net flow for the alternatives:

  $$\Phi(\mathbf{a}) = \Phi^{+}(\mathbf{a}) - \Phi^{-}(\mathbf{a}) \tag{7}$$

- **Step 7**: choose the best alternative.

  The higher net flow $\Phi(a)$, the higher rank has a given alternative.

So here is the step sequence with formulas that must be performed according to the PROMETEE II method. All these steps should be programmed during implementation. All those formulas look simple enough to create a fast and understandable algorithm in the ABAP language.

4.2 Criteria definition

To create a model that is relevant for solving real business tasks it needs to define appropriate criteria and sub-criteria. The first glance at suppliers' selection criteria was made based on the G. W. Dickson work [1]. He identified 23 different evaluation criteria and weights in 1966 (**Table 3**). The data is gathered from the survey based on asking 273 American managers from the National Associations of Purchasing Managers [1, p. 194]. The factors are divided into 4 categories: extremely important, considerably important, average important, and slightly important.

Table 3: Supplier selection criteria according to G. W. Dickson [29]

Rank	Factor	Mean rating	Evaluation
1	Quality	3.508	Extreme important
2	Contract terms for delivery	3.147	
3	Performance history	2.998	
4	Guarantee terms	2.849	Considerable important
5	Structural and manufacturing capacity	2.775	
6	Cost	2.758	
7	Technical capacity	2.545	
8	Financial position	2.514	
9	Conformity to the procedures	2.488	
10	Communication system	2.426	
11	Reputation	2.412	

12	Business attractiveness	2.256	
13	Management and organization	2.216	
14	Operative controls	2.211	
15	Assistance service	2.187	Average important
16	Attitude	2.120	
17	Impression	2.054	
18	Packaging ability	2.009	
19	Ended-works reports	2.003	
20	Geographic position	1.872	
21	Total of ended business	1.597	
22	Training aids	1.537	
23	Reciprocal agreements	0.610	Slight important

Despite the fact that the list of criteria was made quite a long time ago in 1966, lots of those criteria are still relevant. Quality, price, delivery conditions, financial reliability, reputation, and so on are the first criteria that come to mind when talking about SS. Lyes Benyoucef, Hongwei Ding, and Xiaolan Xie [13] gathered together Dikson's criteria, added some new ones, and made a list of criteria and sub-criteria. The main preference was given to the service, which included delivery parameters and suppliers' flexibility. According to Roya Ghamari [2], the most important criteria in the steel industry are price, service, delivery, flexibility, technology, and quality. The article of Agnieszka Konys [30, p. 14] provides a diagram with the criteria that are most frequently discussed in articles about the SS problem. Quality, costs, service, and risks are among them. There are three categories of criteria described in the article [3, p. 1420]: economic, social, and environmental. Price, quality, technological capability, and supplier reliability are the main economic criteria there. So it seems that even now Dickson's criteria are not so bad.

Lots of information and understanding about customers' approach towards supplier quality requirements were taken from the integrated energy company Chevron Australia's Supplier Quality Requirements [40]. Finance reliability, innovations, quality certificates, and guarantee were taken into account.

In the article about SS in high-tech industry [4, p. 7922] the author provides the table, where quality, price, service, and other well-known criteria are divided by the smaller sub-criteria. It is suggested to find a score for each criterion based on the sub-criteria by implementing one of the SS methods. Such an approach fits the SAP Vendor Evaluation concept [37], which will be explained in more detail in the following chapters.

Gathering the information from those materials, three of the most common criteria were taken to be considered in this work: price, quality, and service (**Table 4**, "Criteria" column). All related sub-criteria are represented in the column “Sub-criteria” of the **Table 4**.

One of the purposes of this work is to understand, what criteria may be taken into account to evaluate suppliers’ “green components”. Looking at Dikson’s criteria list it seems, that in the 1960-s the impact on the environment didn’t worry the business community. Seems, that the environmental impact and green economy were not the priority in the 60th of the previous century. Based on the information gathered from the reference [14, p. 4] some green parameters were considered. Unfortunately, most of them are non-numeric and it’s hard to interpret them in the model. But the vector of exploration was taken from this.

Table 4: The set of criteria and sub-criteria, used in the model for the evaluation

Criteria	Sub-criteria	Data	Explanation	Unit of measure	Business object	Field type
Price	Price level	Price level	Determine how good the particular price is among the prices of other suppliers.	currency	Quotation Info-record Purchasing order	Standard
Price	Price changes	Price changes	How much the price for the product was changed within the period (month, year).	%	Info-record Quotation	Standard
Service	Delivery evaluation	Lead time		days	Quotation	Standard
Service	Delivery evaluation	Amount (%) that could be delivered on time	Usually, vendors deliver the ordered amount in batches. This criterion means how much the supplier may deliver in the first batch.	%	Quotation	Standard
Service	Batch flexibility	Max possible Batch size		integer or float	Quotation	Enhanced
Service	Batch flexibility	Min possible Batch size		integer or float	Quotation	Enhanced
Service	The capability of handling abnormal quality	The maximum extra quantity that can be delivered within the specified time frame	Sometimes some unexpected occasions may occur and business needs an extra amount of material. Here is the additional amount that the supplier may deliver according to its production capacity.	integer or float	Quotation	Enhanced
Quality	Return ratio	The ratio of returned quantity.	If the company did some business with the supplier in frames of other contracts, it's possible to collect the data about deliveries and evaluate the percentage of a returned quantity.	returned items / delivered items	Data Base information about Purchase Orders	Standard
Quality	Basic quality certificates	Quality-related certificates		True / False Or the set may be ranked	Business Partner	Enhanced
Quality	Innovations	A number of patents or new product introductions, e.g. 3 patents/year.	The different time horizons are taken (1 and 10 years).	items	Business Partner	Enhanced
Quality	Guarantee	Guarantee period		days, months, or years	Quotation	Enhanced
Quality	Guarantee	Response time to technical inquiries or problems		hours or days	Quotation	Enhanced
Quality	Finance	Current ratio		%	Business Partner	Enhanced
Quality	Finance	Credit rating		A-, AAA, BB	Business Partner	Enhanced

Quality	Finance	Length of time allowed for payment		Days	Business Partner	Enhanced
Green	Pollution control	Carbon footprint	Simplification: we will consider emissions per item. Basic units of measure are the same: kgCO2e [41].	kgCO2e / item	Quotation	Enhanced
Green	Pollution control	Waste water	Simplification: we will consider volume per item. The basic unit of measure is liter/item.	liter/item	Quotation	Enhanced
Green	Pollution control	Solid non-recyclable wastes	Simplification: we will consider volume per item. The basic unit of measure is KG/item.	KG/item	Quotation	Enhanced
Green	Pollution control	Energy consumption	Simplification: we will consider volume per item. The basic unit of measure is kWh / item.	kWh / item	Quotation	Enhanced
Green	Pollution control	Use of harmful/hazardous material	Simplification: we will consider volume per item. The basic unit of measure is KG/item.	KG per Item	Quotation	Enhanced
Green	Green competencies	Certification (as ISO 14000)	Customers' managers may find information about ISO certificates on suppliers' websites or request information from the supplier.	True / False Or a different set. may be ranked	Business Partner	Enhanced
Green	Environment management (product level)	Green packaging	Just check: whether provided or not.	True / False	Quotation	Enhanced
Green	Environment management (product level)	Cost of component disposal		price per item	Quotation	Enhanced
Green	Environmental management (general)	Ability to alter processes and products for reducing the impact on natural resources	We (as a company) can't control this criterion. But could be counted as True/False from the information, provided by the vendor.	True / False	Business Partner	Enhanced
Green	Environmental management (general)	The ratio of green customers to total customers		%	Business Partner	Enhanced
Green	Environmental management (general)	Environmental laws and regulations complaint		How many environmental litigations were made (for the last year?)	Business Partner	Enhanced
Green	Environmental management (general)	Support from top managers in their commitment to the environment.		% of profit, that is spent on Environment	Business Partner	Enhanced
Green	Environmental management (general)	Demonstrates corporate social responsibility.	The info could be checked by reports and social media.	True / False	Business Partner	Enhanced

Green	Environmental management (general)	Green process planning	We (as a company) can't control this criterion. But could be counted as True/False from the information, provided by the vendor. The info could be found on the vendor's website (in the annual report, separate environmental docs, and so on).	True / False	Business Partner	Enhanced

Green parameters become more and more important. Some types of “green” criteria are described in the references [2] and [8]. It’s really valuable for this work, that those articles provide units of measure for explained green parameters because the model is going to be used in IS. The parameters are as follows:

- Energy (Watt) use of units of products supplied
- Waste (Kg) generated from the production of a unit of products supplied
- Carbon emission (Kg) for units of product supplied
- Supplier rating value for green supplies
- etc.

Those ones are suitable to be represented numerically. Unfortunately, not all of them could be available in companies’ ISs. Some businesses do not enter the data into the system even though they know the parameters to save resources and time, others have no necessary data about these parameters or see no value for their business in it. Some companies may collect energy usage and generated wastes, other carbon emissions. Creating the model for this thesis it assumed that those parameters are available from the suppliers. The logic of the model will also consider that the suppliers’ data may not be complete.

As was mentioned, the Ministry of Finance and Statistics in Norway [7] suggests increasing tax on CO2 emissions. So using the “carbon footprint” coefficient may become a relevant criterion in the SS problem. It allows suiting state requirements in increasing “green” component impact in business. This parameter is good for use in ISs as well because it is numeric, it could be calculated and estimated.

The explanation of how “carbon footprint” may influence SS is given by Amit Kumar and Vipul Jain [32]. The authors use the DEA method for SS and describe steps that should be done to collect “carbon footprint” information precisely. The method, represented in this work, is supposed to use by customers. However, it seems, that customers don’t require to know how to calculate “carbon footprint” correctly, because customers just wait for the number for CO_2 for the product from the supplier. But the thing that is more interesting is the standards and related certificates that suppliers may have to be sure that the “carbon footprint” is calculated properly.

It’s possible to collect data about suppliers’ ISO certificates. Usually having such a certificate is a glory for the supplier, and it wouldn’t be difficult for customers’ managers to find this information about the supplier (or request the information from the supplier). Such information may be carried out in the system as a Boolean parameter [22]. The way of collecting and storing that information is represented in the **Attachment 5** (Criterion code 95 “ISO Certificates”).

To have a more complete environmental view of suppliers, the sub-criteria “environmental management” was suggested in the article [4]. It’s a bit more difficult task to find parameters that allow us to evaluate this because they must be represented in a numeric way (more or less). Amy H.I. Lee, He-Yau Kang, Chang-Fu Hsu, and Hsiao-Chu Hung [4] suggest taking into account continuous monitoring, regulatory compliance, and green process planning. Being a customer it’s quite difficult, sometimes impossible to gather and check this information. Of course, the data may be claimed from the supplier or bought from different rating agencies, but in the first case, the customer can’t check it, in the second not all vendors may be in the agencies’ database.

Such criteria as green packaging and cost of component disposal suggested in [4] may be represented as true/false and numerical parameters respectively. Moreover, such parameters should be filled at the product level. It means, that environmental management may be evaluated both: on supplier and product levels. So, those two parameters are taken to evaluate environment management on the product level.

The set of standards for a company’s behavior ESG (Environmental, Social, and Governance) suggests some environmental criteria [38]. The ability to alter processes and products for reducing the impact on natural resources and green process planning are among them.

Green process planning, the ability to alter processes and products for reducing the impact on natural resources, and demonstration of the corporate social responsibility, suggested in [4], are considered environmental management on the supplier level. Customers can't control this vendor's data precisely, but the info could be found on the vendor's website (in the annual report, separate environmental docs, and so on). Such parameters may be represented as true/false in the IS.

Some numeric parameters could be taken into account to evaluate supplier environmental management on a general level. Salman Bashir Memon, Amran Rasli, Abdul Samad Dahri, and Imelda Hermilinda Abas [42] describe the importance of the top-managers involvement in environmental approaches support. This involvement may be measured as the percentage of profit, that is spent on the environment. Usually, such data is listed in companies' annual reports and news on social media or websites.

As was discussed previously, the government is going to track companies' environmental low compliance more precisely, especially in terms of CO_2 emissions. So seems logical to track the number of environmental litigations made by an evaluated vendor. Suppliers' behavior tends to change: ones, who broke the law a long time ago may track their processes more precisely now. So there is no need to record all environmental law violations for the whole company's history. The suggestion is to take into account only recent, for instance for the last year.

It's also possible to take into account the partners with whom the evaluated supplier works. If many partners comply with international standards, like ISO, then indirectly it can be concluded that the evaluated supplier knows how to work with environmental laws.

Gathering presented information **Table 4** was created. The column "Criteria" collect 4 main criteria, that are used in current work for supplier evaluation. There are sub-criteria that make up the given criteria in the respective column "Sub-criteria". Each criterion consists of two to five sub-criteria. Then to assign the score for the sub-criteria various sets of data are used, which is represented in the column "Data". The column "Explanation" has comments for the data that may be unclear. As the criteria are supposed to use in the IS, it's important to define units of measure in advance. The units of measure are represented in the corresponding column. The "Business object" column has the system's objects, where the data is stored or is going to be stored. The "Field type" column explains whether the necessary object field exists ("Standard" mark) or is expected to be enhanced ("Enhanced" mark).

Let's consider the following example to understand how to read the table. The criterion "Green" consists of four sub-criteria: Pollution control, Green competencies, Environment management (product level), and Environmental Management (general). The sub-criterion "Pollution control" consists of some product-specific parameters, such as waste water, solid non-recyclable wastes, energy consumption, and use of harmful/hazardous material. The data parameter "Carbon footprint" is measured by kgCO2e / item and is stored in the quotation's enhanced field.

4.3 Information system for supplier selection

To create an appropriate model, it needs to clarify which SS criteria are most important for the company. Criteria for the current model were defined in the previous chapter, but in real business activity, they may be changed. It should be taken into account during mode architecture development. The next step is to collect supplier data from the company's ISs. Companies may have more than one IS like CRM to work with clients, ERP to run the enterprise, WMS to automatize warehouse processes, and so on. The data required for the evaluation may be stored in different ones. The only SAP ERP system is considered in this work (see **Chapter 2.2)**. Then it is necessary to arrange the obtained data in a model in accordance with the approach, study and evaluate the results obtained, and return the corresponding score.

Nowadays the leaders of the ISs market suggest ready-made solutions for connecting suppliers and customers. For instance, SAP has the product Ariba Sourcing. This is a software-as-a-service (Saas) solution for strategic sourcing to negotiate and implement value-added agreements. It is a cloud solution that allows suppliers and buyers to connect and do business on a single platform. There is no need to buy servers or rent databases. The information about suppliers and customers is stored at the side of SAP. The Ariba Sourcing solution enables companies to automate and expedite sourcing processes, simplifying the control and management of suppliers and agreements [16].

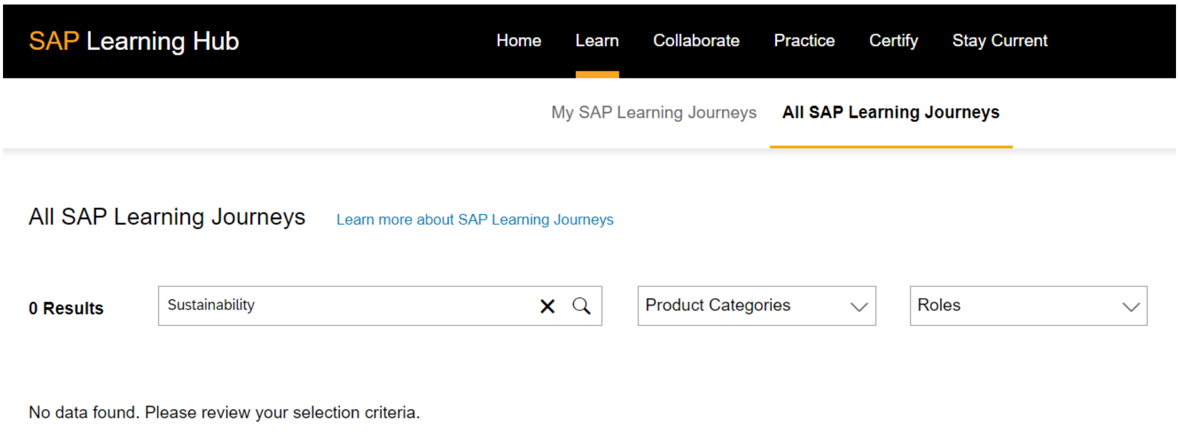
Unfortunately, some blocking factors prevent implementing the current SS model in the Ariba platform (at least from the platform users' side). First of all, this platform doesn’t suggest any instruments for implementing customer logic. Being the cloud platform it provides preset tools, that couldn’t be customized according to all users’ needs. The range of SS criteria is predefined and users may operate only by given ones. In practice, it helps to find a suitable supplier or collect a set of suppliers, but not to choose the most appropriate one. This is one more reason to explore the topic of SS on the basis of companies’ ISs.

Second of all, the platform doesn’t collect the whole information that may be required for supplier evaluation. Even if there were opportunities, then the data in Ariba itself would not be enough to make a decision. It would take to create an implementation of integration requests to other systems because Ariba doesn’t collect transactional data, historical data, and so on. For instance, it’s impossible to evaluate the ratio of returned quantity without integration.

Third of all, the question of the data safety of Cloud platforms is still significant. Even if the platform would provide an opportunity to store transactional and historical data, not all companies would agree to keep this data on Cloud. The question of Cloud Platforms' safety is out of this work, but it's good to keep it in mind.

So, seems, that Ariba may be used as a great tool to collect the range of suppliers and some information about them. The system may be connected with the main company's IS as well. But now it's quite unlikely to use Ariba as the main IS for implementing custom logic for SS.

SAP provides some software solutions that help companies to manage their environmental and social impacts. One of them is SAP Sustainability Performance Management. It helps to improve their sustainability performance as well. These solutions cover various areas, such as environmental compliance, carbon accounting, sustainable supply chain management, product compliance, and sustainability reporting. The solution enables tracking and reporting companies' sustainability performance, identifying opportunities to reduce their environmental impact and costs, and improving their brand reputation and stakeholder engagement [43]. Unfortunately, there is a lack of detailed information about such functionality. Even SAP's Learning Hub doesn't have any documentation (**Figure 7**). It seems that those solutions are not so popular now, they are in the developing stage. It needs some time to collect the data and evaluate the value of those solutions. But this solution seems really relevant for this work.



All SAP Learning Journeys

Learn more about SAP Learning Journeys

0 Results

Sustainability

Product Categories

Roles

No data found. Please review your selection criteria.

Figure 7: SAP Learning Hub, Sustainability search

SAP suggests Environmental Compliance software that helps organizations manage their environmental compliance obligations and reduce their environmental risk. The solution covers a range of environmental compliance requirements, including air emissions, water management, waste management, and hazardous materials handling [44]. This solution has the same problem: it's not widely known, that's why there is not enough information available.

Unlike Ariba, SAP On-premise is flexible enough to make custom implementations. It has lots of available documentation, guidance, and forums discussions, that make it easy to use compared to SAP Environmental Compliance and SAP Sustainability Performance Management. Some special enhancement points in SAP On-premise are provided by SAP. It makes the process of custom functionality implementation much more clear. The approach of custom implementation in SAP is explained in more detail in **Chapter 5.1**.

Using SAP ERP it's necessary, that all required data may be stored in the system. That's why system business objects[1] must be defined for the corresponding parameters. In implementing the current model, the following business objects are required:

- **Business partner (BP).**

This object collects data about the parties with whom the business has an interest. The object is divided into three types: persons, organizations, and groups. Each BP has its type and may be assigned to multiple roles. For instance, the same BP may both supply some materials for production and buy the company's products. Such a supplier would have two roles in the system: vendor and customer. A separate role is used to store payment details. This architecture allows to store the whole information about the same entity in one place. This object is supposed to be enhanced according to SAP users' requests without ABAP implementation [45].

- **Quotation/supplier response.**

This object is used to organize a purchase process in SAP. Usually, the object is used during the tender. The requests for the quotation are sent to suppliers. Basically, it includes the required product, supposed delivery dates, product quantity, the deadline for the application, batch information, and other required information. Responses are gathered into quotations filled by the data from suppliers. Despite that the process is divided into two steps, the system uses the same business object and the same database table. To not confuse users there are separate transactions made [46].

[1] SAP Business Object is a centralized suite for data reporting, visualization, and sharing. It helps to transform data into useful insights, available anytime, anywhere [66]. Usually Business object architecture supposed separate data base tables, transactions and reports to supply an effective work with the corresponding Business Object.

- **Info record**

The purpose of the SAP info record is to provide users with relevant information during the purchasing process, such as details on a relationship between specific material and its vendor, and is commonly used as a source of information for other purchasing documents (purchasing requests and orders). It includes various types of information, including the current pricing for the material, delivery time, and so on. It enables buyers to easily identify the materials that have been previously offered or supplied by a vendor and the vendors that have offered or supplied a specific material [46]. This business object is not enhanced during the current model implementation, its data is used by SAP standard algorithms.

- **Purchase order (PO)**

PO is a formal instruction or request made by a purchasing organization to a vendor or plant, specifying the required quantity of goods or services to be provided at a designated time. It comprises a document header and multiple items. The header contains information that relates to the entire PO, including details such as the terms of payment and delivery conditions. PO items are specified the necessary materials or components that should be provided to the subcontractor for assembly or processing according to each delivery date specified. PO items may reflect services with corresponding service specifications [46]. This business object is not enhanced during the current model implementation, its data is used by SAP standard algorithms.

The mapping between chosen parameters and business objects is represented in the column “Business object” of **Table 4**.

5 Development of supplier selection models

5.1 Standard SAP supplier evaluation functional

The basic supplier evaluation process in SAP is executed in the following order [37]:

1. Configure necessary criteria and sub-criteria.
2. Configure the relationship between organizational levels and appropriate criteria and sub-criteria. Purchasing Organization is the grouping organizational level for that process. Usually Purchasing Organization in SAP MM [46] is an individual or a group of people (department) responsible for the purchase of certain materials and services. Different departments may have their specific criteria for the supplier evaluation so that this is the groping level.
3. Insert necessary master data (business partners, materials, info-records, and so on) and transactional data (quotations, purchase orders, material documents).
4. Run the supplier evaluation transaction (ME61 - Maintain Supplier Evaluation).
5. Run the report with the suppliers' ranks to choose the most appropriate one (ME6B - Display Supplier Evaln. for Material).

5.2 Business process description

To have a full understanding of the purpose of the algorithm the process should be explained from the user's point of view. The algorithm logic is described in **Chapters 5.3, 5.4,** and **5.5**.

As was described previously, the SAP system was chosen as the basis for the study. Basically, ISs serve the same purpose, but definitely, all of them have their specific differences. There are several resources were used to gain information about the SAP-specific features, configurations, instructions, and so on.

The electronic resource www.sap-press.com [47] gathers lots of literature from SAP architects, consultants, and developers. There is literature about different SAP modules on that resource. The book "ABAP: an introduction" [48] was explored to learn its syntaxes and features.

The specified SAP blogs blogs.sap.com [49] and www.tutorialspoint.com [50] were used to find explanations about SAP processes from the other users. Here are lots of articles with screenshots, code examples, and advice for the approaches. Web forums answers.sap.com [51] and abap-blog.ru [52] were attended to find solutions for the errors and issues that appeared during model development. The basic ABAP documentation is represented in the reference [20].

The website microlearning.opensap.com [53] was visited to find information about configurations and business approaches. This website has useful video materials. The references [16] and [21] are used to explore information about SAP solutions: Cloud, like Ariba, and on-premise, like SAP ERP. The references [54], [55], [56], [57], [58], [43] lead to the SAP documentation that is used to explain some terminology.

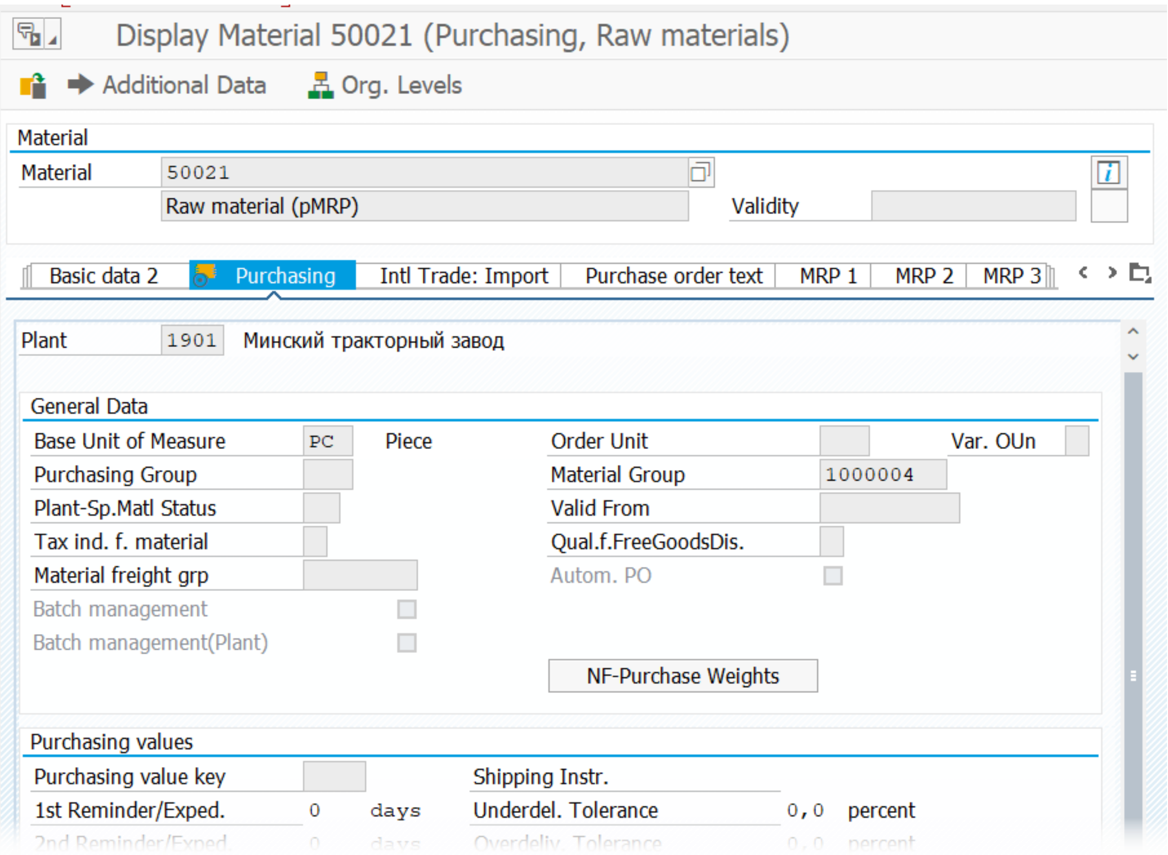
The SS business process takes days, sometimes months, and lost employees from different departments are involved. Moreover, the details of the SS process may be different from company to company, so here the main steps of SS are discovered. The process is described from the customer's point of view. It is supposed, that all required configurations are made by the system administrator. The following steps may be replicated in any ERP system; the given example is based on SAP ERP functionality.

5.2.1 Create master data

The master data (MD) specialist inputs the vendors' master data in the "BP" transaction. In addition to the standard data, like its address and payment details, the worker enters additional data related to the Business Partner object as described in the **Table 4** (column "Business object"). The fields shown in the **Attachment 5** should be filled in as well. Then the MD worker must check, whether the material master data is in the system. MM01, MM02, and MM03 transactions are used (**Figure 8**). In case the necessary data doesn't exist, the worker inputs the corresponding data.

Then the MD specialist must configure the list of suppliers and corresponding material for the evaluation. In the maintenance view, described in **Attachment 6** (**Figure 36**, **Figure 39**, **Figure 40**). The worker chooses an appropriate purchasing organization, then enters a material with the plant, and assigns the flag "Material for the evaluation" against the desired material. Then fulfill the list of suppliers that are going to be evaluated.

For this master thesis, only customer views are generated. Those views may be changed for the transactional maintenance in the productive systems. After the necessary MD is in the system, the MD specialist notifies the purchasing manager, that all data are available.



Display Material 50021 (Purchasing, Raw materials)

Additional Data　Org. Levels

Material

Material	50021		
	Raw material (pMRP)	Validity	

Basic data 2 | Purchasing | Intl Trade: Import | Purchase order text | MRP 1 | MRP 2 | MRP 3

Plant 1901 Минский тракторный завод

General Data

Base Unit of Measure	PC	Piece	Order Unit		Var. OUn	
Purchasing Group			Material Group	1000004		
Plant-Sp.Matl Status			Valid From			
Tax ind. f. material			Qual.f.FreeGoodsDis.			
Material freight grp			Autom. PO	□		
Batch management	□					
Batch management(Plant)	□					

NF-Purchase Weights

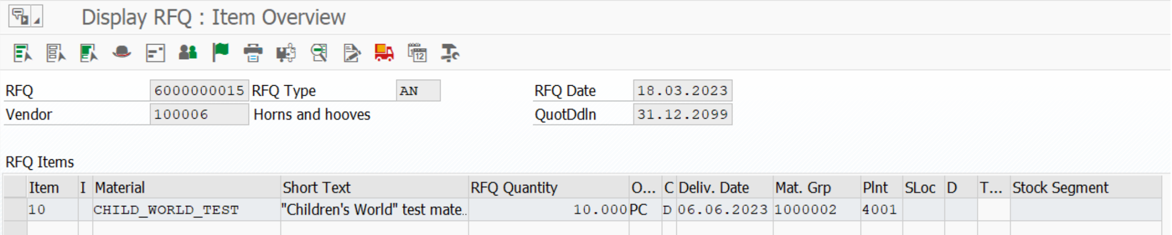
Purchasing values

Purchasing value key			Shipping Instr.		
1st Reminder/Exped.	0	days	Underdel. Tolerance	0,0	percent
2nd Reminder/Exped.	0	days	Overdeliv. Tolerance	0,0	percent

Figure 8: An example of material MD view.

5.2.2 Create and send requests for the quotation

Purchasing manager generates requests for the quotation in the transaction ME47 “Create Quotation”. The purchasing material, required quantity, delivery date, response deadline, purchasing organization, plant, and all other relevant information is listed in the request (**Figure 9**, **Figure 10**). Those requests then are sent to the suppliers by mail, post mail, EDI[1], fax, or any other available method.



Display RFQ : Item Overview

RFQ	6000000015	RFQ Type	AN	RFQ Date	18.03.2023
Vendor	100006	Horns and hooves		QuotDdln	31.12.2099

RFQ Items

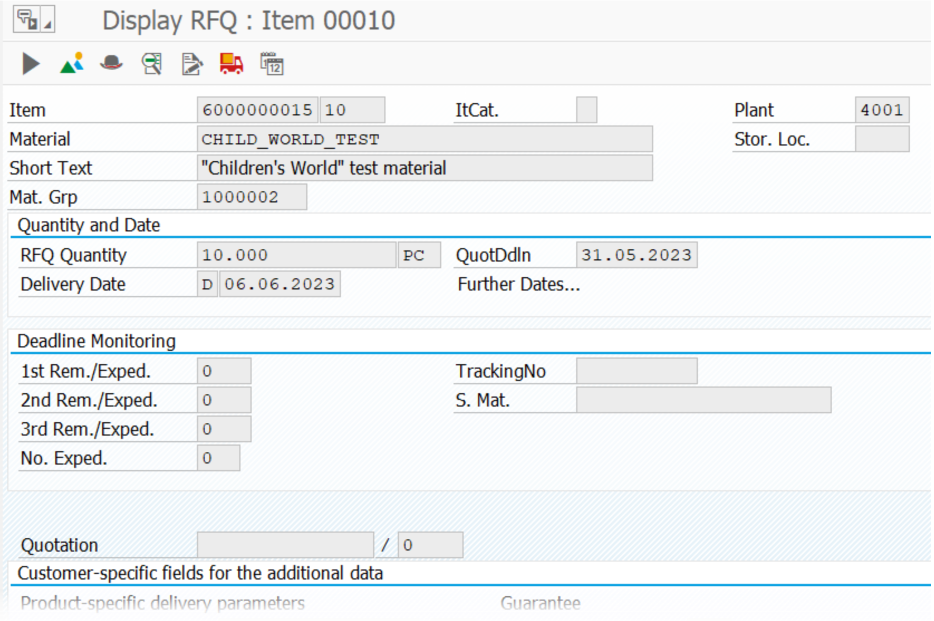
Item	I	Material	Short Text	RFQ Quantity	O...	C	Deliv. Date	Mat. Grp	Plnt	SLoc	D	T...	Stock Segment
10		CHILD_WORLD_TEST	"Children's World" test mate..	10.000	PC	D	06.06.2023	1000002	4001				

Figure 9: Request for the quotation, general view

[1] The Electronic Data Interchange (EDI) component in SAP consists of an Intermediate Document (IDoc) interface. The interface may be used to do the following:

1) Send messages (outbound processing) such as an order confirmation through Electronic Data Interchange (EDI).

2) Receive messages (inbound processing) such as a sales order through EDI [58].



Display RFQ : Item 00010

Item	6000000015 10	ItCat.		Plant	4001
Material	CHILD_WORLD_TEST			Stor. Loc.	
Short Text	"Children's World" test material				
Mat. Grp	1000002				

Quantity and Date

RFQ Quantity	10.000 PC	QuotDdln	31.05.2023
Delivery Date	D 06.06.2023	Further Dates...	

Deadline Monitoring

1st Rem./Exped.	0	TrackingNo	
2nd Rem./Exped.	0	S. Mat.	
3rd Rem./Exped.	0		
No. Exped.	0		

Quotation / 0

Customer-specific fields for the additional data

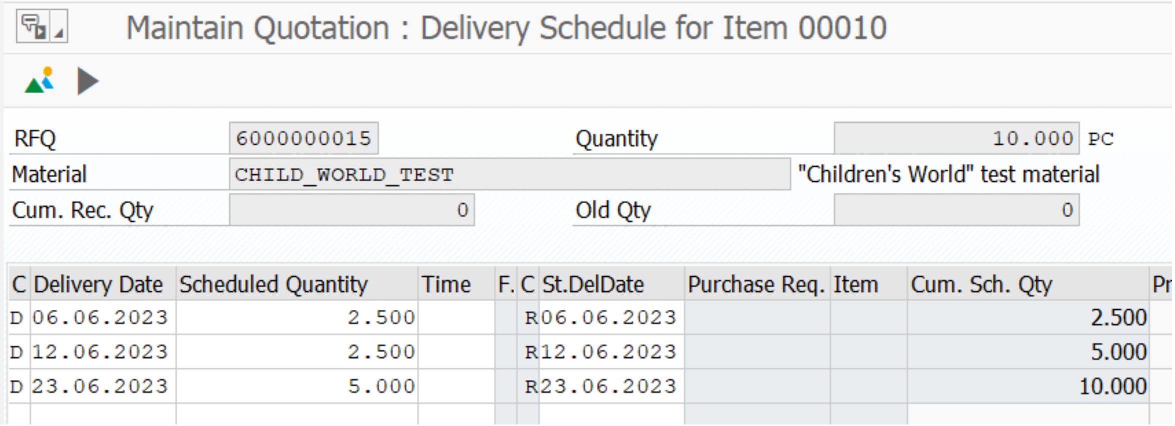
Product-specific delivery parameters Guarantee

Figure 10: Request for the quotation, item view

5.2.3 Enter suppliers' responses into the quotations

Before the deadline for submission of proposals, the data is collected in the system. It could be done automatically by configuring connection customer's and supplier's systems, it could be done manually by the master data specialists. The transaction ME47 is used for the manual entry. The data could be entered by suppliers directly into the customer's system. The way depends on configured infrastructure and relationship.

In the current example, suppliers' responses have the necessary additional data (**Figure 35**). Quotation may have the batch numbers and their delivery dates (**Figure 11**), that are taken into account for the "Quality" criteria calculations.



Maintain Quotation : Delivery Schedule for Item 00010

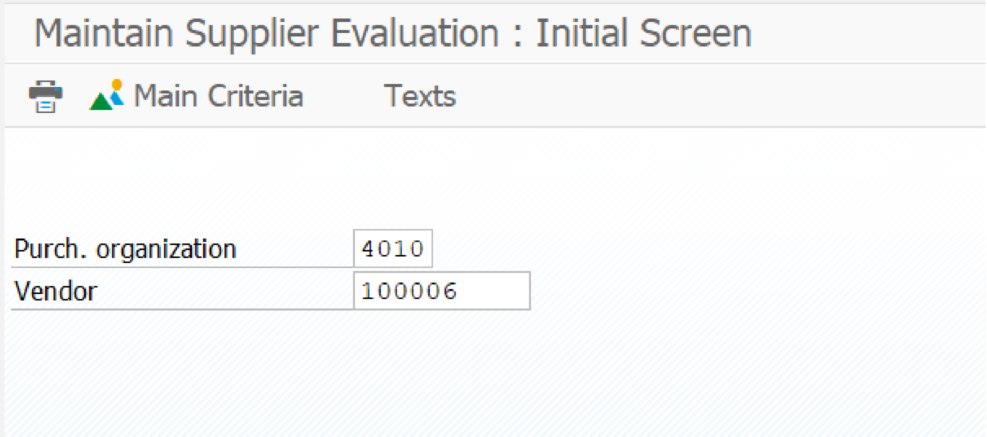
RFQ	6000000015	Quantity	10.000	PC
Material	CHILD_WORLD_TEST	"Children's World" test material		
Cum. Rec. Qty	0	Old Qty	0	

C	Delivery Date	Scheduled Quantity	Time	F.	C	St.DelDate	Purchase Req.	Item	Cum. Sch. Qty	Pr
D	06.06.2023	2.500			R	06.06.2023			2.500	
D	12.06.2023	2.500			R	12.06.2023			5.000	
D	23.06.2023	5.000			R	23.06.2023			10.000	

Figure 11: Quotation batches example

5.2.4 Perform suppliers' evaluation

After the quotation deadline has expired, the purchasing manager starts to perform supplier evaluation. The evaluation can be performed in any of two ways: evaluate each supplier separately or run the evaluation in the background. Let's consider the first one in more detail. Transaction ME61 is used for this purpose. The initial screen requires Purchasing Organization and Supplier number to be entered (**Figure 12**). The evaluation criteria may be different from purchasing department to purchasing department, that's why Purchasing Organization is required.



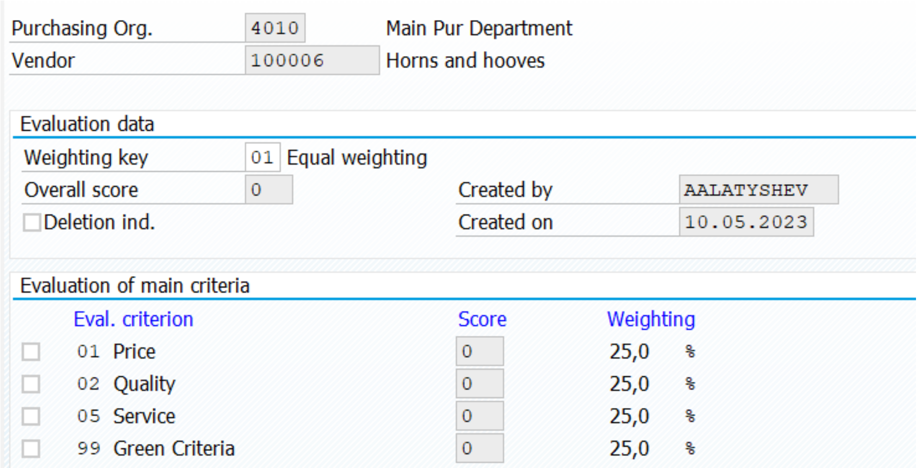
Maintain Supplier Evaluation : Initial Screen

Purch. organization 4010
Vendor 100006

Figure 12: Maintain supplier evaluation, the initial screen

Figure 13 shows the criteria, that are configured in **Chapter 5.3.3** (**Figure 23**). If the supplier is evaluated for the first time. then the scores fields are empty (**Figure 13**). The weighting key, configured in the Chapter **5.3.3** (**Figure 31**) is in the field "Weighting key". It could be changed by the user if required. Double-clicking on any main criteria the corresponding sub-criteria, which are configured in **Chapter 5.3.3** (**Figure 30**)**,** are shown in the **Figure 14**. To run the calculating algorithm, the button "*Auto. New Eval./MCrit*" should be pushed (**Figure 14**, red frame). Button pushing is the starting action for the model, described in **Figure 15** and **Figure 16**. More detailed those figures are explained in **Chapter 5.4**. The values are displayed in the sub-criteria fields (**Figure 17**). Going back (F3 button) the overall score and other calculated main criteria are displayed (**Figure 18**). If required, the user may change the weighting key to "Unequal" to see the output changes. The percentages close to the main criteria values are changed, and the overall score is changed as well (**Figure 19**).

This process then is repeated for the other suppliers under evaluation. If required, the process may be launched for all suppliers in the program RM06LBAT in the background.



Purchasing Org.	4010	Main Pur Department
Vendor	100006	Horns and hooves

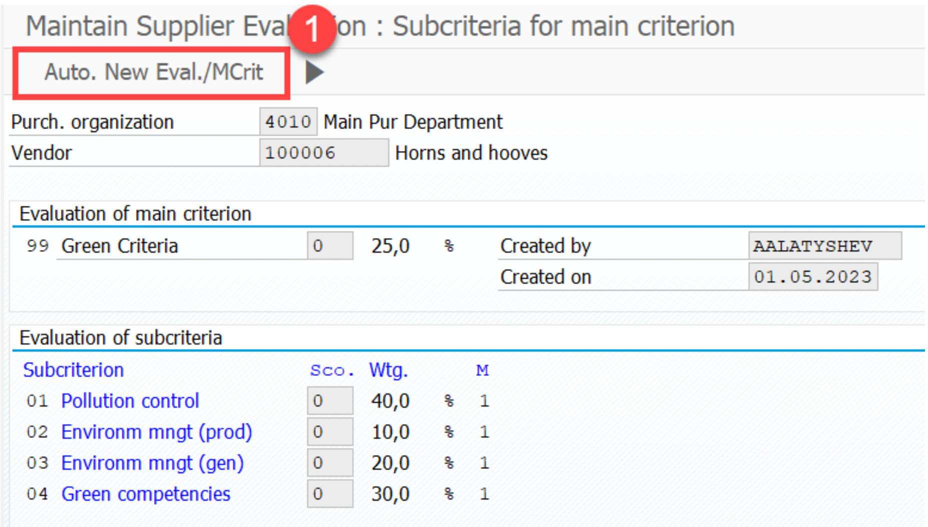
Evaluation data

Weighting key	01	Equal weighting		
Overall score	0		Created by	AALATYSHEV
□ Deletion ind.			Created on	10.05.2023

Evaluation of main criteria

		Eval. criterion	Score	Weighting	
□	01	Price	0	25,0	%
□	02	Quality	0	25,0	%
□	05	Service	0	25,0	%
□	99	Green Criteria	0	25,0	%

Figure 13: Maintain supplier evaluation, main criteria, non-evaluated



Maintain Supplier Eval 1 ion : Subcriteria for main criterion

Purch. organization	4010	Main Pur Department
Vendor	100006	Horns and hooves

Evaluation of main criterion

99	Green Criteria	0	25,0	%	Created by	AALATYSHEV
					Created on	01.05.2023

Evaluation of subcriteria

	Subcriterion	Sco.	Wtg.		M
01	Pollution control	0	40,0	%	1
02	Environm mngt (prod)	0	10,0	%	1
03	Environm mngt (gen)	0	20,0	%	1
04	Green competencies	0	30,0	%	1

Figure 14: Maintain supplier evaluation, non-evaluated sub-criteria. 1 - run the calculation.

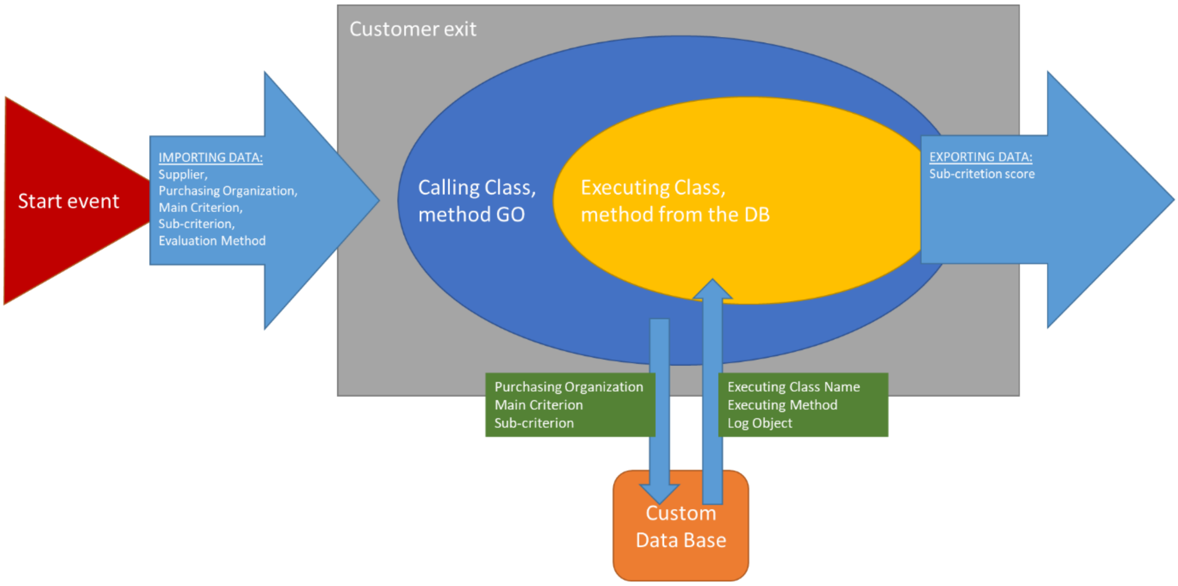


Figure 15: Dynamic customer-exit (EXIT_SAPLMEL0_001) architecture. Own development.

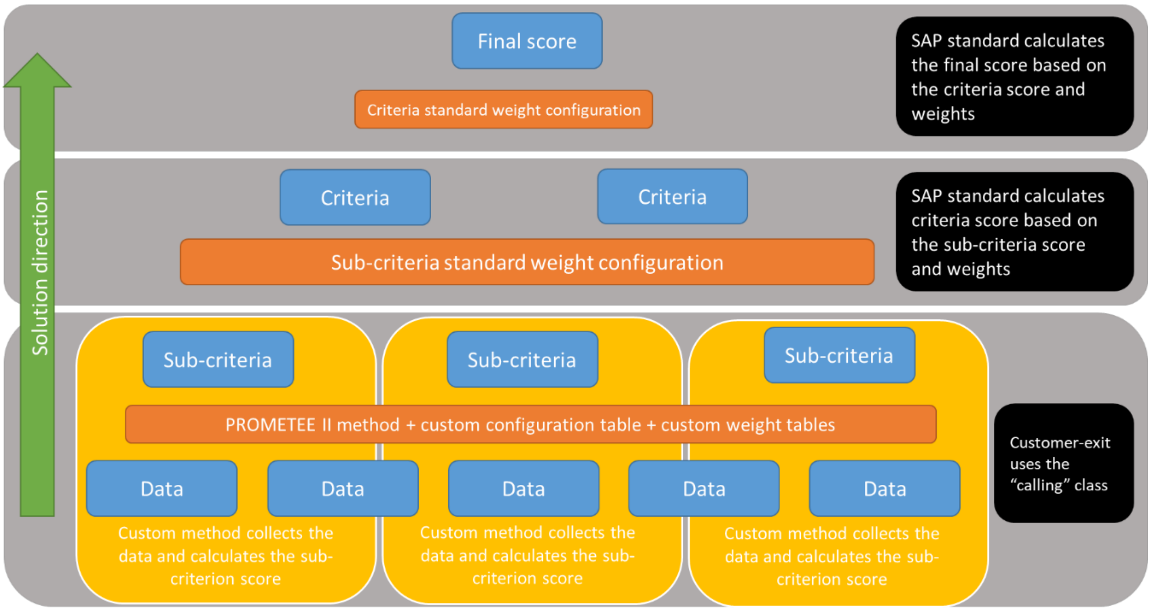
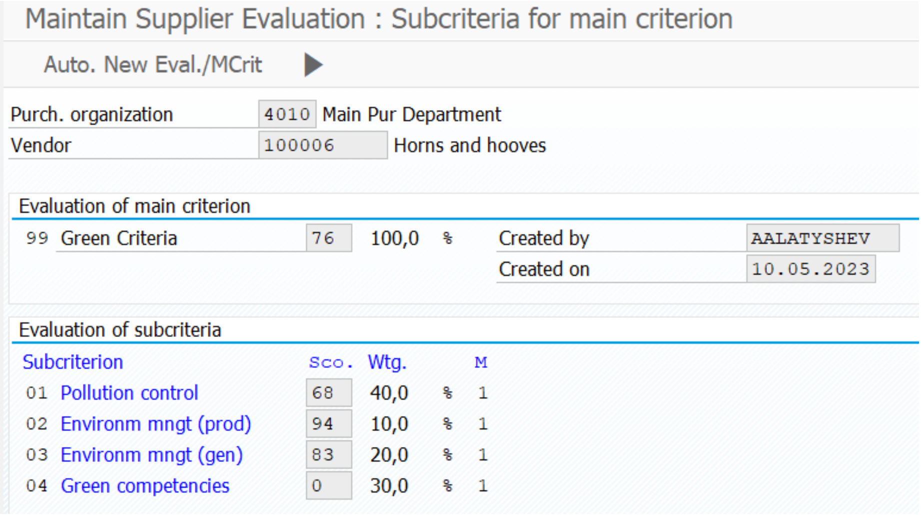

Final score
Criteria standard weight configuration
SAP standard calculates the final score based on the criteria score and weights
Criteria
Criteria
Sub-criteria standard weight configuration
SAP standard calculates criteria score based on the sub-criteria score and weights
Solution direction
Sub-criteria
Sub-criteria
Sub-criteria
PROMETEE II method + custom configuration table + custom weight tables
Data
Data
Data
Data
Data
Custom method collects the data and calculates the sub-criterion score
Custom method collects the data and calculates the sub-criterion score
Custom method collects the data and calculates the sub-criterion score
Customer-exit uses the "calling" class

Figure 16: The project architecture. Own development.



Maintain Supplier Evaluation : Subcriteria for main criterion

Auto. New Eval./MCrit ▶

Purch. organization	4010	Main Pur Department
Vendor	100006	Horns and hooves

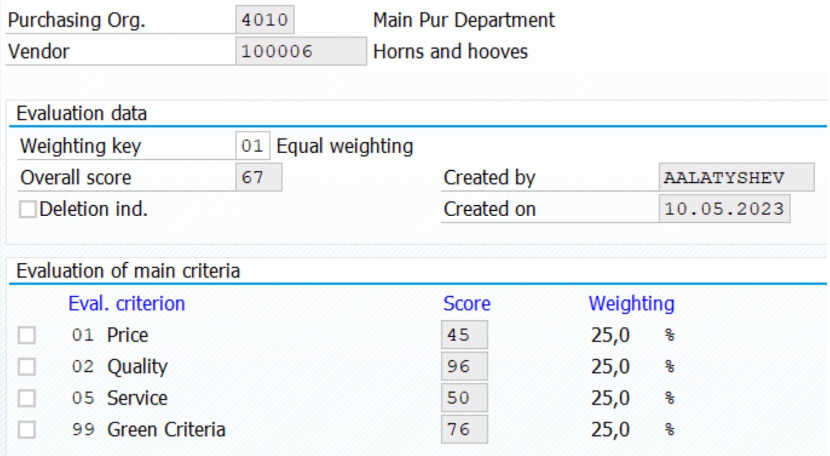
Evaluation of main criterion

99	Green Criteria	76	100,0	%	Created by	AALATYSHEV
					Created on	10.05.2023

Evaluation of subcriteria

Subcriterion		Sco.	Wtg.		M
01	Pollution control	68	40,0	%	1
02	Environm mngt (prod)	94	10,0	%	1
03	Environm mngt (gen)	83	20,0	%	1
04	Green competencies	0	30,0	%	1

Figure 17: Maintain supplier evaluation, evaluated sub-criteria



Purchasing Org.	4010	Main Pur Department
Vendor	100006	Horns and hooves

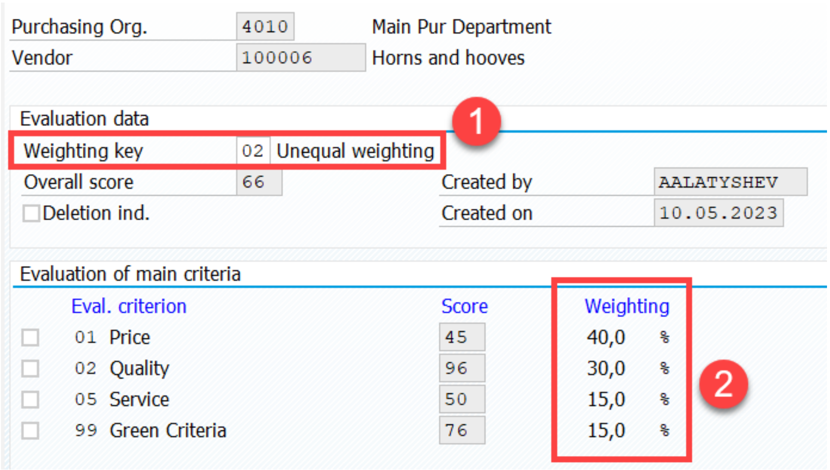
Evaluation data

Weighting key	01	Equal weighting		
Overall score	67		Created by	AALATYSHEV
□ Deletion ind.			Created on	10.05.2023

Evaluation of main criteria

	Eval. criterion		Score	Weighting	
□	01	Price	45	25,0	%
□	02	Quality	96	25,0	%
□	05	Service	50	25,0	%
□	99	Green Criteria	76	25,0	%

Figure 18: Maintain supplier evaluation, main criteria, evaluated



Purchasing Org.	4010	Main Pur Department
Vendor	100006	Horns and hooves

Evaluation data

Weighting key	02 Unequal weighting		
Overall score	66	Created by	AALATYSHEV
□ Deletion ind.		Created on	10.05.2023

Evaluation of main criteria

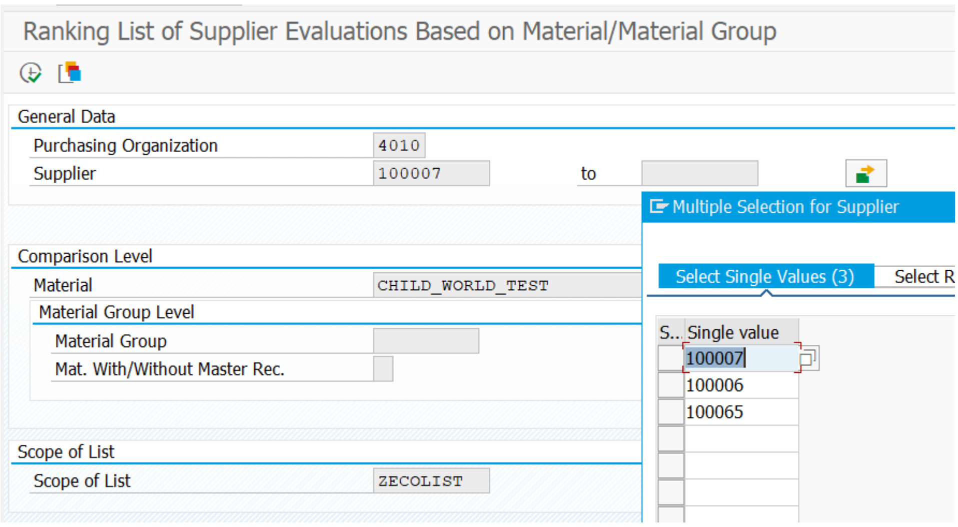
	Eval. criterion	Score	Weighting
□	01 Price	45	40,0 %
□	02 Quality	96	30,0 %
□	05 Service	50	15,0 %
□	99 Green Criteria	76	15,0 %

Figure 19: Maintain supplier evaluation, main criteria, 1 - unequal weighting key, 2 – weight values

5.2.5 Choose the best supplier

SAP provides the report in the transaction ME6B “Evaluations per Material/Material Group”, that allows to display of suppliers, their final score, and rating. The report is not ideal, for instance, it doesn’t display the column with the overall score and there is no user functionality to add it.

The purchasing manager runs the transaction and fills in the initial screen with Purchasing Organization, suppliers’ numbers, and the material (**Figure 20**). The next screen provides the list of evaluated suppliers with their main criteria scores, overall scores, and corresponding ranks (**Figure 21**). On the basis of this report, the decision on which supplier to choose is made.



Ranking List of Supplier Evaluations Based on Material/Material Group

General Data

Purchasing Organization	4010		
Supplier	100007	to	

Comparison Level

Material	CHILD_WORLD_TEST

Material Group Level

Material Group	
Mat. With/Without Master Rec.	

Scope of List

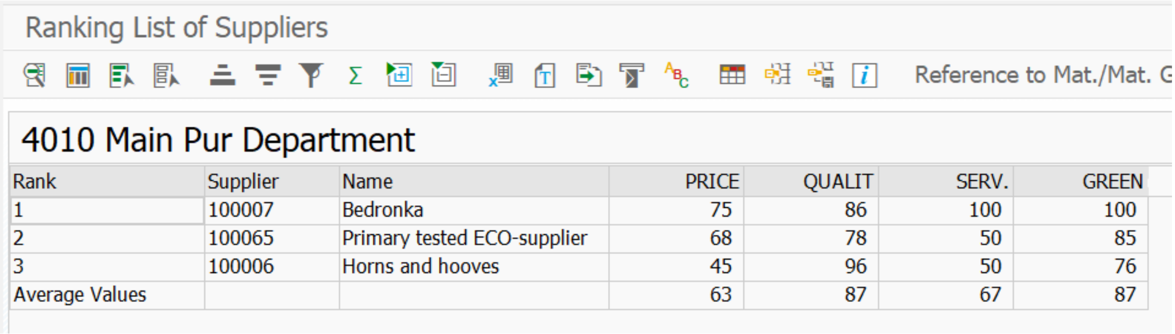
Scope of List	ZECOLIST

Multiple Selection for Supplier

Select Single Values (3) | Select R

S..	Single value
	100007
	100006
	100065

Figure 20: Ranking list of suppliers, initial screen



Ranking List of Suppliers

4010 Main Pur Department

Rank	Supplier	Name	PRICE	QUALIT	SERV.	GREEN
1	100007	Bedronka	75	86	100	100
2	100065	Primary tested ECO-supplier	68	78	50	85
3	100006	Horns and hooves	45	96	50	76
Average Values			63	87	67	87

Figure 21: Ranking list of suppliers

5.3 Custom modifications of SAP functionality

Now let’s consider the necessary configurations and implementation to make the model work in the way, described in the previous chapter. The configurations and implementations are explained based on SAP architecture, but such approach, business objects structure, and algorithm logic may be implemented in any IS, that have flexible tools for customers’ implementation. Moreover, other systems may be programmed on other more convenient languages that are more suitable for dynamic data structures.

Being one of the other IS, SAP supports a vast number of business processes and options for their application. To adopt the system for a personalized business process, SAP suggests a configuration menu (transaction SPRO[1]). The following chapters will outline the steps necessary to execute the model configurations. However, in practice, there may be instances where the SPRO settings are not enough. That is why software modifications are needed to change standard system algorithms. Creating a personalized supplier evaluation model is such a case.

Obviously, it would be impossible to predefine all the requirements of a client's business process, like, for instance, unique validation checks for input fields. A logical solution to enable the integration of custom logic is to permit the inclusion of program code in predefined locations. Of course, the method of such code integration may vary widely. There are two general approaches for custom code inclusion options.

The first (simple) option is to allow modifications to the standard code. As a result, it would be a system without any possibility of updating versions, which also does not allow the use of standard patches that correct problems of both security and business logic in applications. On the other side, such an approach allows customers to make their own implementation easily, and the issuing company wouldn’t be responsible for the support. The company “1C[2]”, which specializes in the development, distribution, and publishing of mass-market software to automate everyday enterprise activities, follows this approach.

The second option is the development of mechanisms for including user code in the standard business logic of the program. SAP company chose this option. The approach

[1] Transaction code SPRO is used configure the SAP System as per clients’ requirement. SPRO is abbreviation for SAP Project Reference Object. After executing SPRO transaction code, IMG (Implementation Management Guide) menu is displayed where customization settings for all modules are located [55].

[2] "1C" company specializes in development, distribution, publishing and support of mass-market software. The most known product is "1C:Enterprise 8", that is an ERP system. It's the main distributor of information systems on the Commonwealth of Independent States area [67].

guarantees that such a system can be upgraded without any problems if the code is correctly implemented at the extension point. The whole variety of enhancements to the SAP systems is described on the SAP Help Portal in the section “Enhancements to the Standard” [59].

In this work, the technique known as Customer exits is used to implement custom logic. Customer exit is a predetermined place in the standard code that is available to be changed by the customer. Basically, it is a function module[1] (FM) or just a function that is called in the code. The function has predefined input, changing, and output parameters, so it’s possible to influence only them. It’s possible to make selects from a database and implement some custom logic, but this logic must make influence only the changing and exporting parameters. Otherwise, the logic inside of the function will not have any influence on the process. It is a really popular approach to keep the balance between the ability for custom logic implementation and providing updates and patches safely without breaking the implemented logic.

Basically, customers’ logic implementations are made during a project. Different companies lead their projects differently, but SAP requires creating project inside of the system and connecting customer exits with it. It’s a good solution for companies without project managing systems, but for the ones who already have, such requirements generate additional work. Moreover, such an approach becomes challenging during writing the extension code by multiple developers, because the program becomes blocked by the user, who made the first change. The customer exit method was chosen because this is the only opportunity to enhance the standard functionality for supplier evaluation. Project creation is made through CMOD/SMOD[2] transactions in SAP.

In the process of current work, the following objects are enhanced:

- Quotation (via customer exit in CMOD/SMOD)
- Supplier evaluation methods (via customer exit in CMOD/SMOD)
- The business partner (via configuration in SPRO)

The fields that should be enhanced in the system to perform the evaluation according to the chosen parameters are marked as “Enhanced” in the column “Field type” of the **Table**

[1] Function modules (or FMs) are procedures that are defined in special ABAP programs only, so-called function groups, but can be called from all ABAP programs. Function groups act as containers for function modules that logically belong together [56].

[2] Customer exits call customer ABAP code at strategic moments to enhance the standard. Hence, customer exits act as 'Hook' points for the custom business functionality. CMOD/SMOD transactions are responsible for necessity configurations [54].

4. For the parameters that are supposed to use the standard data the mark is “Standard”. All custom enhancements have the naming rule: the first letter of the program's name must start with Y* or Z*. This rule allows to define whether the program was made by SAP developers or on the customer side [60].

To make necessary enhancements the common project ZSUPPEVL is created in the CMOD transaction. There are two custom enhancements were assigned to this project: MM06E005 “Customer fields in purchasing document” and MM06L001 “Exits to determine ratings in vendor evaluation” (see **Attachment 3**). The first one allows to addition custom fields to the purchasing documents. The second one allows to implementation custom logic for vendor evaluation.

5.3.1 Quotation enhancement

As shown in **Table 4**, the quotation data is used in the definition of several criteria. Unfortunately, the standard quotation view and data table don’t have the required fields. That’s why this object must be enhanced. The enhancement process is performed by making the following steps:

1. Add new necessary fields to the database
2. Draw fields on an appropriate screen the screen
3. Implement logic that relates savings and changes data on the screen and in the database.

All enhancement steps with the code and detailed description are presented in the reference [61], so to not copy the same information, only fields’ types and the quotation screen with new fields are represented in **Attachment 4**. Here is supposed, that all parameters, gathered from different suppliers, are entered in the same units of measure. It allows simplifying the model regarding units of measure conversion.

5.3.2 Business partner enhancement

A business partner is a much more flexible business object, compared to the quotations. SAP provides the possibility to add custom fields to BP view without ABAP developing. BP role FS0000 “Financial Service BP” has a view “Additional information”, that may be configured according to customers’ requirements. This view allows maintenance of all basic field types: text, Boolean, data, input from the list, and currency. All parameters are divided by categories that influence the fields of which types would be shown to a user.

A detailed configuration guide is provided by SAP in the reference [57]. Categories and respective field types are described in **Attachment 5**.

5.3.3 Supplier evaluation configuration

As was described previously, the standard supplier evaluation configurations are made on the Purchasing Organization level. This groping level for the custom developments should be the same. Supplier evaluation functionality enhancement is more complicated than creating new fields in Business Objects and has a strong relationship with the standard functionality. A deep explanation of the standard supplier evaluation functionality is given on the SAP Help Portal [37]. Only the steps required to set up the model are presented here.

First of all, it’s required to configure the supplier evaluation criteria and grouping level. Configuration is made in the transaction SPRO [55]. This transaction provides a configuration tree, where all standard provided business process settings are made. This tree is divided into processes and business objects. Usually, an exact passage is given in the corresponding manual.

To make configurations for the supplier evaluation criteria the following passages should be performed (**Figure 22**):

- SPRO -- Materials Management – Purchasing -- Supplier Evaluation -- Define Criteria. All criteria and sub-criteria configured here would be available for the entire system.
- SPRO -- Materials Management – Purchasing -- Supplier Evaluation -- Maintain Purchasing Organization Data. Here criteria and sub-criteria are assigned to the particular purchasing organization.

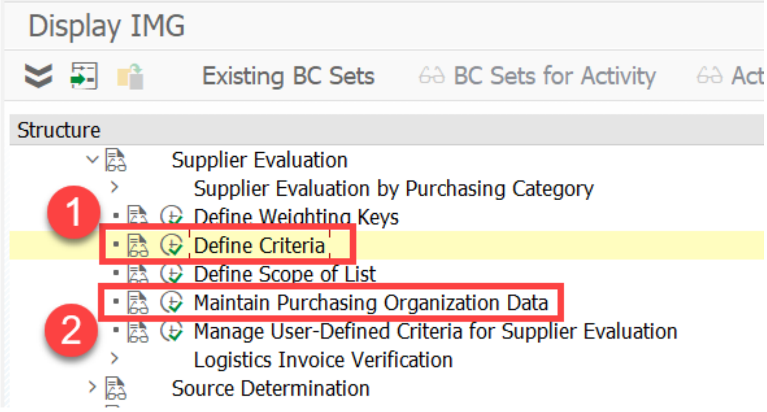
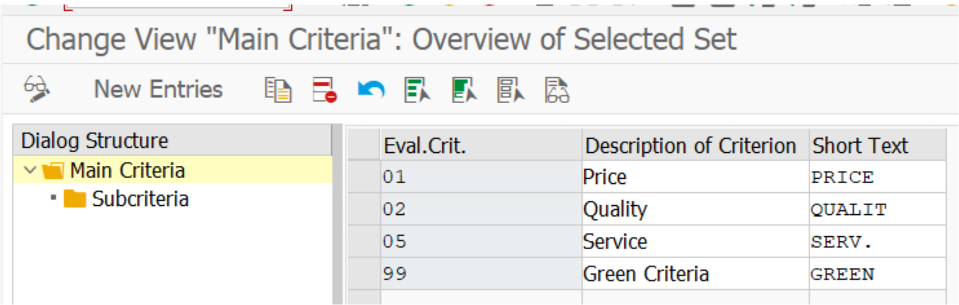

Display IMG
Existing BC Sets
BC Sets for Activity
Act
Structure
Supplier Evaluation
Supplier Evaluation by Purchasing Category
Define Weighting Keys
Define Criteria
Define Scope of List
Maintain Purchasing Organization Data
Manage User-Defined Criteria for Supplier Evaluation
Logistics Invoice Verification
Source Determination
1
2

Figure 22: SPRO passage: 1 - Criteria definition, 2 - Maintain PO data

For the current model the following criteria were configured in the dialog structure "main criteria": 01 "Price", 02 "Quality", 05 "Service" and 99 "Green criteria" (**Figure 23**), which are described in **Table 4**. Here the criteria code and long and short descriptions are defined.

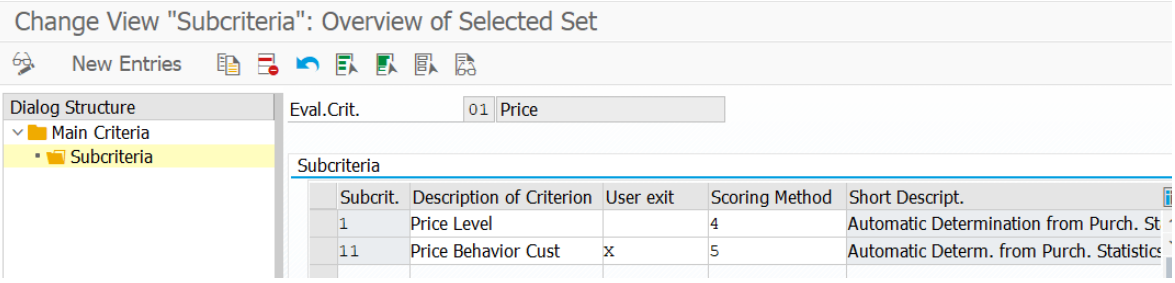


Change View "Main Criteria": Overview of Selected Set

Eval.Crit.	Description of Criterion	Short Text
01	Price	PRICE
02	Quality	QUALIT
05	Service	SERV.
99	Green Criteria	GREEN

Figure 23: Configured evaluation criteria

In the dialog structure, "Sub-criteria" detailed sub-criteria data are maintained. The relations between criteria and sub-criteria are in **Table 4**. In addition to the sub-criteria code and its name, the scored method is maintained as well. In this work, only one standard scoring method is used to determine the "Price level" sub-criteria (**Figure 24**). All other sub-criteria use custom logic. To make the system understand that the scoring must be done according to custom logic, the sign "X" is assigned in the field "User exit". **Figure 24**, **Figure 25**, **Figure 26**, **Figure 27** show necessary sub-criteria configurations. The decision on what logic to use is made on the button pushing, which is explained in **Chapter 5.2.4**.



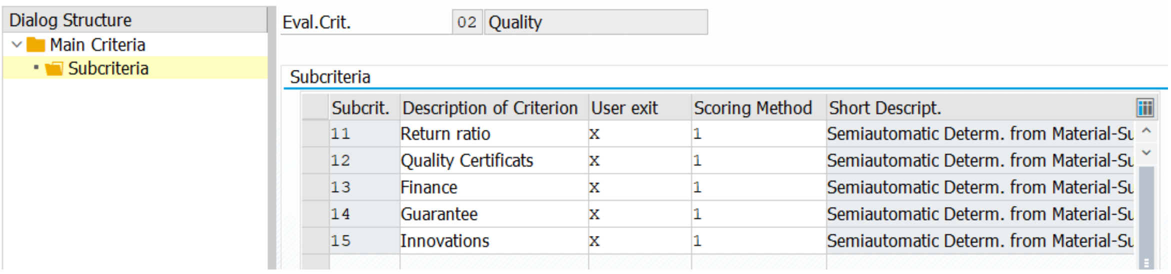
Change View "Subcriteria": Overview of Selected Set

Eval.Crit. 01 Price

Subcriteria

Subcrit.	Description of Criterion	User exit	Scoring Method	Short Descript.
1	Price Level		4	Automatic Determination from Purch. St
11	Price Behavior Cust	X	5	Automatic Determ. from Purch. Statistics

Figure 24: Configured evaluation sub-criteria for Price

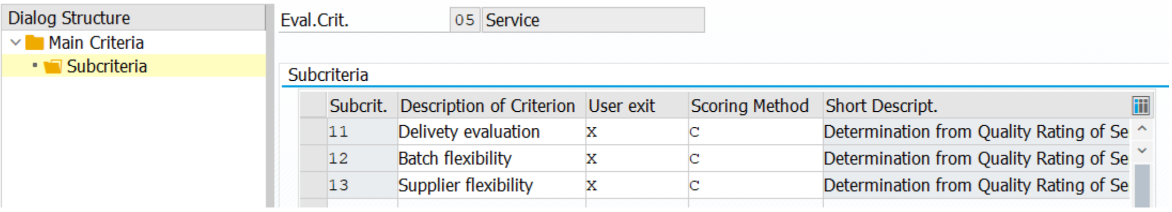


Eval.Crit. 02 Quality

Subcriteria

Subcrit.	Description of Criterion	User exit	Scoring Method	Short Descript.
11	Return ratio	X	1	Semiautomatic Determ. from Material-Su
12	Quality Certificats	X	1	Semiautomatic Determ. from Material-Su
13	Finance	X	1	Semiautomatic Determ. from Material-Su
14	Guarantee	X	1	Semiautomatic Determ. from Material-Su
15	Innovations	X	1	Semiautomatic Determ. from Material-Su

Figure 25: Configured evaluation sub-criteria for Quality



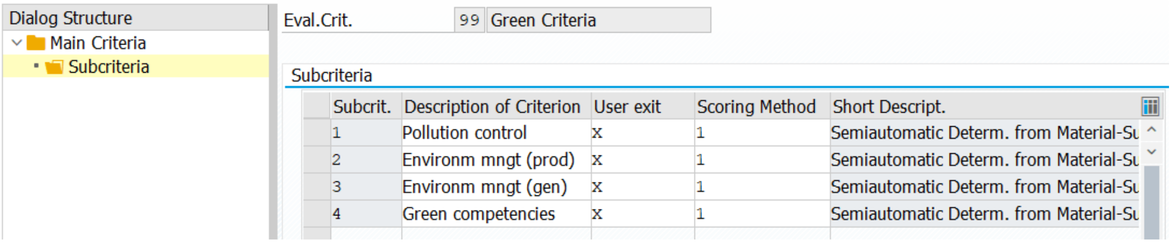
Dialog Structure
- Main Criteria
  - Subcriteria

Eval.Crit. 05 Service

Subcriteria

Subcrit.	Description of Criterion	User exit	Scoring Method	Short Descript.
11	Delivety evaluation	X	C	Determination from Quality Rating of Se
12	Batch flexibility	X	C	Determination from Quality Rating of Se
13	Supplier flexibility	X	C	Determination from Quality Rating of Se

Figure 26: Configured evaluation sub-criteria for Service



Dialog Structure
- Main Criteria
  - Subcriteria

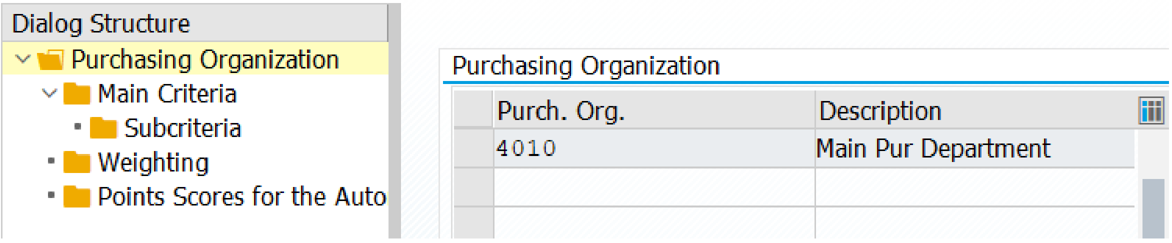
Eval.Crit. 99 Green Criteria

Subcriteria

Subcrit.	Description of Criterion	User exit	Scoring Method	Short Descript.
1	Pollution control	X	1	Semiautomatic Determ. from Material-Su
2	Environm mngt (prod)	X	1	Semiautomatic Determ. from Material-Su
3	Environm mngt (gen)	X	1	Semiautomatic Determ. from Material-Su
4	Green competencies	X	1	Semiautomatic Determ. from Material-Su

Figure 27: Configured evaluation sub-criteria for Green Criteria

After the criteria and sub-criteria definition, it's required to maintain Purchasing Organization Data. For this goal, the Purchasing Organization 4010 "Main Pur Department" is copied from the standard 0001 Purchasing Organization. The view of the "Maintain Purchasing Organization Data" passage looks as follows (**Figure 28**):



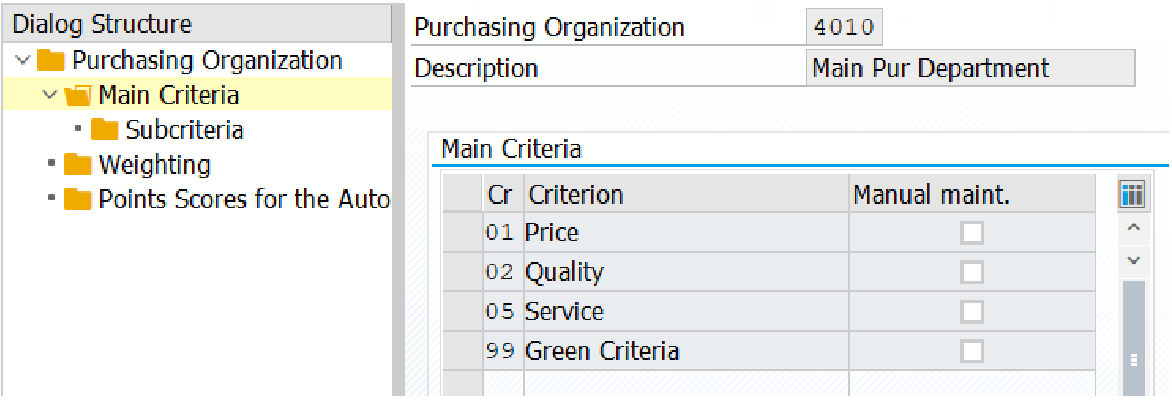
Dialog Structure
- Purchasing Organization
  - Main Criteria
    - Subcriteria
  - Weighting
  - Points Scores for the Auto

Purchasing Organization

Purch. Org.	Description
4010	Main Pur Department

Figure 28: Maintain Purchasing Organization Data view

Here under 4010 purchasing organization the following criteria were assigned on the dialog structure "Main criteria" (**Figure 29**). SAP allows maintenance of some criteria manually if required (put the mark "Manual maint." for the corresponding criteria).



Dialog Structure
- Purchasing Organization
  - Main Criteria
    - Subcriteria
  - Weighting
  - Points Scores for the Auto

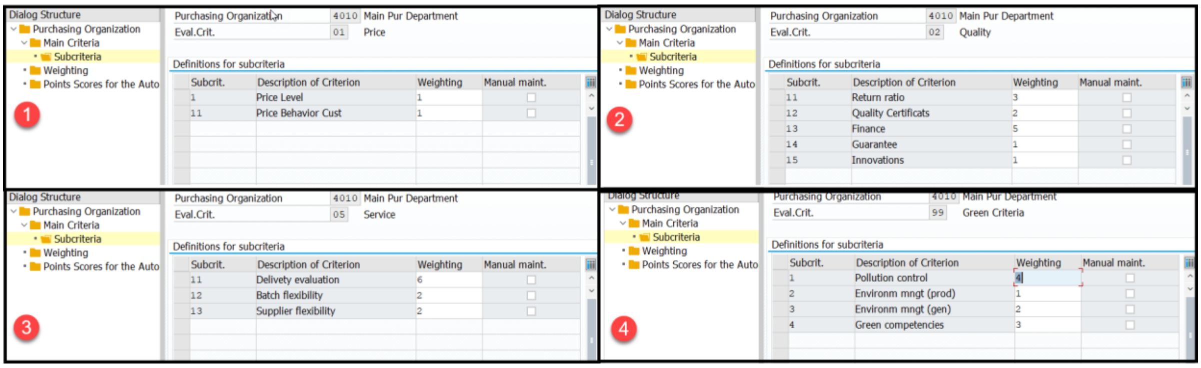
Purchasing Organization 4010
Description Main Pur Department

Main Criteria

Cr	Criterion	Manual maint.
01	Price	□
02	Quality	□
05	Service	□
99	Green Criteria	□

Figure 29: "Main criteria" dialog structure. Purchasing organization assignment

In the dialog “Sub-criteria” corresponding sub-criteria and their weight are assigned to the main criteria (**Figure 30**). The sub-criteria weights are used during the score calculation for the main criteria. The system calculates the main criteria score according to the configured here weights. In cases when there is no data for the sub-criteria calculation, the corresponding weight is not considered. This process is shown in the **Figure 16** (bottom gray block). It’s possible, as well, to mark sub-criteria for manual maintenance.



Purchasing Organization 4010 Main Pur Department
Eval.Crit. 01 Price

Definitions for subcriteria

Subcrit.	Description of Criterion	Weighting	Manual maint.
1	Price Level	1	□
11	Price Behavior Cust	1	□

Purchasing Organization 4010 Main Pur Department
Eval.Crit. 02 Quality

Definitions for subcriteria

Subcrit.	Description of Criterion	Weighting	Manual maint.
11	Return ratio	3	□
12	Quality Certificats	2	□
13	Finance	5	□
14	Guarantee	1	□
15	Innovations	1	□

Purchasing Organization 4010 Main Pur Department
Eval.Crit. 05 Service

Definitions for subcriteria

Subcrit.	Description of Criterion	Weighting	Manual maint.
11	Delivety evaluation	6	□
12	Batch flexibility	2	□
13	Supplier flexibility	2	□

Purchasing Organization 4010 Main Pur Department
Eval.Crit. 99 Green Criteria

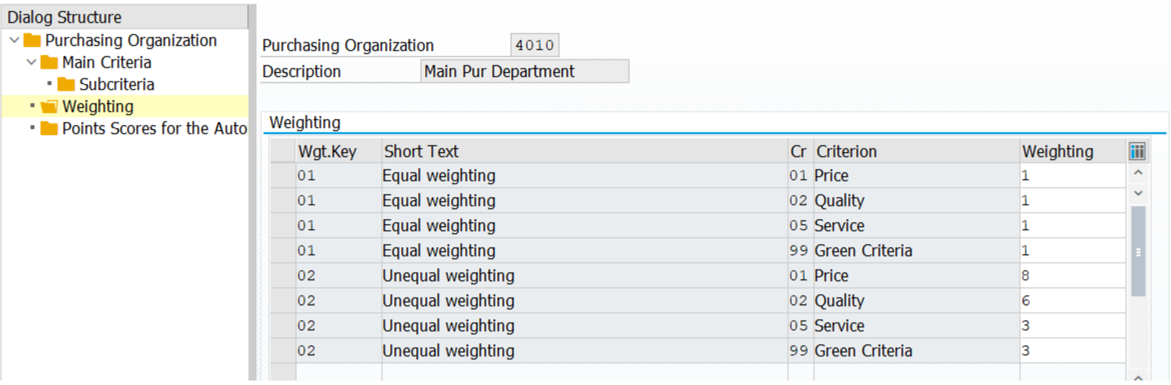
Definitions for subcriteria

Subcrit.	Description of Criterion	Weighting	Manual maint.
1	Pollution control	4	□
2	Environm mngt (prod)	1	□
3	Environm mngt (gen)	2	□
4	Green competencies	3	□

Figure 30: Sub-criteria dialog and weight assignment. 1 – Price, 2 – Quality, 3 – Service, 4 – Green Criteria.

The SS model and PROMETEE II method in particular are supposed to use weights for the parameters. In the model the weights are used on all three levels: during the sub-criteria score definition (**Figure 16**, bottom gray block), on the main criteria definition (**Figure 16**, middle gray block), and during the final or overall score definition (**Figure 16**, top gray block). SAP provides the possibility to configure weight only on the last two levels (**Figure 30** and **Figure 31**).

The IS suggests configuring weighting keys or in other words several weighting sets for the main criteria. It allows users to change weights immediately during the supplier evaluation process by choosing an appropriate key, as described in **Chapter 5.2.4** (**Figure 19**). During long periods the company’s priorities may be changed, so a new key may be configured without losing the previous values. Basically, the weight sizes are defined by the companies themselves. To find the most relevant weight values it’s required real data analysis. The search for those weight values may become a topic for separate scientific work. There are two weighting keys configured for this work (**Figure 31**).



Purchasing Organization 4010

Description Main Pur Department

Weighting

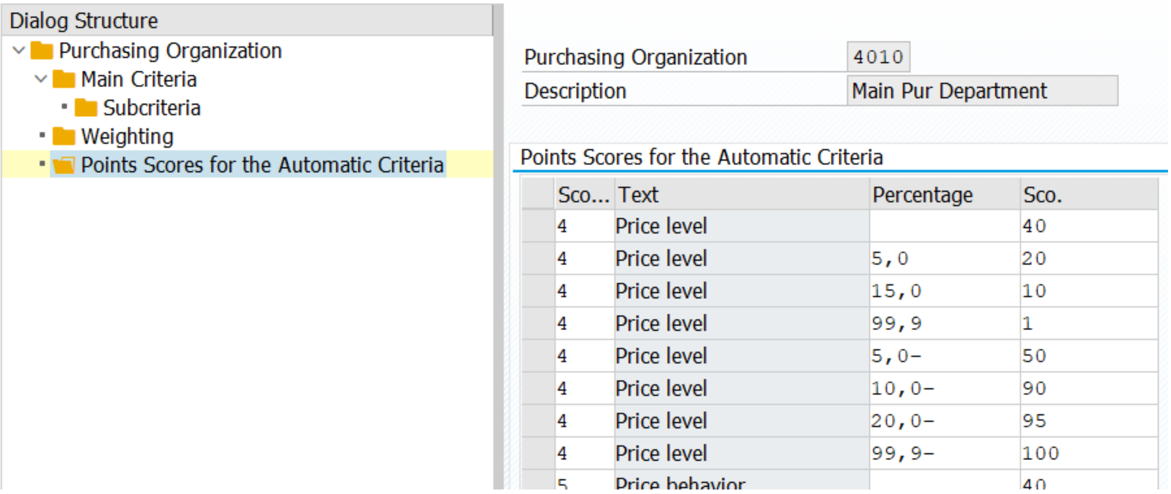
Wgt.Key	Short Text	Cr	Criterion	Weighting
01	Equal weighting	01	Price	1
01	Equal weighting	02	Quality	1
01	Equal weighting	05	Service	1
01	Equal weighting	99	Green Criteria	1
02	Unequal weighting	01	Price	8
02	Unequal weighting	02	Quality	6
02	Unequal weighting	05	Service	3
02	Unequal weighting	99	Green Criteria	3

Figure 31: Weighting keys for main criteria dialog

Being one of many IS that solve the most common business issues, standard SAP scoring functionality suggests some not-obvious but predefined scoring methods. The problem with those predefined methods is the lack of detailed documentation. Lots of hours were spent debugging the program while the method’s logic became clear. Each method returns a calculated value, which is then assigned a score based on the scale configured in the “Points Scores for the Automatic Criteria” dialog (**Figure 32**). The scale works in the following way: first, the system finds the percent difference between the product price and market price (the value in the SAP system, that reflects the current market price for the material or a group of the materials; basically is maintained by the purchasing department), then the system searches an appropriate interval and assigns the score related to the upper bound. For instance, the market price is 1000 NOK, and the product price from the supplier is 1075 NOK. The percentage difference is

$$\frac{1075-1000}{1000} * 100 = 7.5\%.$$

That means that the price races at 7.5%. According to the scale from **Figure 32**, there is no exact score for this value. Then the system takes the interval from 5% to 15% and assigns the score corresponding to its upper bound, which is equal to 10 for 15%. The main disadvantage of such an approach is that suppliers with different results would be assigned a similar score. In the given example, suppliers with prices in-between 1050 and 1150 would achieve the same score. The PROMETEE II method avoids this problem. The only suppliers with the same value would be assigned the similar score.



Dialog Structure
- Purchasing Organization
  - Main Criteria
    - Subcriteria
  - Weighting
  - Points Scores for the Automatic Criteria

Purchasing Organization	4010
Description	Main Pur Department

Points Scores for the Automatic Criteria

Sco...	Text	Percentage	Sco.
4	Price level		40
4	Price level	5,0	20
4	Price level	15,0	10
4	Price level	99,9	1
4	Price level	5,0-	50
4	Price level	10,0-	90
4	Price level	20,0-	95
4	Price level	99,9-	100
5	Price behavior		40

Figure 32: Points Scores for the Automatic Criteria

As most of the sub-criteria are configured to be calculated with custom logic, it's required to implement the necessary logic into the available customer exit. The customer exit is provided inside of the EXIT_SAPLMEL0_001 FM (**Figure 15**, gray rectangle). The function imports the following parameters: Supplier, Purchasing Organization, Main Criterion, Sub-criterion, and Evaluation Method (**Figure 15**, blue left arrow). The function exports the only perimeter – score (**Figure 15**, blue right arrow). This score then is assigned to the corresponding sub-criteria. Unfortunately, the functional module doesn't have material and the range of evaluated suppliers as importing parameters. Those parameters are required to implement the PROMETEE II method. It's more practical to implement separate methods to calculate the score for each sub-criterion instead of writing the full logic inside of the same function. So more complex and more flexible architecture is required. The necessary architecture is described in **Chapter 5.4**.

5.4 Model architecture description

5.4.1 Maintenance tables

The EXIT_SAPLMEL0_001 FM has restricted parameters (**Figure 15**). It means that a custom configuration is required. Importing given parameters allows for defining necessary data and making appropriate calculations. To solve this problem, the following tables were developed (**Figure 15**, orange rectangle on the bottom):

- ZTSUP_EV_EKORG "Purchasing organization for the supplier evaluation" (**Table 5**). This table collects Purchasing organizations, that are configured for the custom logic. Serves as the top grouping level. The maintenance view is represented in **Figure 36**.

- ZSUPEV_OBJDEF “Handler class definition PO/Criterion” (**Table 6**). This table subordinates to the ZTSUP_EV_EKORG by the field “Purchasing organization”. Here the custom Class names are stored. The system defines classes for the corresponding criteria. The table store the log objects that are used to record the performed actions. The maintenance view is represented in **Figure 37**.
- ZSUPEV_SUBOBJDEF “Defining Methods for Supplier Evaluation Sub Criteria” (**Table 7**). This table subordinates to the ZSUPEV_OBJDEF by the fields “Purchasing organization” and “Key for evaluation criterion”. The maintenance view is represented in **Figure 38**.
- ZTSUPEV_MARC “Database for evaluated materials” (**Table 8**). This table subordinates to the ZTSUP_EV_EKORG by the field “Purchasing organization”. This table let understand which material is used for the evaluation by marking it with the radio button. The maintenance view is represented in **Figure 39.**
- ZTSUPEV_LFM1 “Suppliers for evaluation” (**Table 9**). This table subordinates to the ZTSUP_EV_EKORG by the field “Purchasing organization”. This table provides the list of suppliers that are evaluated at the corresponding Purchasing Organization level. The maintenance view is represented in **Figure 40**.

Table 5: Purchasing organization for the supplier evaluation (ZTSUP_EV_EKORG) table description

Field name	Key	Type	Description	Comment
EKORG	Yes	CHAR 4	Purchasing organization	The program checks whether the given purchasing organization in the custom table. If yes, then the program uses the custom logic. It allows to divide different logic between purchasing organizations according to the standard approach.

Table 6: Handler class definition PO/Criterion (ZSUPEV_OBJDEF) table description

Field name	Key	Type	Description	Comment
EKORG	Yes	CHAR 4	Purchasing organization	Here Purchasing organization is used to make relationship with the table ZTSUP_EV_EKORG.
HKRIT	Yes	CHAR 2	Key for evaluation criterion	Here is the code for the main criteria is stored. At the level of given criteria system defenses the programming Class that allows to handle necessary logic.

CLSNAME	No	CHAR 30	Object Type Name	The programming Class name is assigned here. Such an approach allows to define different custom Classes for different criteria that makes new instrument more flexible then standard one.
SUBOBJECT	No	CHAR 20	Application Log: Subobject	The field store the log object name. This object allows to store messages and errors during supplier evaluation. It allows to analyze problems even if the uses have closed the transaction.

Table 7: Defining Methods for Supplier Evaluation Sub Criteria (ZSUPEV_SUBOBJDEF) table description

Field name	Key	Type	Description	Comment
EKORG	Yes	CHAR 4	Purchasing organization	Here Purchasing organization is used to make relationship with the table ZSUPEV_OBJDEF.
HKRIT	Yes	CHAR 2	Key for evaluation criterion	Here the key is used to make relationship with the table ZSUPEV_OBJDEF.
TKRIT	Yes	NUMC 2	Subcriterion	Sub-criterion text. It's used to define calculation method to gain corresponding sub-criterion score.
ZMETH_NUM	Yes	CHAR 2	Sequence method number for vendor evaluation	In case if the logic of the score is supposed more than one method, it's possible to assign the sequence number.
METHOD_NAME	No	CHAR 61	Full Component Name	The name of the method that is used to calculate the score for the sub-criteria. The Class is given in the table ZSUPEV_OBJDEF. The fields "Purchasing Organization" and "Key for evaluation criterion" provide necessary relation.
ZZ_WEIGHT_TABLE	No	CHAR 16	Table name, 16 characters	In case, when the parameters for the criteria definition require weight, it's possible to insert the name of the table that is used to take weights into account.
ZIS_INACTIV	No	CHAR 1	Functionality not active?	Sometimes business situations require to switch off the custom logic immediately. This indicator allows to make it.

Table 8: Data base for evaluated materials (ZTSUPEV_MARC) table description

Field name	Key	Type	Description	Comment
EKORG	Yes	CHAR 4	Purchasing organization	Here Purchasing organization is used to make relationship with the table ZTSUP_EV_EKORG.
MATNR	Yes	CHAR 40	Material Number	The number of materials that are used to evaluate suppliers.
WERKS	Yes	CHAR 4	Plant	Plant is an organizational level in SAP. Usually it corresponds to the physical place for the material production. The Material Master

				Data is subordinated to the plant level.
ZZ_MAT_TO_EVAL	No	CHAR 1	Material to evaluate	Radio button. According to the simplifications, the only one material is used for the evaluation.

Table 9: Suppliers for evaluation (ZTSUPEV_LFM1) table description

Field name	Key	Type	Description	Comment
EKORG	Yes	CHAR 4	Purchasing organization	Here Purchasing organization is used to make relationship with the table ZTSUP_EV_EKORG.
LIFNR	Yes	CHAR 10	Vendor's account number	SAP vendor’s unique code.

All those tables are gathered in the single maintenance view[1] ZVC_SUPPL_EVAL “View cluster for vendor evaluation settings”, which is represented in **Attachment 6**.

5.4.2 Programming Classes development

The dynamic programming principle is used to make a flexible architecture. It means that a “calling” class is required. This “calling” class is implemented to the customer exit (**Figure 15**, blue ellipse), the corresponding customer-exit code is represented in **Attachment 8**. The constructor of the class is going to have the same input parameters as the customer-exit function EXIT_SAPLMEL0_001 (**Figure 15**, blue left arrow). Using the input parameters, the “calling” class selects the “executing” class and methods from the configuration tables ZSUPEV_OBJDEF and ZSUPEV_SUBOBJDEF and calls them (**Figure 15**, the blue arrow from the blue ellipse toward the database, parameters in the left green rectangle are used; blue arrow from the database toward the yellow ellipse, parameters in the right green rectangle are returned). The architecture and code of the “calling” class are represented in **Attachment 9**. Such architecture allows the implementation of the “calling” class once into the customer exit. Then it is possible to develop customer's classes and methods and record them via the cluster view, described in **Attachment 6**. There is no need to create enhancement projects and find appropriate places inside the standard code. This flexible architecture allows the easy launch of customers’ logic. This approach simplifies architecture maintenance since it is enough to analyze a separate method to find an error in the calculation of a single sub-criteria, instead of analyzing the entire customer-exit logic.

[1] A maintenance view is a special view for performing writes on multiple tables using extended table maintenance. A single maintenance view can be used to modify the content of multiple related database table consistently. A maintenance view is not defined on the database, it only connects data base tables and represent them to the user in a predefined, convenient way [20].

After the scores for the sub-criteria were calculated, the system uses the standard configured weights, explained in **Chapter 5.3.3** “Supplier evaluation ”, **Figure 30**, to calculate the main criteria score. Then on the basis of the weighting key, entered by the user, and the weight, assigned to that key (**Figure 31**) the system calculates the final score. The explained architecture may be represented as shown in **Figure 16**.

According to the architecture each sub-criteria has its separate executing method (**Figure 38**). Each separate method gets the same parameters, as the FM EXIT_SAPLMEL0_001, and returns the score related to the corresponding criteria (**Figure 15**). Generally, public methods of executing class consist of two parts: data collection and PROMETHEE II algorithm usage. The logic of executing the class’s public methods is represented in **Chapter 5.5** “PROMETEE II supplier selection model on ABAP”.

It was decided to make the executing class the “heir[1]” of the calling class because the executing class requite the same attributes. Being the heir of the calling class, the executing Class type is the same. That allows for avoiding type mismatch problems in dynamic programming. The executing class methods could be divided into three groups: inherited base class methods, public methods[2] for calculating sub-criteria scores, and helper methods (PROMETEE II logic, data transformation, and so on). The names of the calculating sub-criteria score methods are stored in the table ZSUPEV_SUBOBJDEF “Defining Methods for Supplier Evaluation Sub Criteria” (**Table 7**, **Figure 38**). Other methods use inside of those public methods. The architecture and code of the “executing” class are represented in **Attachment 10**.

5.5 PROMETEE II supplier selection model on ABAP

5.5.1 Numeric example

Let’s consider the PROMETEE II algorithm on the “Pollution control” sub-criteria definition with testing data. **Chapter 4.1** “Supplier selection method” describes the algorithm steps, but before executing them the data collection step is required. As was described in **Table 4**, to calculate the score for the “Pollution control” sub-criteria, the

[1] Subclasses, derived classes, heir classes, or child classes are modular derivative classes that inherit one or more language entities from one or more other classes (called superclass, base classes, or parent classes). The semantics of class inheritance vary from language to language, but commonly the subclass automatically inherits the instance variables and member functions of its super classes [48].

[2] Public methods are methods that are accessible both inside and outside the scope of your class. Any instance of that class will have access to public methods and can invoke them [48].

following data is required: carbon footprint, waste water, solid non-recyclable wastes, energy consumption, and use of harmful/hazardous material. This data is stored in the Quotation business object (**Figure 35**). So, the data selection is made. The result of this selection is the three quotations from three different suppliers with the data, represented in **Table 10**. Quotation number and quotation creation date are required for the initial checks, that are not related to the PROMETEE II logic.

Table 10: PROMETEE II example. Step 0: data selection

Index	Quotation No	Supplier No	Creation Date	Carbon footprint kgCO2e / item	Waste water liter / item	Solid wastes KG / item	Energy consumption kWh / item	Harmful material KG per item
1	6000000015	100006	18.03.2023	0,025	1,500	0,100	200,000	0,010
2	6000000016	100007	18.03.2023	0,030	1,400	0,085	230,000	0,020
3	6000000014	100065	18.03.2023	0,020	1,550	0,099	210,000	0,005

Table 10 has row data that should be normalized (Step 1 of the PROMETEE II algorithm). The method FILL_EVALUATION_VALUES of the class ZCL_CUSTOM_SUPPL_ASSESSMENT is used for this purpose. Here all criteria are non-beneficial, because the less their values are, the better the supplier score. The result of the normalized data is represented in **Table 11**. The value 1 means, that this supplier-suggested value is the best in this category, and vice versa for 0 value. Let’s look at the “Energy consumption” column. The value 200 – is the best one among others, as the less energy the supplier spends to produce one item of the material, the more ecological it is. The value 230 is the worst one, and the normalized value is 0. The value 0,667 is calculated by the formula (2): $\frac{\max(200,210,230)-210}{\max(200,210,230)-\min(200,210,230)} = \frac{230-210}{230-200} = \frac{20}{30} = 0,(6) \approx 0,667$. The other values were calculated in a similar way.

Table 11: PROMETEE II example. Step 1: data normalization

Index	Supplier No	Carbon footprint normalized value	Waste water normalized value	Solid wastes normalized value	Energy consumption normalized value	Harmful material normalized value
1	100006	0,500	0,333	0,000	1,000	0,667
2	100007	0,000	1,000	1,000	0,000	0,000
3	100065	1,000	0,000	0,067	0,667	1,000

Then on the basis of **Table 11**, it’s required to perform steps 2 and 3. Despite the fact, that those steps are explained separately in the algorithm, technically it’s more effective to perform them simultaneously. The results of steps 2 and 3 are represented in **Table 12**.

The formula (3) is used. Let's use the formula on the example of the suppliers 100006 and 100007 for the criterion "Carbon footprint". $P_{1(100006,100007)} = (0{,}5 - 0) = 0{,}5$. $0{,}5 > 0, \rightarrow P_{1(100006,100007)} = 0{,}5$. The calculations for the criterion "Waste water" are the following: $P_{2(100006,100007)} = (0{,}333 - 1) = -0{,}777$. $-0{,}777 < 0, \rightarrow P_{2(100006,100007)} = 0$.

Table 12: PROMETEE II example. Steps 2 and 3: calculate the evaluative differences and use a preference function. Brown rows – the comparison of the same suppliers. May be discarded.

Supplier A	Supplier B	Carbon footprint normalized value	Waste water normalized value	Solid wastes normalized value	Energy consumption normalized value	Harmful material normalized value
100006	100006	0	0	0	0	0
100006	100007	0,5	0	0	1	0,667
100006	100065	0	0,333	0	0,333	0
100007	100006	0	0,667	1	0	0
100007	100007	0	0	0	0	0
100007	100065	0	1	0,933	0	0
100065	100006	0,5	0	0,067	0	0,333
100065	100007	1	0	0	0,667	1
100065	100065	0	0	0	0	0

Step 4 aggregates the values from **Table 12** adjusting the values according to the given weights. Here it is assumed, that all data for the "Pollution control" sub-criteria has the same significance. That means that the weight of each parameter is 1, and the summary weight is 5. The result of step 4 is represented in **Table 13** by using the formula (4). To calculate the value 0,433 for the supplier 100006 in the row and 100007 in the column the following calculations are made: $\frac{0{,}5*1+0*1+0*1+1*1+0{,}667*1}{5} = \frac{2{,}167}{5} \approx 0{,}433$. Other values are calculated in a similar way. Steps 2, 3, and 4 are performed by the method fill_calculation_matrix of the class ZCL_CUSTOM_SUPPL_ASSESSMENT.

Table 13: PROMETEE II example. Step 4: aggregated values

	100006	100007	100065
100006	0,000	0,433	0,133
100007	0,333	0,000	0,387
100065	0,180	0,533	0,000

According to step 5, the positive and negative flows are calculated. The formulas (5) and (6) are used. In other words, the positive flow of supplier *i* is the sum of the values in the row for the corresponding supplier in **Table 13**. The negative flow is the sum of the values in the column for the respective supplier. The result of step 5 is in **Table 14.**

Table 14: PROMETEE II example. Step 5: positive and negative flows determination

Supplier	Φ^{+}	Φ^{-}
100006	0,567	0,513
100007	0,720	0,967
100065	0,713	0,520

The net flow is calculated in step 6 by the formula (7). This is the difference between columns Φ^{+} and Φ^{-} of **Table 14.** The result is in **Table 15**. Values in column Φ represent the rank of the supplier. The higher the value, the higher the supplier rank. According to the values in column Φ, it's possible to evaluate how far each supplier is from the others. On the basis of the values in the column Φ, the corresponding score is assigned to the sub-criterion. As the sub-criteria's score is between 0 and 100, the following technique is used: $\frac{\Phi_i - min(\Phi)}{|\max(\Phi) - min(\Phi)|} \times 100$. The final score is represented in the column "Score" of **Table 15**. The same score is represented in **Figure 17**.

Table 15: PROMETEE II example. Step 6: the net flow calculation

Supplier	Φ	**Score**
100006	0,053	68
100007	-0,247	0
100065	0,194	100

5.5.2 Model difficulty analysis

The SS method PROMETHEE II is an MCDM technique that is commonly used for evaluating and ranking alternatives based on multiple criteria. The study divides the criteria into sub-criteria and sub-criteria by the data (**Figure 16**). The difficulty of the "PROMETHEE II" method can increase with an increasing number of evaluated suppliers, enlarging the preference function difficulty, and expanding the set of estimated parameters.

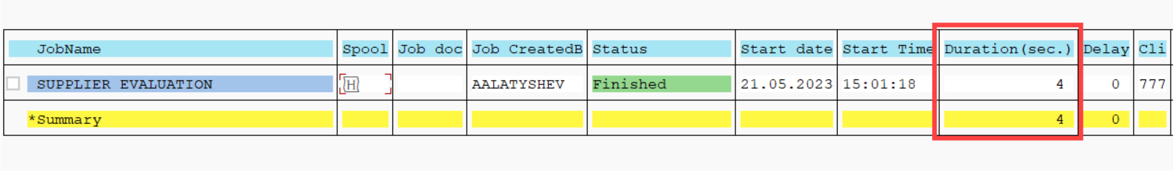
In the "PROMETHEE II" method, pairwise comparisons are performed between all pairs of suppliers based on each criterion to establish preference relationships. With a larger number of suppliers, the number of pairwise comparisons to be made significantly increases. So for two suppliers, the number of pairwise comparisons is $2 \times 2 - 2 = 2$ (the comparisons between themselves are not counted). For the three it is $3 \times 3 - 3 = 6$ (represented in **Table**

12). For five it is 20, for 10 it is 90, and so on. The general formula may be represented as $(N^2 - N)$, where N is the number of evaluated suppliers. Increasing the number of evaluated suppliers enlarges the algorithm difficulty exponentially.

The preference function usage step may influence the model difficulty as well. The calculation time may increase when the logic is not constant and takes into account some changed parameters such as the number of criteria or suppliers. In this study the preference function doesn't depend on the number of suppliers, criteria, or any other changeable parameter, it just performs the same operation each time.

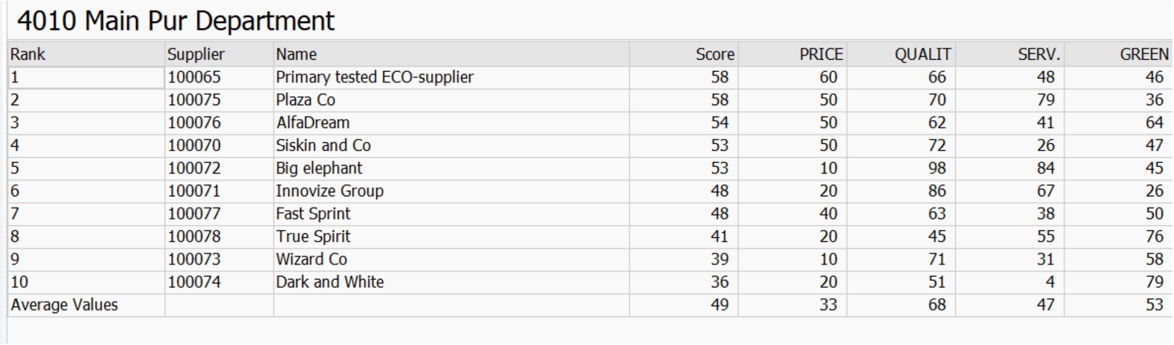
The larger the number of evaluated data parameters, the harder to interpret the results and make decisions. Actions, such as analyzing the preference rankings, understanding the impact of each data parameter, and drawing meaningful conclusions become more complex. For the range of three suppliers with one data parameter, it's required to make $3 \times 3 - 3 = 6$ actions. For the same range with two data parameters, it takes $(3 \times 3 - 3) \times 2 = 12$ actions. With three it takes $(3 \times 3 - 3) \times 3 = 18$ actions, with five – $(3 \times 3 - 3) \times 5 = 30$, with ten – 60. The growth of the number of data parameters makes the algorithm difficulty increase linearly. The general difficulty may be represented as $(N^2 - N) \times M$, where N is the number of evaluated suppliers, and M is the number of data parameters.

The model with 10 suppliers is configured in the system and run in the background to overview its performance. The data considered in the example is represented in **Attachment 7**. **Table 16** and **Table 17** represent the input data that is taken to perform the evaluation. The suppliers were evaluated within 4 seconds (**Figure 33**, see column "Duration (sec.)"). The final result, based on the input data, is represented in **Figure 34**. So it's possible to conclude that the model can solve business tasks with 10 suppliers and has the potential to be enlarged.



	JobName	Spool	Job doc	Job CreatedB	Status	Start date	Start Time	Duration(sec.)	Delay	Cli
□	SUPPLIER EVALUATION	H		AALATYSHEV	Finished	21.05.2023	15:01:18	4	0	777
	*Summary							4	0	

Figure 33: Model execution time (duration column is in seconds)



4010 Main Pur Department

Rank	Supplier	Name	Score	PRICE	QUALIT	SERV.	GREEN
1	100065	Primary tested ECO-supplier	58	60	66	48	46
2	100075	Plaza Co	58	50	70	79	36
3	100076	AlfaDream	54	50	62	41	64
4	100070	Siskin and Co	53	50	72	26	47
5	100072	Big elephant	53	10	98	84	45
6	100071	Innovize Group	48	20	86	67	26
7	100077	Fast Sprint	48	40	63	38	50
8	100078	True Spirit	41	20	45	55	76
9	100073	Wizard Co	39	10	71	31	58
10	100074	Dark and White	36	20	51	4	79
Average Values			49	33	68	47	53

Figure 34: The result of the evaluation of suppliers (10 ones). Weightings: Price – 40%, Quality – 30%, Service – 15%, Green Criteria – 15%.

6 Conclusion

6.1 Discussion

As a result of the research the green criteria are discovered that could be used in practice. Despite the fact that those criteria may be different from company co company, lots of articles suggest almost the same basis with insignificant differences. For instance, the basis of articles [13] and [29] are criteria set, that was created in the 1960s. The current study aligns with the idea, that the core criteria such as price, quality, delivery, and service are still relevant. This study shows, that some criteria, suggested in [13] and [29], such as the effectiveness of the supplier’s transportation, the capability of the supplier’s software to serve business processes, the efficiency of plan layout, and so on are really hard to explore for the purchasing company. Such criteria seem relevant theoretically, but in practice, they are almost impossible to collect, normalize and evaluate. Moreover, it’s still unclear how to store those parameters in the IS. But generally, criteria that are provided in observed research are relevant to use in the IS.

The fact that was discovered during the work is that IS's environmental solutions are really limited. The green SS problem and green criteria had been started to explore in 2009 (the most cited reference is [4]), but still, there is a lack of turnkey IS solutions for storing green parameters without the implementation of custom logic or fields. The work is done on an SAP basis, so this system is explored more than others. Information about its green and environmental solutions exists, but at least not in open access.

The method PROMETEE II is chosen for the SS problem-solving. The current work proves, that the method is suitable to use it in practice. Nishant Agrawal [27] explains the steps really thoroughly, but during making programming some steps are joined into one to improve the algorithm efficiency. This study acknowledges its conclusion, that the PROMETEE II method reduces the complexity of the SS and has clear steps. At the same time, this technique can solve a quite complex problem and allows assigning preferences in the form of appropriate weights. Those conclusions align with Vahid Balali and his colleagues' results as well [28]. So, PROMETHEE II is an effective algorithm for the companies’ managers because it is easy to understand calculation steps and manage necessary weights.

During the configuration process, it was found out that the forums like [49], [50], [51], [52] have more useful information regarding the step-by-step configuration and implementation guides. Even though the IS provides a flexible architecture, usually the standard documentation explains only the general approach, while forums provide more details.

The algorithm difficulty is not as optimal as it could be because of some programming language restrictions. Sometimes to make some data gathering from plenty of local tables the only possibility in the ABAP language is making double loop logic, while popular Python allows to make it just in one line. Anyway, in the business reality, this performance depends not only on the algorithm but on the capacity of the servers that the company has (in the case of on-premise solutions) or may rend (for Cloud solutions).

6.2 Contribution

The purposes of this study are to find out the basic and green criteria for the SS problem; review the main SS method and choose the suitable one for the implementation of the IS; describe the connection between chosen criteria and IS's storage. The work unites the reviewing and exploring of the existing literature with the real implementation process that could be used in practice. Here the requirements for the evaluated criteria were formulated; the basic criteria and the ones to evaluate suppliers' greenness were gathered in one model. Those parameters were described from the IS point of view which means corresponding data types and units of measure were provided. From the theory point of view, the main contribution is the definition of those criteria, that could be stored in an IS. The way how to represent non-numeric sub-criteria such as Environmental management, Green competencies, Innovations, and so on in IS was suggested. Corresponding units of measure were represented as well.

The basic SS methods were gathered and overviewed. The observation was made from the side of methods’ easiness for implementation and their steps’ intelligibility. Being based on the IS, this work uses three-level architecture to solve the SS problem, compared to other works. For instance, Amy H.I. Lee with colleagues [4] made a two-level hierarchy model for the SS problem. Lots of their suggested sub-criteria, like Green process planning, Capability of preventing pollution, Technology level, and so on are raw and should be divided into smaller parts to be evaluated. The current study divides sub-criteria into smaller parts that may be represented numerically and as a result stored in the IS and be taken into account during model creation.

The method PROMETEE II was chosen and described step by step with the green parameters example. Some steps were gathered into the one to improve the method’s performance. This method was implemented into the described three-level structure as its base to collect the raw data and return a score for the sub-criteria. This implementation is at the interface between science and technology. The well-known method’s logic was taken and successfully implemented into existing IS. This work may be considered as a creation supplier selection tool guide, that allows purchase managers to choose the most appropriate and perfect supplier by considering a range of the most important criteria.

This study discovers the basic IS business objects that are suitable to store corresponding data. Generally, it’s not a problem to store numeric data in the system, special custom tables may be created for this. However, such an approach is really difficult to support. So what is more challenging, is to find objects that are common for the purchasing process and are suitable to store necessary data. This work explains those relations, the Business Partner and Quotation are used. The explained objects may be a part of any IS, which means this structure and their relations with the criteria may be replicated in other IS.

In the current work, the IS specific from the SAP side was explained, but the disadvantages of this system were described as well. The implementation process was explained from scratch, starting with making necessary configurations and ending with programming architecture development and implementation. The architecture was developed for the high-level strong typed programming language, that allows replicating the logic to similar programming languages without serious charges. A flexible architecture was developed that makes supporting the supplier evaluation process easier. The flexibility was gained by the dynamic programming approaches and by creating custom tables and maintenance views. The tables and maintenance views are the parts of any modern IS, so they could be easily adapted to any IS as well.

The result of the developed architecture is more than 2200 lines of ABAP code, which allows gathering required data from the database, performing PROMETEE II logic, and gaining necessary model output in the form of the score.

7 Further research

7.1 Pros and Cons of evaluating a supplier for a single product

Supplier evaluation by one product may not provide a full picture of the supplier's capabilities or suitability for a particular business.

Such an approach may not be representative of the supplier's overall quality, reliability, or cost-effectiveness, but may discover some service quality. The supplier may have great competences in producing one particular product but may have some significant disadvantages and weaknesses in producing other products (type of products) or supplying services.

Generally, it is more relevant to evaluate a supplier based on the range of products or services they provide, taking into account their overall quality, reliability, cost-effectiveness, sustainability, customer service, greenness, and so on. This approach may provide a more comprehensive rating of the supplier's accordance with a particular business wishes and requests and help to be sure that the chosen supplier is able to meet the organization's needs over the long term.

But there are some cases when it could be acceptable to choose a supplier based on one specific product. There are a couple of examples below:

- Custom-specific products. Sometimes companies may need custom-specific, highly specialized, or niche products, that may be produced only by limited numbers of suppler. In that case, it’s possible to evaluate suppliers by their ability to produce the product.
- Trial purchases. If an organization is going to deal with a new supplier it’s possible to make a trial purchase to gather the overview about the vendor based on the trial delivery.
- Patented products. In that case, the required product may be produced by either one or some vendors. If there is only one producer on the market, then there is no possibility for choice at all. If there are some suppliers, then the main factor of choice is the suppliers’ capability to deliver that material at the minimum cost, shortest time, with minimum environmental damage, and so on.

7.2 Possible model improvement

First of all, there are several products that may be used to enlarge the model's relevance. Despite that the current model version allows to cover some business cases, making the model evaluate suppliers by the range of products or materials allows to test the model at larger number real cases. The improved model will cover current cases as well. Moreover, such a model may be interesting for companies that could produce productive data. The productive data will help to look deeply at the strengths and weaknesses of the algorithm and let understand whether the chosen method of SS suits true business reality.

Second of all, other approaches and models for supplier evaluation could be explored and implemented. The model described in this work is based on the PROMETEE II algorithm, which generally has exponential difficulty. That means, that the difficulty of calculations enlarges with the increasing numbers of initial data (suppliers for the evaluation) exponentially. It’s not a significant problem for businesses that choose between a couple of dozen suppliers but becomes a problem for choosing suppliers in a highly competitive market. Additionally, the difficulty of the algorithm will be getting higher by adding new products for evaluation. So it makes sense to consider other models of SS.

Third of all, the other IS could be used as the basis for the implementation. The current algorithm was written in ABAP language [20]. This is an internal high-level programming language in the SAP environment that is used to develop and enhance SAP functionality. This language requires the declaration of types of variables and structures, which makes dynamic programming, especially necessary when working with data, a bit difficult. So other ISs may be considered where the programming language is more flexible.

To improve the quality of the algorithm, it is necessary to conduct research to determine the optimal weights. Working on productive data will solve this problem. Knowing the optimal weights of criteria, sub-criteria, and parameters, the quality of the model will improve significantly. This will let evaluate vendors more accurately.

As mentioned in the restrictions, the data for the model is contained in one IS. It is possible to collect data from different ISs in further research. To do this it is necessary to integrate the current system and external ones. As an option, Ariba or any external CRM system could be considered for integration. Such work allows studying existing approaches for integration, such as API and EDI.

8 Reference list

[1] Ghiani, Gianpaolo, Gilbert Laporte, and Roberto Musmanno, Introduction to Logistics Systems Management, 2 ed., New York: John Wiley & Sons, Incorporated, 2013.

[2] Ghamari, Roya, Mohammad Mahdavi-Mazdeh, and Seyed Farid Ghannadpour, "Resilient and Sustainable Supplier Selection Via a New Framework: A Case Study from the Steel Industry.," *Environment, Development and Sustainability,* vol. 24, no. 8, pp. 10403-10441, 2022.

[3] Dutta, P., Jaikumar, B., & Arora, M. S., "Applications of data envelopment analysis in supplier selection between 2000 and 2020: A literature review.," *Annals of Operations Research,* vol. 315, no. 2, pp. 1399-1454, 2022.

[4] Amy H.I. Lee, He-Yau Kang, Chang-Fu Hsu, Hsiao-Chu Hung, "A green supplier selection model for high-tech industry," *Expert Systems with Applications,* vol. 36, no. 4, pp. 7917-7927, 2009.

[5] electricenergyonline.com, "Norway's strategy for developing a green, circular economy," *Electric energy online,* August 11, 2021.

[6] The Norvegian Agency for Public and Funancial Management, "Action plan to increase the proportion of green public procurements and green innovation," The Norvegian Agency for Public and Funancial Management, Oslo, September 9th, 2021.

[7] Ministry of Finance and Statistics Norway, "A budget for security and fair distribution," 06 10 2022. [Online]. Available: https://www.regjeringen.no/en/aktuelt/a-budget-for-security-and-fair-distribution/id2930559/.

[8] Gupta, Srikant, Prasenjit Chatterjee, Morteza Yazdani, and Ernesto DR Santibanez Gonzalez, "A Multi-Level Programming Model for Green Supplier Selection," *Management Decision,* vol. 59, no. 10, pp. 2496-2527, 2021.

[9] B. P. R. Kelly Rainer, Introduction to Information Systems, 9 ed., John Wiley & Sons, 2021.

[10] D. M. Baymou, "ERP systems in supply chain management," *INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IJOAR .ORG,* vol. 2, no. 3, pp. 1-21, March, 2014.

[11] enlyft.com, "Market Share of Enterprise Resource Planning (ERP) products," [Online]. Available: https://enlyft.com/tech/enterprise-resource-planning-erp.

[12] Eurostat, "Eurostat: Statistics Explained," 1 10 2021. [Online]. Available: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Glossary:Herfindahl_Hirschman_Index_(HHI).

[13] Lyes Benyoucef, Hongwei Ding, Xiaolan Xie, "Supplier Selection Problem : Selection Criteria and Methods," *INRIA Lorraine LORIA,* vol. 4, no. 4726, 02 2003.

[14] Behzad Masoomi, Iman Ghasemian Sahebi, Masood Fathi, Figen Yıldırım, Shahryar Ghorbani, "Strategic supplier selection for renewable energy supply chain under green capabilities (fuzzy BWM-WASPAS-COPRAS approach)," *Energy Strategy Reviews,* vol. 40, 2022.

[15] AGORA, "AGORA: company information," [Online]. Available: https://agorab2b.com/en/.

[16] SAP Ariba, "Strategic Sourcing Solutions and Software," [Online]. Available: https://www.ariba.com/solutions/solutions-overview/strategic-sourcing/sourcing-solutions.

[17] Microsoft, "Microsoft Excel," [Online]. Available: https://www.microsoft.com/en-us/microsoft-365/excel.

[18] C.-B. C. Y.-C. T. Chin-Tsai Lin, "An ERP model for supplier selection in electronics industry," *Expert Systems with Applications,* pp. 1760-1765, 2011.

[19] Mbiatem, Jane & Taghipour, Atour & Canel-Depitre, Beatrice., "Supplier Selection Approaches for Decision Makers," in *Materials of ICICM '18: Proceedings of the 8th International Conference on Information Communication and Management*, 2018.

[20] SAP ABAP, "ABAP Programming Language - Overview," [Online]. Available: https://help.sap.com/doc/abapdocu_752_index_htm/7.52/en-us/abenabap_overview.htm#:~:text=ABAP%20is%20a%20programming%20language,makes%20object%2Doriented%20programming%20possible.&text=ABAP%20is%20the%20programming%20interface,AS%20ABAP)%20in%20SAP%20NetW.

[21] SAP Enterprise Resource Planning, "Enterprise Resource Planning," SAP, [Online]. Available: https://www.sap.com/products/erp.html.

[22] IBM, "Boolean values," IBM, 19 01 2022. [Online]. Available: https://www.ibm.com/docs/en/ias?topic=list-boolean-values. [Accessed 04 2023].

[23] A. Latyshau, *Assignment of LOG904-101 22H Research Design,* Molde, 2022.

[24] A. Latyshau, *"Green" supplier selection problem solving in information systems on the SAP-basis, LOG904-101,* Molde, 2022.

[25] Tarma Software Research Ltd, "About Publish or Perish," Tarma Software Research Ltd, 1990-2016. [Online]. Available: https://harzing.com/pophelp/index.htm. [Accessed 10 2022].

[26] Nitin Sachdeva, Avinash K. Shrivastava, Ankur Chauhan, "Modeling supplier selection in the era of Industry 4.0," *Benchmarking: An International Journal,* vol. 28, no. 5, pp. 1809-1836, 2019.

[27] N. Agrawal, "Multi-criteria decision-making toward supplier selection: exploration of PROMETHEE II method," *Benchmarking,* pp. 2122-2146, 2022.

[28] Vahid Balali, Banafsheh Zahraie and Abbas Roozbahani., «A Comparison of AHP and PROMETHEE Family Decision Making Methods for Selection of Building Structural System,» *American Journal of Civil Engineering and Architecture,* т. 2, № 5, pp. 149-159, 2014.

[29] J. R. C. W. B. Charles A. Weber, "Vendor selection criteria and methods," *European Journal of Operational Research,* vol. 50, no. 1, pp. 2-18, 1991.

[30] A. Konys, "Green Supplier Selection Criteria: From a Literature Review to a Comprehensive Knowledge Base," *Sustainability,* vol. 11, no. 15, p. 4208, 2019.

[31] C. C. Pertsova, Ecological Economics Research Trends, New York: Nova Publishers, 2007, p. 366.

[32] Kumar, Amit & Jain, Vipul, "Supplier selection: A green approach with carbon footprint monitoring," in *Proceedings of 8th International Conference on Supply*

Chain Management and Information Systems (SCMIS 2010), Hong Kong, China, 2010.

[33] Bethany Field, Craig Simmons, *Product Carbon Footprinting for Beginners,* United Kindom: Department of business innovations and skills, 2020.

[34] D. A. M. &. P. J. Pandey, "Carbon footprint: current methods of estimation," *Environmental Monitoring and Assessment,* pp. 1573-2959, 01 07 2011.

[35] International Organization for Standardization, "www.iso.org," ISO, [Online]. Available: https://www.iso.org/home.html.

[36] International Organization for Standartization, *Environmental management systems — Requirements with guidance for use,* 2015.

[37] SAP Help Portal, "MM Vendor Evaluation," SAP, [Online]. Available: https://help.sap.com/docs/SAP_ERP/ec6cdb924d4549bebf1adc4b5cf87b19/ace1b853dcfcb44ce10000000a174cb4.html?locale=en-US&version=6.00.latest.

[38] The Investopedia Team, "www.investopedia.com," 22 03 2023. [Online]. Available: https://www.investopedia.com/terms/e/environmental-social-and-governance-esg-criteria.asp.

[39] Amir Albadvi, S. Kamal Chaharsooghi, Akbar Esfahanipour, "Decision making in stock trading: An application of PROMETHEE," *European Journal of Operational Research,* vol. 177, no. 2, pp. 673-683, 2007.

[40] J. Laing, M. Fort, G. Buckley, "Supplier Quality Requirements," Australasia Business Unit – West Operations, 2019.

[41] O. Rabo, "What is CO2e and how is it calculated?," Coolerfuture, 18 11 2020. [Online]. Available: https://www.coolerfuture.com/blog/co2e.

[42] Salman Bashir Memon, Amran Rasli, Abdul Samad Dahri, Imelda Hermilinda Abas, "Importance of Top Management Commitment to Organizational Citizenship Behaviour towards the Environment, Green Training and Environmental Performance in Pakistani Industries," *Sustainability,* vol. 14, no. 17, 2022.

[43] SAP: Sustainability Performance Management, "Sustainability Performance Management 4.0 Based on SAP NetWeaver," SAP, [Online]. Available: https://help.sap.com/docs/SAP_SUSTAINABILITY_PERFORMANCE_MANAGEMENT/f61cc730a56547d99616ffa2fee7add9/7ac3c9507a90e85ee10000000a44538d.html?locale=en-US.

[44] SAP: Environmental Compliance, "SAP Environmental Compliance," SAP, [Online]. Available: https://help.sap.com/docs/SAP_ENVIRONMENTAL_COMPLIANCE?locale=en-US.

[45] SAP Business Partners, "Master Data: Business Partners," SAP, [Online]. Available: https://help.sap.com/docs/SAP_S4HANA_FOR_CUSTOMER_MANAGEMENT/b90203d3616f482ebd9776775ac722d8/48b3d5a903d0356be10000000a421937.html?version=1.0&locale=en-US.

[46] SAP Purchasing (MM-PUR), "Purchasing (MM-PUR)," SAP, [Online]. Available: https://help.sap.com/docs/SAP_ERP/16ec4da603a84f06bd1f544112c95577/737eb65334e6b54ce10000000a174cb4.html?locale=de-DEversion%3D6.00.latest&version=6.03.latest.

[47] Sap Press, "Sap Press: Rheinwerk Publishing".

[48] Brian O'Neill, Jelena Perfiljeva, ABAP: an introduction, vol. 2, E-book format, 2020.

[49] SAP Blogs, "Blogs SAP: community," SAP, [Online]. Available: https://blogs.sap.com/.

[50] Tutorials Point, "SAP ABAP - Useful Resources," Tutorials Point, [Online]. Available: https://www.tutorialspoint.com/sap_abap/sap_abap_useful_resources.htm#.

[51] SAP Answers, "SAP Answers Community: All questions," SAP, [Online]. Available: https://answers.sap.com/index.html.

[52] ASTRAFOX, "ABAP Blog: everything about development," ASTRAFOX, [Online]. Available: https://abap-blog.ru/.

[53] OpenSap, "Evaluating suppliers with SAP S/4HANA," OpenSap, 10 May 2022. [Online]. Available: https://microlearning.opensap.com/media/How+to+Evaluate+Suppliers+with+SAP+S+4HANA/1_29acu4hn.

[54] SAP Wiki, "Customer Exits (CMOD)," SAP, 2010. [Online]. Available: https://wiki.scn.sap.com/wiki/pages/viewpage.action?pageId=107774353.

[55] SAP Wiki, "SPRO - SAP Project Reference Object," 2013. [Online]. Available: https://wiki.scn.sap.com/wiki/display/Basis/SPRO+-+SAP+Project+Reference+Object#:~:text=SPRO%20is%20abbreviation%20for%20SAP,customization%20settings%20for%20all%20modules..

[56] SAP: ABAP Programming (BC-ABA), "Function Modules," SAP, [Online]. Available: https://help.sap.com/docs/SAP_NETWEAVER_731_BW_ABAP/cfae740a0a21455dbe6e510c2d86e36a/9fdb988735c111d1829f0000e829fbfe.html?version=7.31.22&locale=en-US.

[57] SAP: Credit Management Configuration Guide, "Configuring Business Partner Additional Information," SAP, [Online]. Available: https://help.sap.com/docs/SAP_ERP/8e7ca28e0beb4e399aafb38ffce21c52/a64c195466432b18e10000000a174cb4.html?version=6.02.latest&locale=en-US.

[58] SAP: Sales and distribution, "Electronic Data Interchange / IDoc Interface (SD-EDI)," SAP, [Online]. Available: https://help.sap.com/docs/SAP_S4HANA_ON-PREMISE/7b24a64d9d0941bda1afa753263d9e39/3854b753128eb44ce10000000a174cb4.html.

[59] SAP Enhancements to the Standard, "Changing the SAP Standard (BC)," SAP, [Online]. Available: https://help.sap.com/docs/SAP_NETWEAVER_AS_ABAP_751_IP/2b28ffa716c24348903f8ffbfeb81df8/c819756543b111d1896f0000e8322d00.html.

[60] SAP Docs: Customer name range, "Customer name range for SAP ABAP or not ABAP objects," SAP , [Online]. Available: https://sap-docs.com/en/customer-name-range-for-sap-abap-or-not-abap-objects/.

[61] G. Serfiotis, "Adding customer (Z-) fields to Purchase Orders," SAP, 30 09 2015. [Online]. Available: https://blogs.sap.com/2015/09/30/adding-customer-z-fields-to-purchase-orders/.

[62] S&P, "S&P Global Rarings," S&P, [Online]. Available: https://www.spglobal.com/ratings/en/index.

[63] SAP Insights, "What is master data management (MDM)," [Online]. Available: https://www.sap.com/insights/what-is-master-data-management-mdm.html.

[64] stechies.com, "Master Data and Transactional Data," [Online]. Available: https://www.stechies.com/master-data-transactional-data/#:~:text=This%20data%20cannot%20be%20changed,journal%20entry%20(FI)%20etc..

[65] I. Object Management Group, "Business Process Model And Notation," 01 12 2010. [Online]. Available: https://www.omg.org/spec/BPMN/2.0/.

[66] SAP Business Objects, "SAP Business Objects Business Intelligence suite," SAP, [Online]. Available: https://www.sap.com/products/technology-platform/bi-platform.html#:~:text=What%20is%20SAP%20BusinessObjects%20Business,insights%2C%20available%20anytime%2C%20anywhere..

[67] 1C, "1C COMPANY," 1C, [Online]. Available: https://1c.ru/eng/title.htm.

9 Attachments

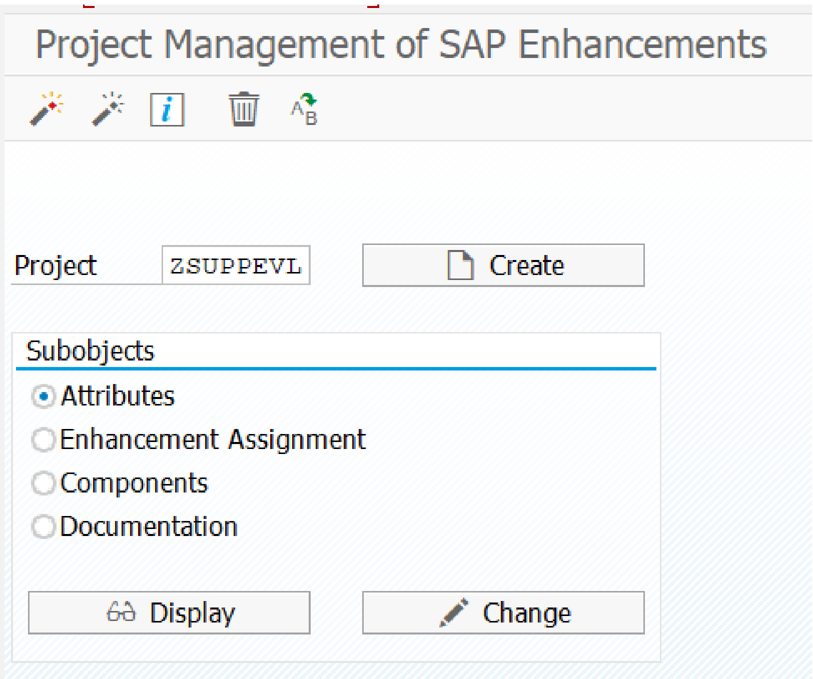
Attachment 1: ESG evaluation for supplier risk evaluation (example from the real business)

ESG evaluation for supplier risk evaluation
General
How does the supplier involve employees (non-management) in decision-making processes?
Does the supplier have an established system for managing grievances from employees?
If yes, please describe briefly?
If yes, how is this communicated to all employees?
Does the supplier have a Code of Conduct, or other set of criteria, that it sends to its own suppliers?
Environment
Does the supplier employ a dedicated person to ensure compliance with environmental laws and regulations?
Is the supplier certified to ISO 14001 or similar?
Is the supplier certified to ISO 14064 (GHG Emissions]?
Does the supplier set goals to reduce emissions?
Is the supplier willing to share emissions data with Freyr?
How does the supplier ensure they are always in compliance with applicable environmental regulations?
Does the supplier monitor effluence (waste) to land/water?
Does the supplier have a system in place to monitor and control the flow of hazardous substances?
Do staff receive training on how to store, handle and dispose of kazardous waste?
Does the supplier set goals for:
a) Energy efficiency
b) Reductions in freshwater
c) Waste reduction
d) Recovering and/or recycling material waste flows?
Does the supplier have available an LCA for the products they deliver to Freyr/24M?
If so, is this in accordance with ISO 14044?
Has the supplier done an assessment of the impact of their operations on biodiversity?
Would you like to add more information about the supplier's commitment to reducing the environmental impact of their operations?
Could be added
Carbon Trust Product Footprint Certification
Health, Safety and Wellbeing
Does the supplier have a HSE Management System certified to 150 45001/OHSAS 18001 or similar?
When did the supplier conduct the last HSE risk assessment for their operations?
Does the supplier have a worker health and safety committee, or similar?
Does the supplier have a system for reporting health and safety Incidents?
If yes, how is this communicated to all employees?
Human and labor rights
Does the supplier have a policy stating their commitment to respecting human rignts?
Does the supplier have routines in place to identify, assess and mitigate the risk of breach to human rights in their 39 operations?
Does the supplier have routines/procedures in place to engage h local stakeholders to avoid negative impacts from their operations?
Does the supplier use a contractor to provide security services for the facility/facilities producing goods for 24M?
Non-discrimination
Does the supplier have a policy stating their commitment to non-discrimination (this can be part of another policy or a separate policy)?
Modern Slavery and Child labor
Does the supplier have a policy stating their commitment to avoiding forced or bonded labour, including child labour (this can be part of another policy or a separate policy)?
Does the supplier have routines/procedures in place to identify, assess and mitigate the risk of modern slavery, including child labour', in their operations?
Does the supplier have routines/procedures in place to protect young workers (under 18)?
Freedom of association & Collective bargaining
Does the supplier have a policy stating their commitment to respecting the right to organise and bargain collectively(this can be part of another policy or a separate policy)?
Are employees' working conditions covered by a collective agreement?
Wages & Working Hours
Does the supplier have a policy stating their commitment to respecting local laws regarding working hours (this can be part of another policy or…)

Attachment 2: Summary of MCDM technique (Table 1 of [27])

Method	Author	Advantage	Disadvantage
VIKOR	Yu (1973)	The best alternative is preferred by maximizing the utility group and minimizing the regret group	Crisp data are inadequate to model real-life situations;
Analytic hierarchy process (AHP)	Saaty (1986), Liu et al (2018)	Calculation of pairwise comparison and weights easy; more flexibility	lack of flexibility
Analytic network process (ANP)	Saaty (1996), Abdollahi et al (2015), Liu et al. (2018)	Deal quantitatively with the dependencies and interactions across the elements at various levels	Does not perform accurately during interactions and dependencies across the criteria
PROMETHEE	Velasquez and Hester (2013)	Streamline human perception; easy to use; does not require the assumption that criteria are proportionate	Work inefficiently when the number of alternatives increases; lower ability for capturing complexity Does not provide a clear method by which to assign weights
DEMATEL	Abdollahi et al. (2015)	Determining the accurate interdependencies between the proposed criteria	Determines the ranking of alternatives based on interdependent relationships among them
Data envelopment analvsis (DEA)	Velasquez and Hester (2013)	Capable to handle multiple inputs and outputs; efficiency can be analyzed and quantified	Not deal with imprecise data and assumes that all input and output data are exactly known Difficult to explain the outcome in simple term
ELECTREE	Velasquez and Hester (2013)	Takes uncertainty and vagueness into account	Not able to handle complex situation and real time problems
Best worst method (BWM)	Rezaei et al. (2016)	More Structure, requires fewer data, reliable result	Highly dependent on expert's judgment and result
Interpretive Structural modeling (ISM)	Agrawal (2019)	Captures the complexities of real life problems	may vary
TOPSIS	Velasquez and Hester (2013)	Easy to use and have simple process; number of steps remains the same regardless of the number of attributes	Difficult to weight and keep consistency of judgment
Simple additive weighting (SAW)	Velasquez and Hester (2013), Sen et al. (2015)	Do not need complex computer code	Result may not be logical
Aggregated indices randomization method (AIRM)	Hovanov et al. (2008), Sen et al. (2015)	Can use ordinal and interval data and non-complete expert opinion	Aggregated index value may vary
Weighted sum model (WSM)	Goh et al. (1996), Sen et al (2015)	Elimination of extreme value	If the extreme value were not eliminated the rank reversal occurs

Attachment 3: Custom project in SAP (CMOD transaction)

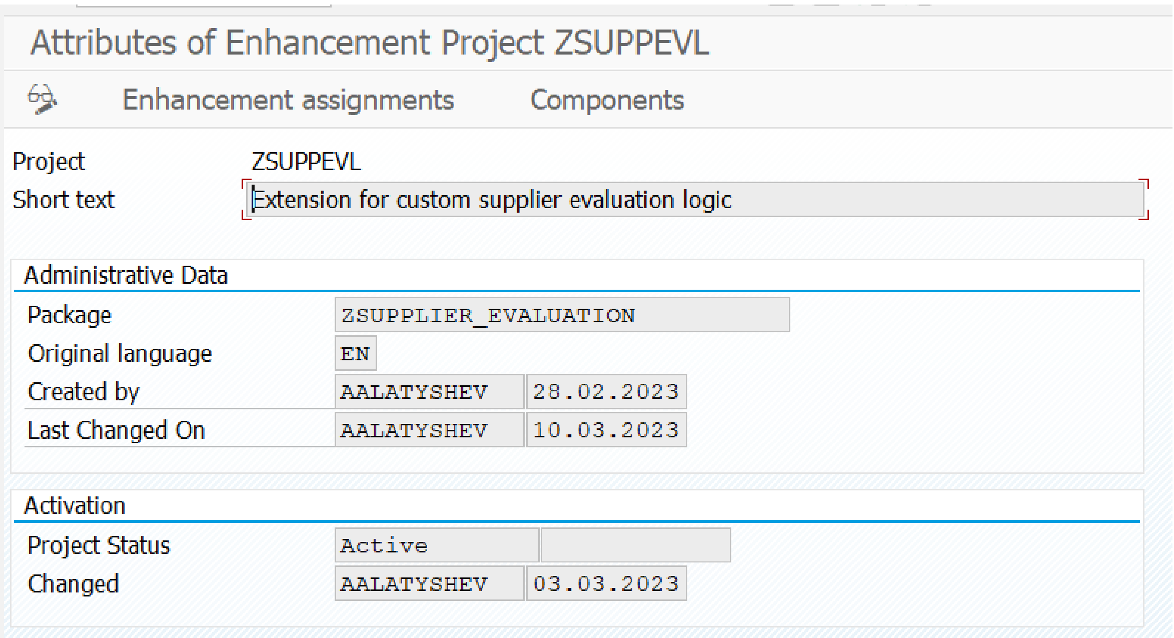


Project Management of SAP Enhancements

Project ZSUPPEVL

Subobjects

● Attributes
○ Enhancement Assignment
○ Components
○ Documentation



Attributes of Enhancement Project ZSUPPEVL

Project ZSUPPEVL

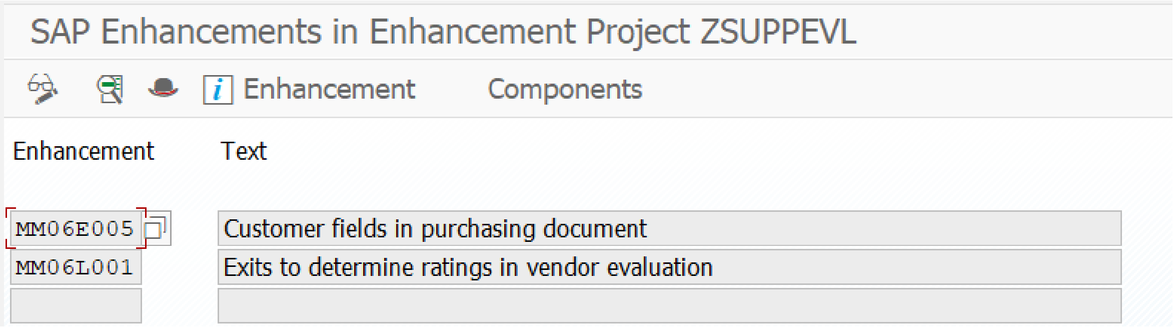
Short text Extension for custom supplier evaluation logic

Administrative Data

Package	ZSUPPLIER_EVALUATION	
Original language	EN	
Created by	AALATYSHEV	28.02.2023
Last Changed On	AALATYSHEV	10.03.2023

Activation

Project Status	Active	
Changed	AALATYSHEV	03.03.2023

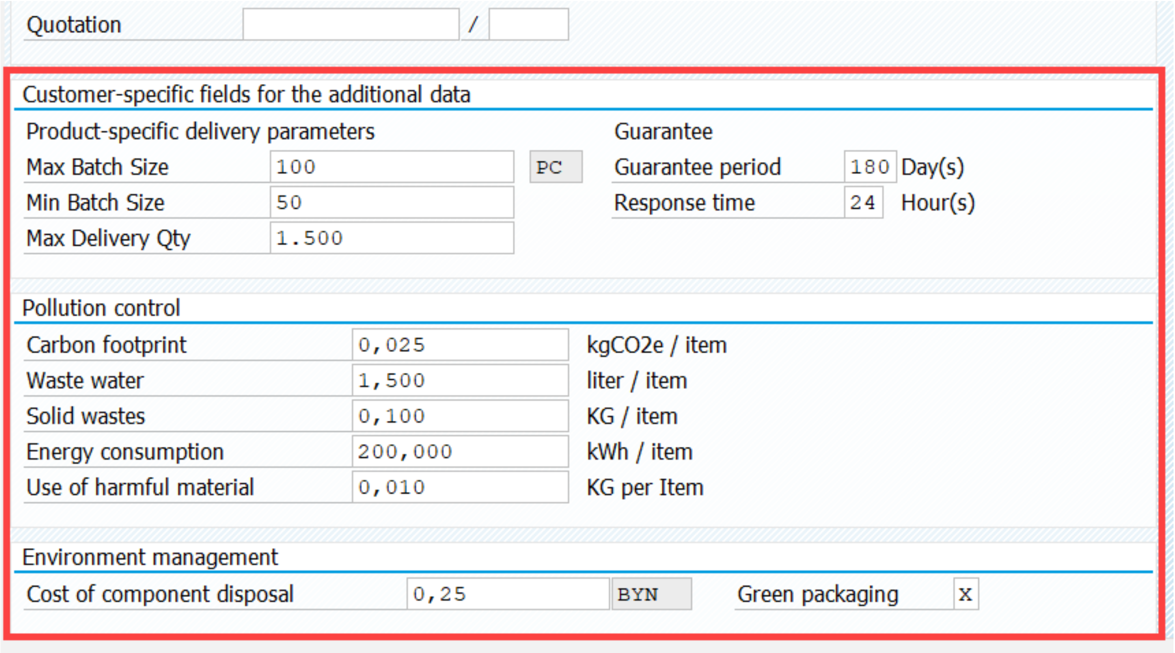


SAP Enhancements in Enhancement Project ZSUPPEVL

Enhancement	Text
MM06E005	Customer fields in purchasing document
MM06L001	Exits to determine ratings in vendor evaluation

Attachment 4: Quotation fields' types (EKPO table) and the quotation screen with new fields

Field	Data Element	Data type	Length	Decimals	Description
ZZ_CHARG_MAX_SIZE	ZDE_CHARG_MAX_SIZE	QUAN	13	3	Supplier's maximum lot size
ZZ_CHARG_MIN_SIZE	ZDE_CHARG_MIN_SIZE	QUAN	13	3	Supplier's minimum lot size
ZZ_CHARG_SPEC_SIZE	ZDE_CHARG_SPEC_SIZE	QUAN	13	3	Supplier's specific lot size
ZZ_EPLIF_GARANTEE	ZDE_EPLIF_GARANTEE	DEC	3	0	Guarantee period (days)
ZZ_RESPONSE_TIME	ZDE_RESPONSE_TIME	DEC	2	0	Response time to technical inquiries or problems (hours)
ZZ_CARBON_FOOTPRINT	ZDE_CARBON_FOOTPRINT	DEC	11	3	Carbon footprint (kgCO2 / item)
ZZ_WASTE_WATER	ZDE_WASTE_WATER	DEC	11	3	Waste water (liter / item)
ZZ_SOLID_WASTES	ZDE_SOLID_WASTES	DEC	11	3	Solid non-recyclable wastes (KG / item)
ZZ_ENERGY_CONSUMP	ZDE_ENERGY_CONSUMP	DEC	11	3	Energy consumption (kWh / item)
ZZ_HARMFUL_MAT	ZDE_HARMFUL_MAT	DEC	11	3	Use of harmful / hazard material (kg / item)
ZZ_GREEN_PACK	ZDE_GREEN_PACK	CHAR	1	0	Green packaging (yes / no)
ZZ_COST_COMP_DISPOS	ZDE_COST_COMP_DISPOS	CURR	11	2	Cost of component disposal (price per item)


Quotation /

Customer-specific fields for the additional data

Product-specific delivery parameters

Max Batch Size 100 PC

Min Batch Size 50

Max Delivery Qty 1.500

Guarantee

Guarantee period 180 Day(s)

Response time 24 Hour(s)

Pollution control

Carbon footprint 0,025 kgCO2e / item

Waste water 1,500 liter / item

Solid wastes 0,100 KG / item

Energy consumption 200,000 kWh / item

Use of harmful material 0,010 KG per Item

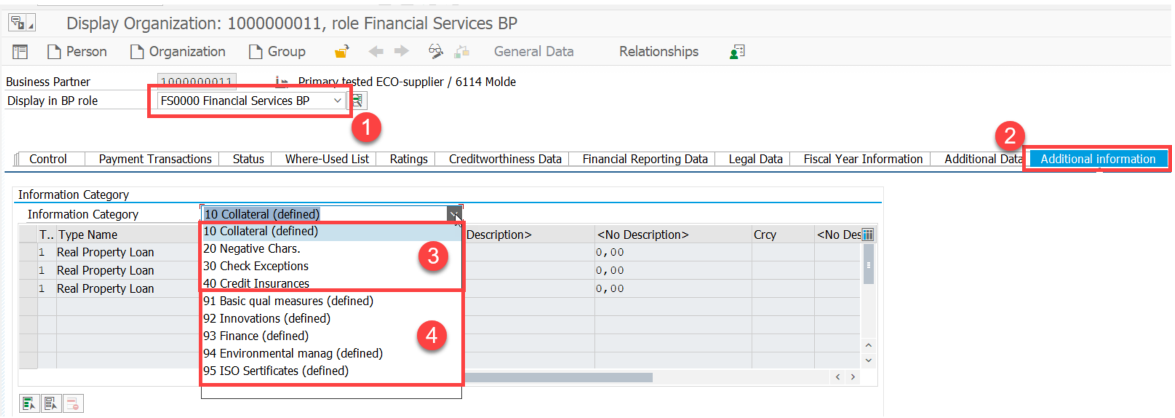
Environment management

Cost of component disposal 0,25 BYN

Green packaging X

Figure 35: Custom specific fields in the Quotation object

Attachment 5: Business Partner Additional Information view (transaction BP, role FS0000)



Display Organization: 1000000011, role Financial Services BP

Business Partner 1000000011 Primary tested ECO-supplier / 6114 Molde

Display in BP role FS0000 Financial Services BP

Information Category

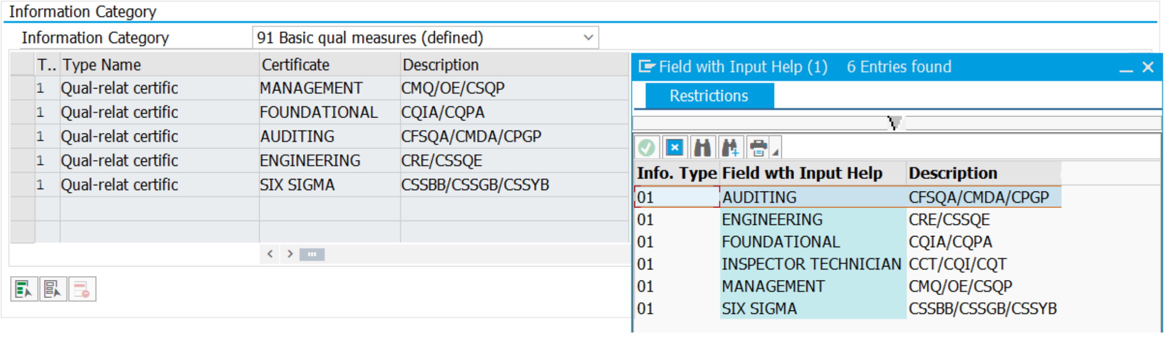
Information Category 10 Collateral (defined)

10 Collateral (defined)
20 Negative Chars.
30 Check Exceptions
40 Credit Insurances
91 Basic qual measures (defined)
92 Innovations (defined)
93 Finance (defined)
94 Environmental manag (defined)
95 ISO Sertificates (defined)

T..	Type Name	Description>	<No Description>	Crcy	<No Des
1	Real Property Loan		0,00		
1	Real Property Loan		0,00		
1	Real Property Loan		0,00		

1 – BP role
2 – Additional information view
3 – Standard categories
4 – Custom categories that are configured for the current work

91 “**Basic quality measures**”. One information type 01 “Quality related certificate”. Field type “Input from the help-list”. The help list is predefined with the main quality-related types of certificates.



Information Category

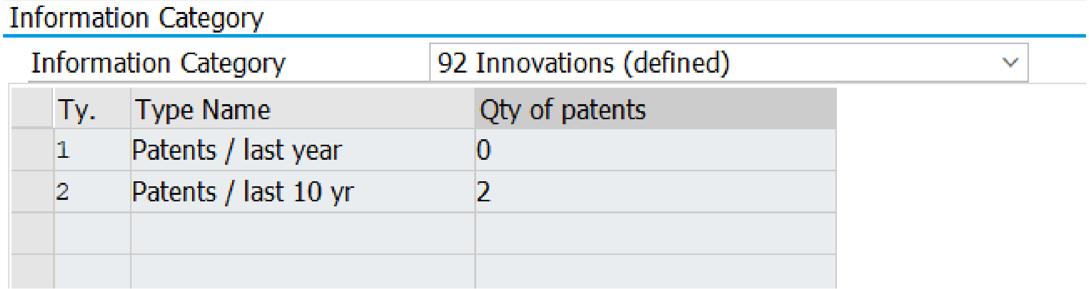
Information Category 91 Basic qual measures (defined)

T..	Type Name	Certificate	Description
1	Qual-relat certific	MANAGEMENT	CMQ/OE/CSQP
1	Qual-relat certific	FOUNDATIONAL	CQIA/CQPA
1	Qual-relat certific	AUDITING	CFSQA/CMDA/CPGP
1	Qual-relat certific	ENGINEERING	CRE/CSSQE
1	Qual-relat certific	SIX SIGMA	CSSBB/CSSGB/CSSYB

Field with Input Help (1) 6 Entries found

Info. Type	Field wth Input Help	Description
01	AUDITING	CFSQA/CMDA/CPGP
01	ENGINEERING	CRE/CSSQE
01	FOUNDATIONAL	CQIA/CQPA
01	INSPECTOR TECHNICIAN	CCT/CQI/CQT
01	MANAGEMENT	CMQ/OE/CSQP
01	SIX SIGMA	CSSBB/CSSGB/CSSYB

92 “**Innovations**”. Two information types: 01 “The number of patents for the last year”, 02 “The number of patents for the last 10 years”. Field type “Character”.



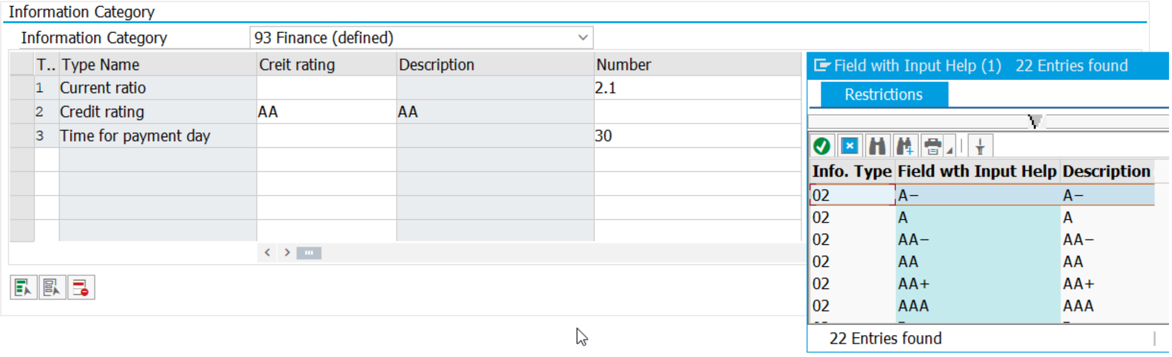
Information Category

Information Category 92 Innovations (defined)

Ty.	Type Name	Qty of patents
1	Patents / last year	0
2	Patents / last 10 yr	2

93 “**Finance**”. Three information types:

- 01 “Current ratio”, field type “Character”
- 02 “Credit rating”, field type “Input from the help-list”, help list is predefined with the S&P range of the credit rating [62]
- 03 “Time for payment (days)”, field type “Character”



Information Category

Information Category: 93 Finance (defined)

T..	Type Name	Creit rating	Description	Number
1	Current ratio			2.1
2	Credit rating	AA	AA	
3	Time for payment day			30

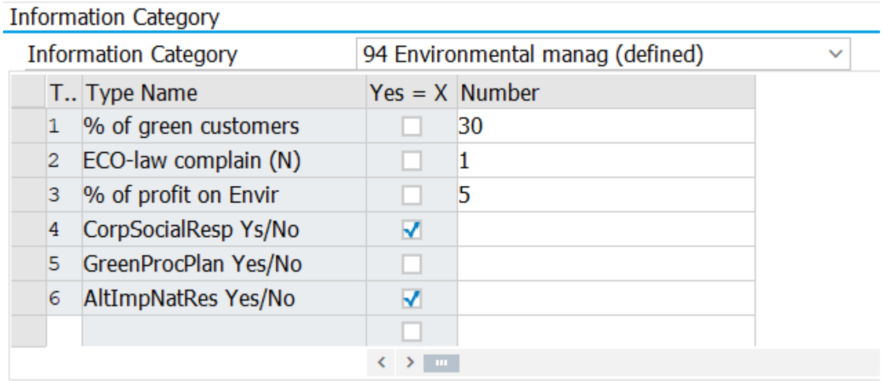
Field with Input Help (1) 22 Entries found

Info. Type	Field wth Input Help	Description
02	A–	A–
02	A	A
02	AA–	AA–
02	AA	AA
02	AA+	AA+
02	AAA	AAA

22 Entries found

94 “Environmental management”. Six information types:

- 01 “Percentage of green customers”, field type “Character”
- 02 “ECO-law complaints (N) for the last year”, field type “Character”
- 03 “Percentage of profit that managers spend to support Environment”, field type “Character”
- 04 “Corporate Social Responsibility (Yes/No)”, field type “Boolean”
- 05 “Green Process Planning (Yes/No)”, field type “Boolean”
- 06 “Ability to alter process and product for reducing the impact on natural resources (Yes/No)”, field type “Boolean”

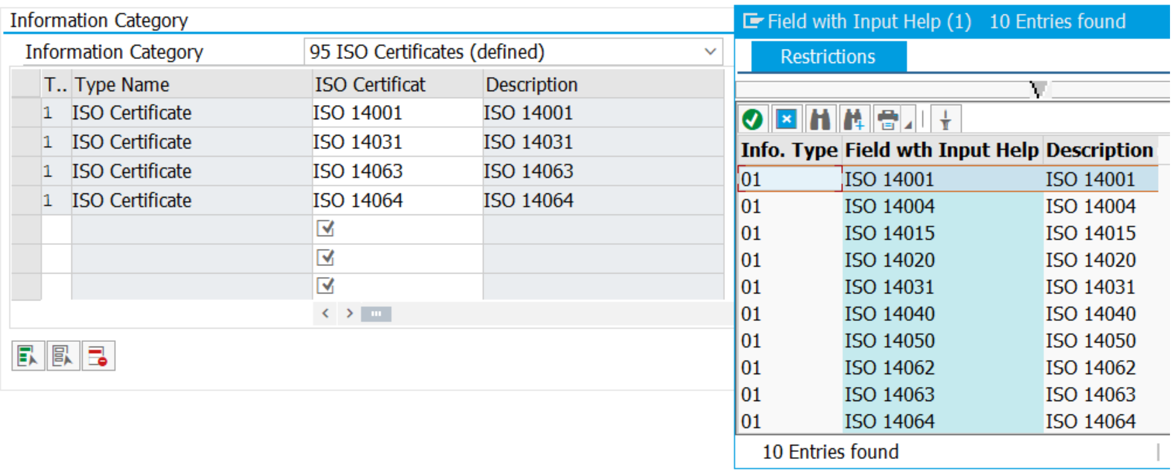


Information Category

Information Category: 94 Environmental manag (defined)

T..	Type Name	Yes = X	Number
1	% of green customers	□	30
2	ECO-law complain (N)	□	1
3	% of profit on Envir	□	5
4	CorpSocialResp Ys/No	☑	
5	GreenProcPlan Yes/No	□	
6	AltImpNatRes Yes/No	☑	
		□	

95 “ISO Certificates”. One information type 01 “ISO Certificate”. Field type “Input from the help-list”. Help list is predefined with the main ISO certificates [35].



Information Category

Information Category: 95 ISO Certificates (defined)

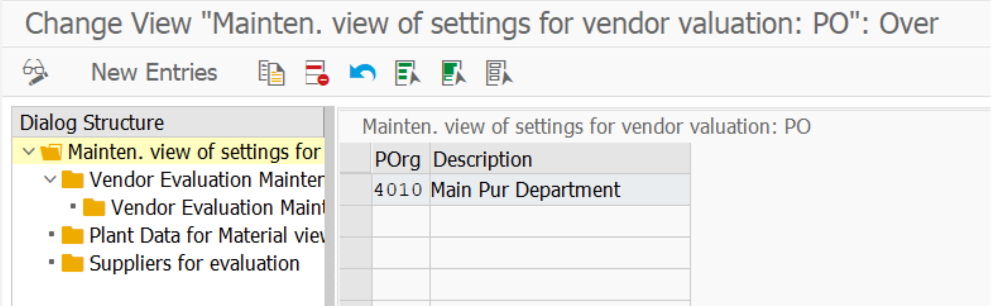
T..	Type Name	ISO Certificat	Description
1	ISO Certificate	ISO 14001	ISO 14001
1	ISO Certificate	ISO 14031	ISO 14031
1	ISO Certificate	ISO 14063	ISO 14063
1	ISO Certificate	ISO 14064	ISO 14064
		☑	
		☑	
		☑	

Field with Input Help (1) 10 Entries found

Info. Type	Field wth Input Help	Description
01	ISO 14001	ISO 14001
01	ISO 14004	ISO 14004
01	ISO 14015	ISO 14015
01	ISO 14020	ISO 14020
01	ISO 14031	ISO 14031
01	ISO 14040	ISO 14040
01	ISO 14050	ISO 14050
01	ISO 14062	ISO 14062
01	ISO 14063	ISO 14063
01	ISO 14064	ISO 14064

10 Entries found

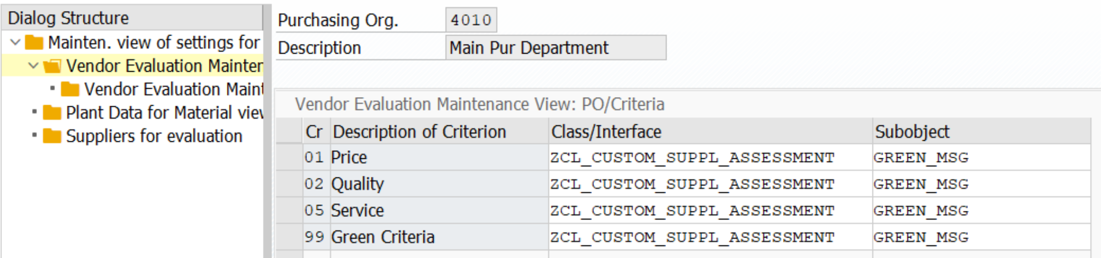
Attachment 6: View cluster for vendor evaluation settings


Change View "Mainten. view of settings for vendor valuation: PO": Over
New Entries
Dialog Structure
Mainten. view of settings for
Vendor Evaluation Mainter
Vendor Evaluation Maint
Plant Data for Material viev
Suppliers for evaluation

Mainten. view of settings for vendor valuation: PO

POrg	Description
4010	Main Pur Department

Figure 36: Maintain view of settings for vendor valuation: PO

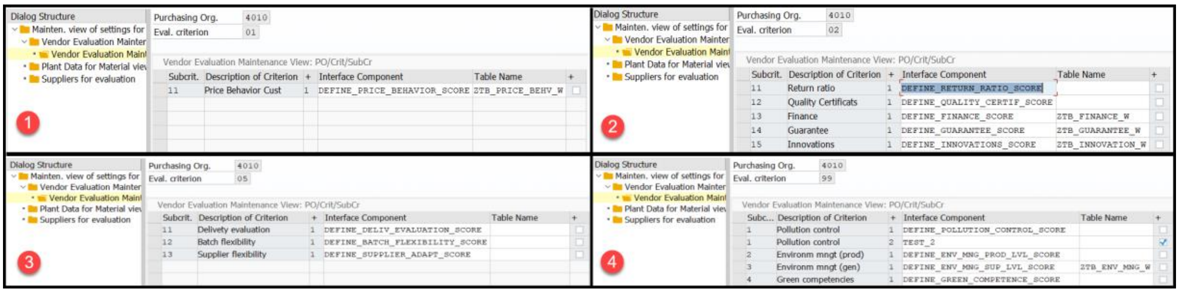

Dialog Structure
Mainten. view of settings for
Vendor Evaluation Mainter
Vendor Evaluation Maint
Plant Data for Material viev
Suppliers for evaluation

Purchasing Org. 4010
Description Main Pur Department

Vendor Evaluation Maintenance View: PO/Criteria

Cr	Description of Criterion	Class/Interface	Subobject
01	Price	ZCL_CUSTOM_SUPPL_ASSESSMENT	GREEN_MSG
02	Quality	ZCL_CUSTOM_SUPPL_ASSESSMENT	GREEN_MSG
05	Service	ZCL_CUSTOM_SUPPL_ASSESSMENT	GREEN_MSG
99	Green Criteria	ZCL_CUSTOM_SUPPL_ASSESSMENT	GREEN_MSG

Figure 37: Vendor Evaluation Maintenance View: PO/Criteria


Dialog Structure
Mainten. view of settings for
Vendor Evaluation Mainter
Vendor Evaluation Maint
Plant Data for Material viev
Suppliers for evaluation
1
2
3
4

1

Purchasing Org. 4010
Eval. criterion 01

Vendor Evaluation Maintenance View: PO/Crit/SubCr

Subcrit.	Description of Criterion	+	Interface Component	Table Name	+
11	Price Behavior Cust	1	DEFINE_PRICE_BEHAVIOR_SCORE	ZTB_PRICE_BEHV_W	☐

2

Purchasing Org. 4010
Eval. criterion 02

Vendor Evaluation Maintenance View: PO/Crit/SubCr

Subcrit.	Description of Criterion	+	Interface Component	Table Name	+
11	Return ratio	1	DEFINE_RETURN_RATIO_SCORE		☐
12	Quality Certificats	1	DEFINE_QUALITY_CERTIF_SCORE		☐
13	Finance	1	DEFINE_FINANCE_SCORE	ZTB_FINANCE_W	☐
14	Guarantee	1	DEFINE_GUARANTEE_SCORE	ZTB_GUARANTEE_W	☐
15	Innovations	1	DEFINE_INNOVATIONS_SCORE	ZTB_INNOVATION_W	☐

3

Purchasing Org. 4010
Eval. criterion 05

Vendor Evaluation Maintenance View: PO/Crit/SubCr

Subcrit.	Description of Criterion	+	Interface Component	Table Name	+
11	Delivety evaluation	1	DEFINE_DELIV_EVALUATION_SCORE		☐
12	Batch flexibility	1	DEFINE_BATCH_FLEXIBILITY_SCORE		☐
13	Supplier flexibility	1	DEFINE_SUPPLIER_ADAPT_SCORE		☐

4

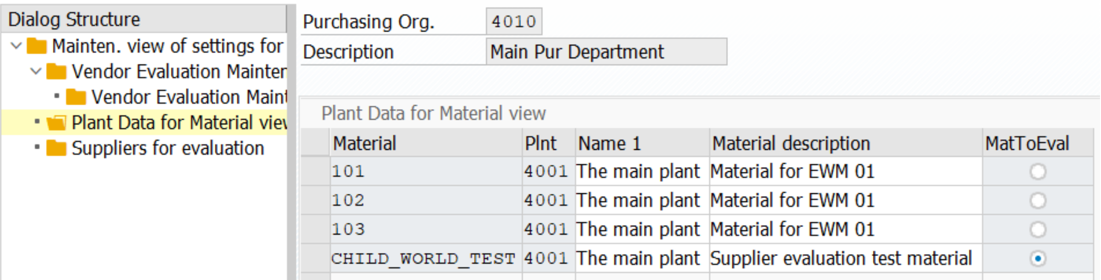
Purchasing Org. 4010
Eval. criterion 99

Vendor Evaluation Maintenance View: PO/Crit/SubCr

Subc...	Description of Criterion	+	Interface Component	Table Name	+
1	Pollution control	1	DEFINE_POLLUTION_CONTROL_SCORE		☐
1	Pollution control	2	TEST_2		☑
2	Environm mngt (prod)	1	DEFINE_ENV_MNG_PROD_LVL_SCORE		☐
3	Environm mngt (gen)	1	DEFINE_ENV_MNG_SUP_LVL_SCORE	ZTB_ENV_MNG_W	☐
4	Green competencies	1	DEFINE_GREEN_COMPETENCE_SCORE		☐

Figure 38: Vendor Evaluation Maintenance View: PO/Crit/SubCriteria.

1 – Price, 2 – Quality, 3 – Service, 4 – Green criteria

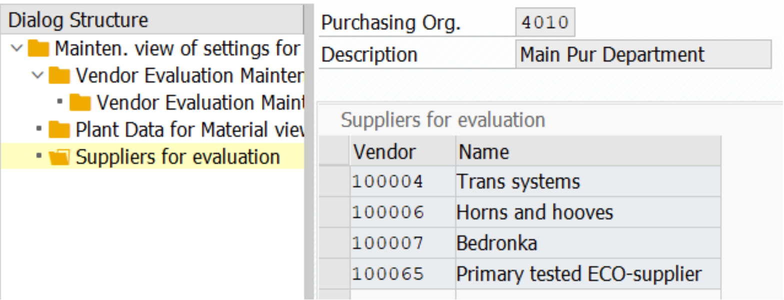

Dialog Structure
Mainten. view of settings for
Vendor Evaluation Mainter
Vendor Evaluation Maint
Plant Data for Material viev
Suppliers for evaluation

Purchasing Org. 4010
Description Main Pur Department

Plant Data for Material view

Material	Plnt	Name 1	Material description	MatToEval
101	4001	The main plant	Material for EWM 01	○
102	4001	The main plant	Material for EWM 01	○
103	4001	The main plant	Material for EWM 01	○
CHILD_WORLD_TEST	4001	The main plant	Supplier evaluation test material	●

Figure 39: Plant Data for Material view


Dialog Structure
Mainten. view of settings for
Vendor Evaluation Mainten
Vendor Evaluation Maint
Plant Data for Material view
Suppliers for evaluation

Purchasing Org.	4010
Description	Main Pur Department

Suppliers for evaluation

Vendor	Name
100004	Trans systems
100006	Horns and hooves
100007	Bedronka
100065	Primary tested ECO-supplier

Figure 40: Suppliers for evaluation

Attachment 7: The case study input data example (10 suppliers)

Table 16: Input suppliers' data in the Business Partner object

			91 Quality Cert						92 Innovations		93 Finance			94 Environmental management						95 ISO Certificates									
Supplier	**Object**	**Currency**	**AUDITING**	**ENGINEERING**	**FOUNDATIONAL**	**INSPECTOR TECHNICIA**	**MANAGEMENT**	**SIX SIGMA**	**Patents / last year**	**Patents / last 10 years**	**Current ratio**	**Credit rating**	**Time for payment day**	**% of green customers**	**ECO-law complain (N)**	**% of profit on Environment**	**Corporate Social Responsibility Ys/No**	**Green Procurement**	**Alter Impact on Natural**	**ISO 14001**	**ISO 14004**	**ISO 14015**	**ISO 14020**	**ISO 14031**	**ISO 14040**	**ISO 14050**	**ISO 14062**	**ISO 14063**	**ISO 14064**
100065	BP	EUR	1	1	1	0	1	1	0	2	2,1	AA	30	30	1	2,3	1	0	0	1	0	0	0	1	0	0	0	1	1
100070	BP	EUR	1	0	1	0	0	1	1	3	2,5	AA-	30	1	2	0,5	1	1	0	1	1	0	0	0	1	0	1	0	0
100071	BP	EUR	0	1	1	1	1	0	0	2	2,9	AAA	60	10	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1
100072	BP	EUR	0	0	0	0	1	0	0	0	3	AAA	45	0	1	0,8	0	1	1	1	0	0	0	0	0	0	0	0	0
100073	BP	EUR	1	0	0	1	0	0	1	1	1	BBB	35	7	0	1,2	1	0	1	0	1	1	0	0	0	0	0	0	0
100074	BP	EUR	0	1	1	0	0	0	2	2	1,3	B-	28	2	2	1,1	1	0	0	1	1	1	1	1	1	0	0	0	0
100075	BP	EUR	1	1	1	1	1	0	0	3	1,9	AA	30	12	0	0,7	0	1	0	0	0	0	0	0	0	0	1	1	1
100076	BP	EUR	0	0	0	0	1	1	0	0	2,7	BBB+	40	15	1	1,5	1	1	1	1	1	1	1	1	1	1	0	0	0
100077	BP	EUR	0	0	1	1	1	0	1	2	2	A-	42	28	3	3	0	1	1	0	0	1	1	1	1	1	0	0	0
100078	BP	EUR	1	1	0	0	1	1	0	1	1,4	BBB-	39	21	4	0,2	1	0	0	0	0	0	0	0	1	1	1	1	1

Table 17: Input suppliers' data in the Quotation object

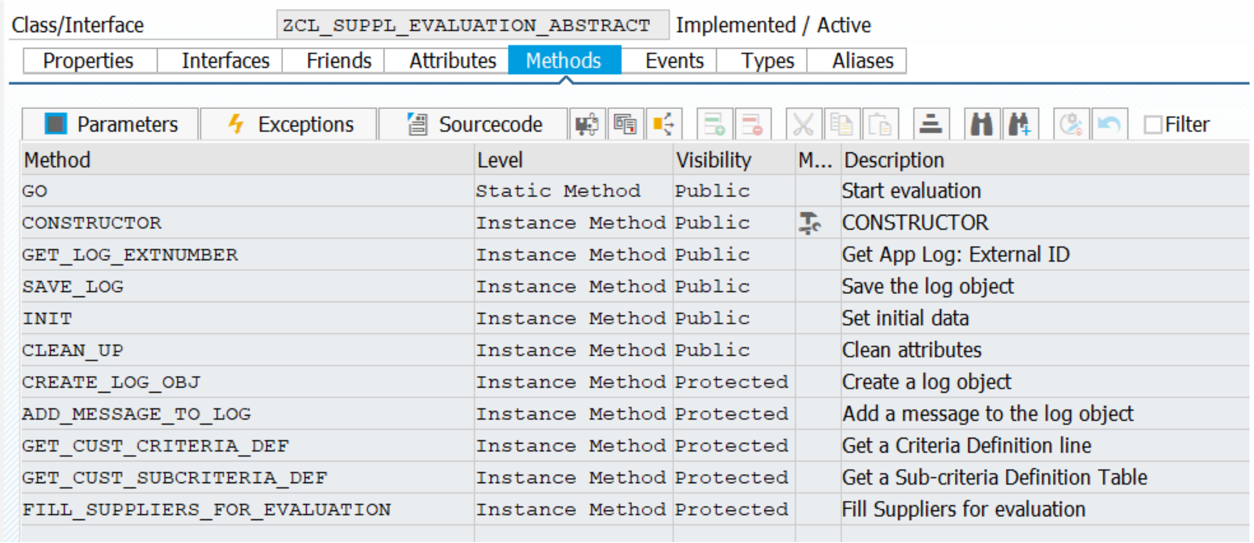
			Standard			Product-specific delivery parameters					Pollution control					Environmental Management	
Supplier	**Object**	**Quotation No**	**Price**	**Quality delivered on time**	**Lead time**	**Max Batch Size**	**Min Batch Size**	**Max Delivery quality**	**Guarantee period**	**Response time**	**Carbon footprint**	**Waste water**	**Solid wastes**	**Energy consumption**	**Use of harmful material**	**Cost of component disposal**	**Green packaging**
100065	Qt	6000000014	30,08	25%	15	150	25	2500	185	36	0,023	1,55	0,099	213	0,006	0,2	0
100070	Qt	6000000018	29,00	50%	20	200	50	5000	185	48	0,03	1,4	0,1	215	0,005	0,21	1
100071	Qt	6000000019	31,00	70%	14	500	250	10000	150	24	0,032	1,5	0,11	208	0,006	0,18	0
100072	Qt	6000000020	33,00	100%	14	500	100	10000	180	30	0,03	1,3	0,13	220	0,004	0,2	0
100073	Qt	6000000021	32,50	30%	17	350	100	5000	165	12	0,04	1,34	0,09	215	0,002	0,19	1
100074	Qt	6000000022	30,25	10%	20	100	100	3000	180	36	0,019	1,39	0,1	203	0,008	0,23	0
100075	Qt	6000000023	28,50	100%	15	1000	1000	15000	110	48	0,035	1,42	0,12	201	0,007	0,22	0
100076	Qt	6000000024	29,08	55%	22	750	100	7500	155	40	0,038	1,38	0,095	220	0,006	0,15	0
100077	Qt	6000000025	30,00	100%	25	250	200	12000	170	24	0,035	1,51	0,098	218	0,005	0,19	1
100078	Qt	6000000026	31,02	60%	15	300	125	9000	125	28	0,021	1,33	0,2	201	0,004	0,2	1

Attachment 8: The customer exit code

```
*"*"Local Interface:
*"  IMPORTING
*"     VALUE(XLIFNR) LIKE  ELBK-LIFNR
*"     VALUE(XEKORG) LIKE  ELBK-EKORG
*"     VALUE(XHKRIT) LIKE  ELBP-HKRIT
*"     VALUE(XTKRIT) LIKE  T147C-TKRIT
*"     VALUE(XKRTYP) LIKE  T147I-KRTYP
*"  EXPORTING
*"     VALUE(XBEURT) LIKE  ELBP-BEURT
*"----------------------------------------------------------------------

zcl_suppl_evaluation_abstract=>go(
  EXPORTING
    iv_xlifnr     = xlifnr
    iv_xekorg     = xekorg
    iv_xhkrit     = xhkrit
    iv_xtkrit     = xtkrit
    iv_xkrtyp     = xkrtyp
  IMPORTING
    ev_severity   = DATA(lv_severity)
    et_bapiret    = DATA(lt_bapiret)
    ev_xbeurt     = xbeurt
  EXCEPTIONS
    error         = 1
    OTHERS        = 2 ).

IF sy-subrc <> 0 OR lv_severity CA 'EA'.
  READ TABLE lt_bapiret INTO DATA(ls_bapiret) WITH KEY type = 'E'.
  IF sy-subrc EQ 0.
    MESSAGE ID ls_bapiret-id TYPE ls_bapiret-type NUMBER ls_bapiret-number
            WITH ls_bapiret-message_v1 ls_bapiret-message_v2 ls_bapiret-message_v3 ls_bapiret-message_v4.
  ELSE.
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
  ENDIF.
ENDIF.
```

Attachment 9: The calling class ZCL_SUPPL_EVALUATION_ABSTRACT.

Architecture and code.



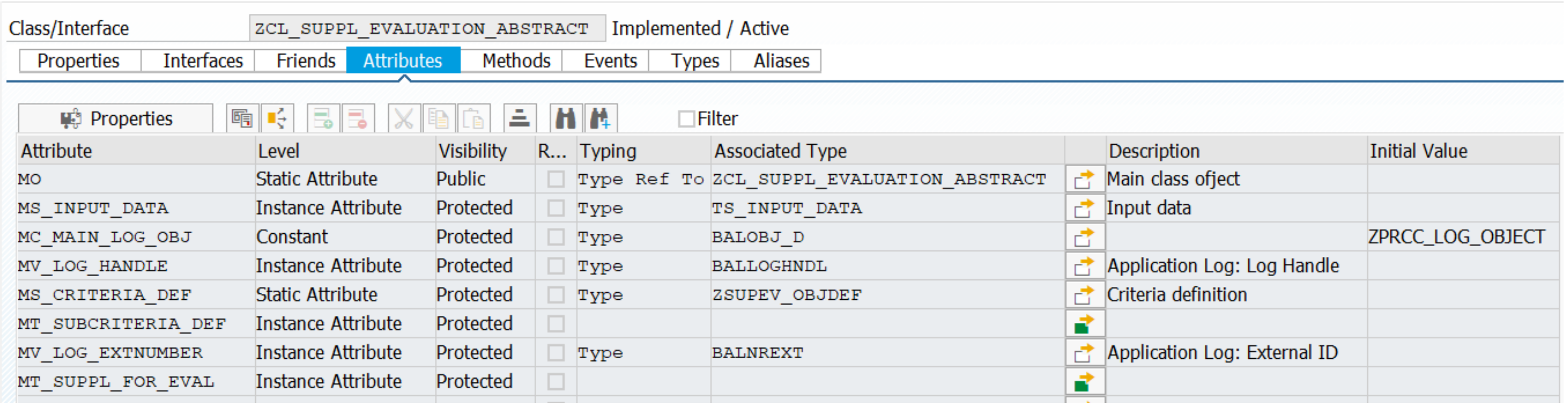
Class/Interface ZCL_SUPPL_EVALUATION_ABSTRACT Implemented / Active

Method	Level	Visibility	M...	Description
GO	Static Method	Public		Start evaluation
CONSTRUCTOR	Instance Method	Public		CONSTRUCTOR
GET_LOG_EXTNUMBER	Instance Method	Public		Get App Log: External ID
SAVE_LOG	Instance Method	Public		Save the log object
INIT	Instance Method	Public		Set initial data
CLEAN_UP	Instance Method	Public		Clean attributes
CREATE_LOG_OBJ	Instance Method	Protected		Create a log object
ADD_MESSAGE_TO_LOG	Instance Method	Protected		Add a message to the log object
GET_CUST_CRITERIA_DEF	Instance Method	Protected		Get a Criteria Definition line
GET_CUST_SUBCRITERIA_DEF	Instance Method	Protected		Get a Sub-criteria Definition Table
FILL_SUPPLIERS_FOR_EVALUATION	Instance Method	Protected		Fill Suppliers for evaluation

Figure 41: The calling class ZCL_SUPPL_EVALUATION_ABSTRACT methods list

Method	Parameters	Description	Use
GO	Importing: • Supplier, • Purchasing Organization, • Main Criterion, • Sub-criterion, • Evaluation Method Exporting: • Sub-criteria score • Table with Return Information • Message type: S Success, E Error, W Warning, I Info, A Abort	Start evaluation	This method is called the customer exit. Run the process of supplier evaluation. Accept the same parameters, as the customer-exit FM. Returns the score for the sub-criteria.

	Exceptions: • Error		
CONSTRUCTOR	Importing: • Supplier, • Purchasing Organization, • Main Criterion, • Sub-criterion, • Evaluation Method Exceptions: • Error	Constructor	This method is called during the creating of the Class object. Initialize required attributes insight.
GET_LOG_EXTNUMBER	Returning: • Application Log: External ID	Get App Log: External ID	Generates the ID for the log.
SAVE_LOG	-	Save the log object	Save the log messages into the data base.
INIT	Exceptions: • Error	Set initial data	Declares required attributes.
CLEAN_UP	-	Clean attributes	Clean attributes
CREATE_LOG_OBJ	Exceptions: • ZCX_MSG	Create a log object	Create a log object. ZCX_MSG – a custom exception class.
ADD_MESSAGE_TO_LOG	Importing • Message Type • Message Class • Message Number • Message Variable • Message Variable • Message Variable • Message Variable	Add a message to the log object	Add a message to the log object.
GET_CUST_CRITERIA_DEF	Exceptions: • ZCX_MSG	Get a Criteria Definition line.	Declares the attribute MS_CRITERIA_DEF with executing class name and log object from the data table ZSUPEV_OBJDEF.
GET_CUST_SUBCRITERIA_DEF	Exceptions: • ZCX_MSG	Get a Sub-criteria Definition Table	Declares the attribute MT_SUBCRITERIA_DEF with executing method names from the data table ZSUPEV_SUBOBJDEF.
FILL_SUPPLIERS_FOR_EVALUATION	Exceptions: • ZCX_MSG	Fill Suppliers for evaluation	Declares the attribute MT_SUPPL_FOR_EVAL with suppliers for the evaluation from the data table ZTSUPEV_LFM1.



Class/Interface ZCL_SUPPL_EVALUATION_ABSTRACT Implemented / Active

Attribute	Level	Visibility	R...	Typing	Associated Type		Description	Initial Value
MO	Static Attribute	Public	□	Type Ref To	ZCL_SUPPL_EVALUATION_ABSTRACT		Main class ofject	
MS_INPUT_DATA	Instance Attribute	Protected	□	Type	TS_INPUT_DATA		Input data	
MC_MAIN_LOG_OBJ	Constant	Protected	□	Type	BALOBJ_D			ZPRCC_LOG_OBJECT
MV_LOG_HANDLE	Instance Attribute	Protected	□	Type	BALLOGHNDL		Application Log: Log Handle	
MS_CRITERIA_DEF	Static Attribute	Protected	□	Type	ZSUPEV_OBJDEF		Criteria definition	
MT_SUBCRITERIA_DEF	Instance Attribute	Protected	□					
MV_LOG_EXTNUMBER	Instance Attribute	Protected	□	Type	BALNREXT		Application Log: External ID	
MT_SUPPL_FOR_EVAL	Instance Attribute	Protected	□					

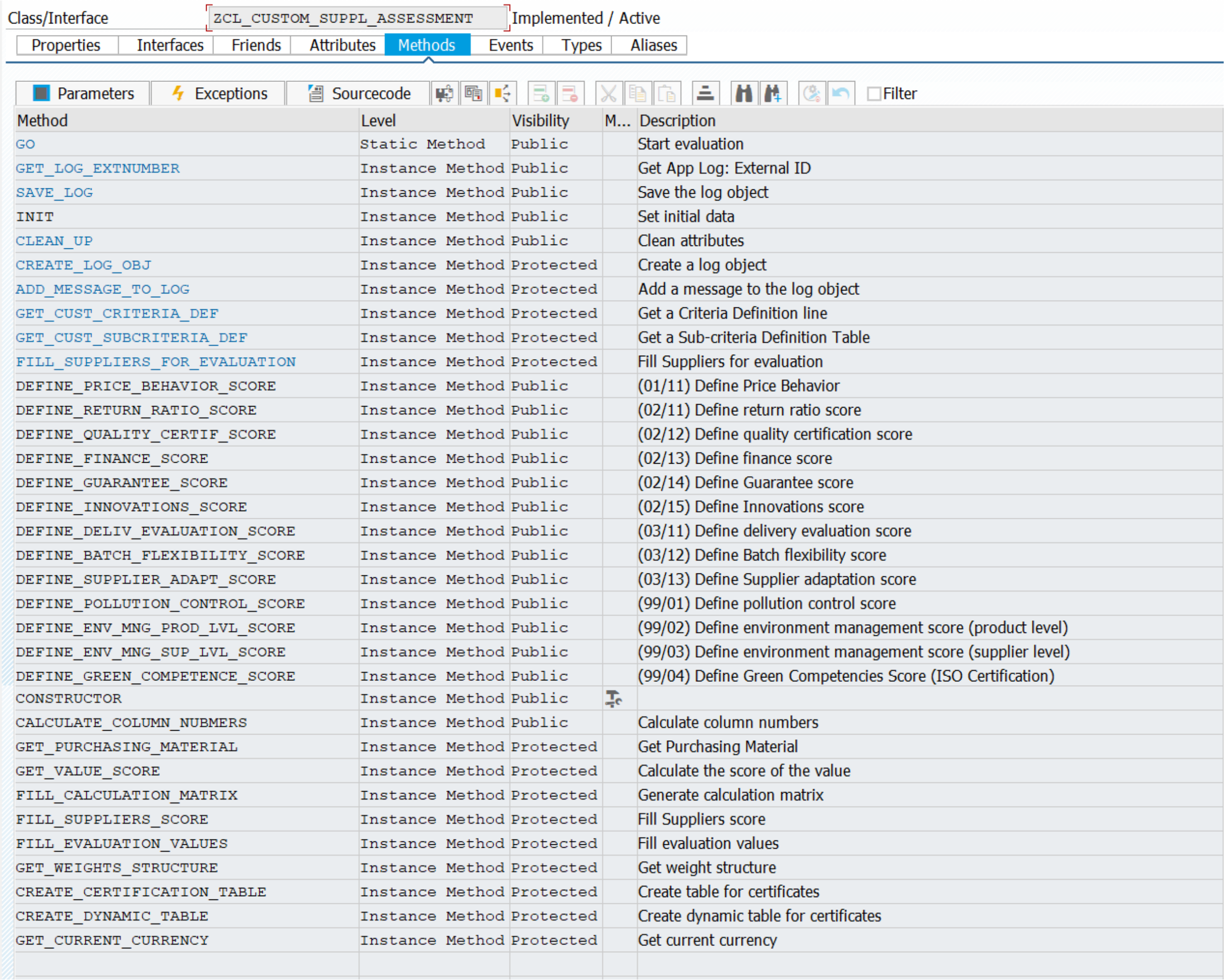
Figure 42: The calling class ZCL_SUPPL_EVALUATION_ABSTRACT attributes list

Attribute	Type	Fields	Use
MO	Class object with the type of the current Class.	-	It’s necessary in dynamic programming.
MS_INPUT_DATA	Structure	Supplier, Purchasing Organization, Main Criterion, Sub-criterion, Evaluation Method	It’s used inside of the Calling class to have necessary data insight.
MC_MAIN_LOG_OBJ	Constant	-	ID of the SAP log object.
MV_LOG_HANDLE	Value	-	Log ID
MS_CRITERIA_DEF	Structure	Purchasing organization Main Criterion Executing Class Name Log: Sub-object	It’s used inside of the Calling class to have necessary data insight.
MT_SUBCRITERIA_DEF	Table	Purchasing organization Main Criterion Sub-criterion Sequence method number Executing Method Name Weighting Table name Indicator: method not active?	It’s used inside of the Calling class to have necessary data insight.
MV_LOG_EXTNUMBER	Value	-	The name of the SAP log object.
MT_SUPPL_FOR_EVAL	Table	Supplier, Purchasing Organization	It’s used inside of the Calling class to have necessary data insight.

```
METHOD go.
  TYPES: BEGIN OF ts_custom_config,
           clsname TYPE seoclsname,
           sub_obj TYPE subobject.
           INCLUDE TYPE zsupev_subobjdef.
  TYPES: END OF ts_custom_config.

  DATA: lt_custom_config TYPE TABLE OF ts_custom_config.
  DATA: lv_XBEURT TYPE elbp-beurt.
  DATA: lv_message TYPE string.

  "check custom config
  SELECT *
    INTO CORRESPONDING FIELDS OF TABLE @lt_custom_config
    FROM zsupev_objdef AS obj
    JOIN zsupev_subobjdef AS sb_obj ON obj~ekorg = sb_obj~ekorg
                                   AND obj~hkrit = sb_obj~hkrit
      WHERE sb_obj~ekorg = @iv_xekorg
      AND   sb_obj~hkrit = @iv_xhkrit
      AND   sb_obj~tkrit = @iv_xtkrit.

  IF sy-subrc <> 0.
    ev_xbeurt = 0.
    MESSAGE e001(zprc_sup_ev) INTO lv_message WITH iv_xhkrit.
    RETURN.
  ENDIF.

  SORT lt_custom_config BY zmeth_num.

  TRY.
      DATA(lv_class_name) = lt_custom_config[ 1 ]-clsname.
      CREATE OBJECT mo TYPE (lv_class_name)
        EXPORTING
          iv_xlifnr = iv_xlifnr
          iv_xekorg = iv_xekorg
          iv_xhkrit = iv_xhkrit
          iv_xtkrit = iv_xtkrit
          iv_xkrtyp = iv_xkrtyp
        EXCEPTIONS
          error     = 1
          OTHERS    = 2.

      IF sy-subrc NE 0.
        MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
          WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4 INTO lv_message.
        RAISE error.
      ENDIF.

    CATCH cx_sy_itab_line_not_found.
      RAISE error.
  ENDTRY.

  LOOP AT lt_custom_config INTO DATA(ls_custom_config) WHERE zis_inactiv = abap_false.
    TRY.
        CALL METHOD mo->(ls_custom_config-method_name)
          EXPORTING
            iv_XLIFNR = iv_xlifnr
            iv_XEKORG = iv_xekorg
            iv_XHKRIT = iv_xhkrit
            iv_XTKRIT = iv_xtkrit
            iv_XKRTYP = iv_xkrtyp
          IMPORTING
            ev_XBEURT = lv_XBEURT.

        IF sy-subrc NE 0.
          MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
            WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4 INTO lv_message.
          mo->save_log( ).
          RAISE error.
        ENDIF.
      CATCH cx_sy_dyn_call_error.          CONTINUE.
    ENDTRY.
    mo->save_log( ).
    ev_xbeurt = ev_xbeurt + lv_XBEURT.
  ENDLOOP.
ENDMETHOD.

METHOD constructor.

  DATA: lv_message TYPE string.

  clean_up( ).

  ms_input_data-xekorg = iv_xekorg.
  ms_input_data-xhkrit = iv_xhkrit.
  ms_input_data-xkrtyp = iv_xkrtyp.
  ms_input_data-xlifnr = iv_xlifnr.
  ms_input_data-xtkrit = iv_xtkrit.

  init(
    EXCEPTIONS
      error  = 1
      OTHERS = 2  ).

  IF sy-subrc NE 0.
    IF sy-msgid IS INITIAL.
      MESSAGE e002(zprc_sup_ev) INTO lv_message.
    ENDIF.
    RAISE error.
  ENDIF.

ENDMETHOD.

METHOD get_log_extnumber.
  rv_extnumber = |{ ms_input_data-xlifnr }_{ ms_input_data-xekorg }_{ ms_input_data-
xhkrit }_{ ms_input_data-xtkrit }_{ ms_input_data-xkrtyp }|.
ENDMETHOD.

METHOD save_log.

  DATA: lt_log_handle TYPE bal_t_logh.
  DATA: bal_t_logh LIKE LINE OF lt_log_handle.

  INSERT me->mv_log_handle INTO TABLE lt_log_handle.

  CALL FUNCTION 'BAL_DB_SAVE'
    EXPORTING
      i_t_log_handle   = lt_log_handle
    EXCEPTIONS
      log_not_found    = 1
      save_not_allowed = 2
      numbering_error  = 3
      OTHERS           = 4.
ENDMETHOD.

METHOD init.
  DATA: lv_message TYPE string.

  TRY.
      get_cust_criteria_def( ).                   "fill in criteria config
      get_cust_subcriteria_def( ).
      mv_log_extnumber = get_log_extnumber( ).
      create_log_obj( ).
      fill_suppliers_for_evaluation( ).

    CATCH zcx_msg INTO DATA(lo_msg).
      IF sy-msgid IS INITIAL.
        MESSAGE e002(zprc_sup_ev) INTO lv_message.
      ENDIF.
      RAISE error.
  ENDTRY.

ENDMETHOD.
```

```
METHOD clean_up.

  CLEAR:
    ms_input_data,
    mv_log_handle,
    ms_criteria_def,
    mt_subcriteria_def,
    mv_log_extnumber.

ENDMETHOD.


METHOD create_log_obj.
  DATA: ls_log TYPE bal_s_log.
  DATA: lv_message TYPE string.

  ls_log-object    = mc_main_log_obj.
  ls_log-subobject = ms_criteria_def-subobject.
  ls_log-extnumber = mv_log_extnumber.
  ls_log-aldate    = sy-datum.
  ls_log-altime    = sy-uzeit.
  ls_log-aluser    = sy-uname.
  ls_log-alprog    = sy-repid.

  CALL FUNCTION 'BAL_LOG_CREATE'
    EXPORTING
      i_s_log                 = ls_log
    IMPORTING
      e_log_handle            = mv_log_handle
    EXCEPTIONS
      log_header_inconsistent = 1
      OTHERS                  = 2.

  IF sy-subrc NE 0.
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
            WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4 INTO lv_message.
    RAISE EXCEPTION TYPE zcx_msg EXPORTING ms_syst = sy.
  ENDIF.

ENDMETHOD.


METHOD add_message_to_log.
  DATA: ls_msg TYPE bal_s_msg.

  "-------------------------------------------------------
  IF iv_msgid IS INITIAL OR iv_msgno IS INITIAL.
    ls_msg-msgty = sy-msgty.
    ls_msg-msgid = sy-msgid.
    ls_msg-msgno = sy-msgno.
    ls_msg-msgv1 = sy-msgv1.
    ls_msg-msgv2 = sy-msgv2.
    ls_msg-msgv3 = sy-msgv3.
    ls_msg-msgv4 = sy-msgv4.
  ELSE.
    ls_msg-msgty = iv_msgty.
    ls_msg-msgid = iv_msgid.
    ls_msg-msgno = iv_msgno.
    ls_msg-msgv1 = iv_msgv1.
    ls_msg-msgv2 = iv_msgv2.
    ls_msg-msgv3 = iv_msgv3.
    ls_msg-msgv4 = iv_msgv4.
  ENDIF.

  IF ls_msg-msgty IS INITIAL.
    ls_msg-msgty = 'E'.
  ENDIF.

  IF ls_msg-msgid IS INITIAL OR ls_msg-msgno IS INITIAL.
    ls_msg-msgid = 'ZPRC_SUP_EV'.
    ls_msg-msgno = 000.
    ls_msg-msgv1 = 'The message is not defined.'.
  ENDIF.

  "-------------------------------------------------------
  CALL FUNCTION 'BAL_LOG_MSG_ADD'
    EXPORTING
      i_log_handle     = mv_log_handle
      i_s_msg          = ls_msg
    EXCEPTIONS
      log_not_found    = 1
      msg_inconsistent = 2
      log_is_full      = 3
      OTHERS           = 4.

  IF sy-subrc NE 0.

  ENDIF.
ENDMETHOD.

METHOD get_cust_criteria_def.
  DATA: lv_message TYPE string.

  SELECT SINGLE *
    INTO ms_criteria_def
    FROM zsupev_objdef
      WHERE ekorg = ms_input_data-xekorg
      AND   hkrit = ms_input_data-xhkrit.

  IF sy-subrc NE 0.
    MESSAGE e001(zprc_sup_ev) INTO lv_message WITH ms_input_data-xhkrit.
    RAISE EXCEPTION TYPE zcx_msg EXPORTING ms_syst = sy.
  ENDIF.
ENDMETHOD.

METHOD get_cust_subcriteria_def.
  DATA: lv_message TYPE string.

  SELECT *
    FROM zsupev_objdef AS obj
    JOIN zsupev_subobjdef AS sb_obj ON obj~ekorg = sb_obj~ekorg
                                   AND obj~hkrit = sb_obj~hkrit
      WHERE sb_obj~ekorg = @ms_input_data-xekorg
      AND   sb_obj~hkrit = @ms_input_data-xhkrit
      AND   sb_obj~tkrit = @ms_input_data-xtkrit
  INTO CORRESPONDING FIELDS OF TABLE @mt_subcriteria_def.

  IF sy-subrc NE 0.
    MESSAGE e003(zprc_sup_ev) INTO lv_message
      WITH ms_input_data-xekorg ms_input_data-xhkrit ms_input_data-xtkrit.
    RAISE EXCEPTION TYPE zcx_msg EXPORTING ms_syst = sy.
  ENDIF.
ENDMETHOD.

METHOD fill_suppliers_for_evaluation.
  DATA: lv_message TYPE string.

  SELECT *
    FROM ztsupev_lfm1
      WHERE ekorg = @ms_criteria_def-ekorg
  INTO CORRESPONDING FIELDS OF TABLE @mt_suppl_for_eval.

  IF sy-subrc NE 0.
    MESSAGE e005(zprc_sup_ev) INTO lv_message WITH ms_input_data-xhkrit.
    RAISE EXCEPTION TYPE zcx_msg EXPORTING ms_syst = sy.
  ELSE.
    MESSAGE i006(zprc_sup_ev) INTO lv_message.
    me->add_message_to_log( ).
  ENDIF.
ENDMETHOD.
```

Attachment 10: The executing class ZCL_CUSTOM_SUPPL_ASSESSMENT. Architecture and code.



Class/Interface ZCL_CUSTOM_SUPPL_ASSESSMENT Implemented / Active

Method	Level	Visibility	M...	Description
GO	Static Method	Public		Start evaluation
GET_LOG_EXTNUMBER	Instance Method	Public		Get App Log: External ID
SAVE_LOG	Instance Method	Public		Save the log object
INIT	Instance Method	Public		Set initial data
CLEAN_UP	Instance Method	Public		Clean attributes
CREATE_LOG_OBJ	Instance Method	Protected		Create a log object
ADD_MESSAGE_TO_LOG	Instance Method	Protected		Add a message to the log object
GET_CUST_CRITERIA_DEF	Instance Method	Protected		Get a Criteria Definition line
GET_CUST_SUBCRITERIA_DEF	Instance Method	Protected		Get a Sub-criteria Definition Table
FILL_SUPPLIERS_FOR_EVALUATION	Instance Method	Protected		Fill Suppliers for evaluation
DEFINE_PRICE_BEHAVIOR_SCORE	Instance Method	Public		(01/11) Define Price Behavior
DEFINE_RETURN_RATIO_SCORE	Instance Method	Public		(02/11) Define return ratio score
DEFINE_QUALITY_CERTIF_SCORE	Instance Method	Public		(02/12) Define quality certification score
DEFINE_FINANCE_SCORE	Instance Method	Public		(02/13) Define finance score
DEFINE_GUARANTEE_SCORE	Instance Method	Public		(02/14) Define Guarantee score
DEFINE_INNOVATIONS_SCORE	Instance Method	Public		(02/15) Define Innovations score
DEFINE_DELIV_EVALUATION_SCORE	Instance Method	Public		(03/11) Define delivery evaluation score
DEFINE_BATCH_FLEXIBILITY_SCORE	Instance Method	Public		(03/12) Define Batch flexibility score
DEFINE_SUPPLIER_ADAPT_SCORE	Instance Method	Public		(03/13) Define Supplier adaptation score
DEFINE_POLLUTION_CONTROL_SCORE	Instance Method	Public		(99/01) Define pollution control score
DEFINE_ENV_MNG_PROD_LVL_SCORE	Instance Method	Public		(99/02) Define environment management score (product level)
DEFINE_ENV_MNG_SUP_LVL_SCORE	Instance Method	Public		(99/03) Define environment management score (supplier level)
DEFINE_GREEN_COMPETENCE_SCORE	Instance Method	Public		(99/04) Define Green Competencies Score (ISO Certification)
CONSTRUCTOR	Instance Method	Public		
CALCULATE_COLUMN_NUBMERS	Instance Method	Public		Calculate column numbers
GET_PURCHASING_MATERIAL	Instance Method	Protected		Get Purchasing Material
GET_VALUE_SCORE	Instance Method	Protected		Calculate the score of the value
FILL_CALCULATION_MATRIX	Instance Method	Protected		Generate calculation matrix
FILL_SUPPLIERS_SCORE	Instance Method	Protected		Fill Suppliers score
FILL_EVALUATION_VALUES	Instance Method	Protected		Fill evaluation values
GET_WEIGHTS_STRUCTURE	Instance Method	Protected		Get weight structure
CREATE_CERTIFICATION_TABLE	Instance Method	Protected		Create table for certificates
CREATE_DYNAMIC_TABLE	Instance Method	Protected		Create dynamic table for certificates
GET_CURRENT_CURRENCY	Instance Method	Protected		Get current currency

Figure 43: The executing class ZCL_CUSTOM_SUPPL_ASSESSMENT methods list.

Blue records – inherited methods, **Gray records** – native class methods

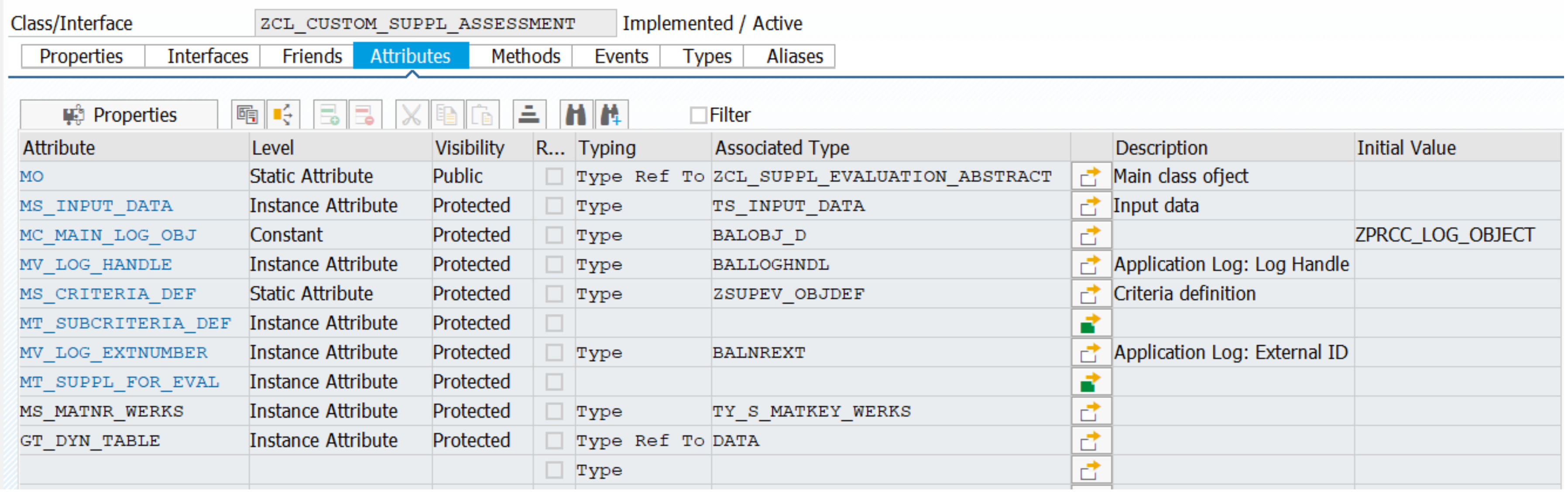
Method	Parameters	Description	Use
DEFINE_PRICE_BEHAVIOR_SCORE	Importing: • Supplier, • Purchasing Organization, • Main Criterion, • Sub-criterion, • Evaluation Method Exporting: • Score Exceptions: • Error	(01/11) Define Price Behavior	Define the score for the criterion 01 “Price”, Sub-criterion 11 “Price Behavior”.
DEFINE_RETURN_RATIO_SCORE	Importing: • Supplier, • Purchasing Organization, • Main Criterion, • Sub-criterion, • Evaluation Method Exporting: • Score Exceptions: • Error	(02/11) Define return ratio score	Define the score for the criterion 02 “Quality”, Sub-criterion 11 “Return ratio”.
DEFINE_QUALITY_CERTIF_SCORE	Importing: • Supplier, • Purchasing Organization, • Main Criterion, • Sub-criterion, • Evaluation Method Exporting: • Score Exceptions: • Error	(02/12) Define quality certification score	Define the score for the criterion 02 “Quality”, Sub-criterion 12 “Quality Certificats”.
DEFINE_FINANCE_SCORE	Importing: • Supplier, • Purchasing Organization, • Main Criterion, • Sub-criterion, • Evaluation Method Exporting: • Score Exceptions: • Error	(02/13) Define finance score	Define the score for the criterion 02 “Quality”, Sub-criterion 13 “Finance”.

DEFINE_GUARANTEE_SCORE	Importing: • Supplier, • Purchasing Organization, • Main Criterion, • Sub-criterion, • Evaluation Method Exporting: • Score Exceptions: • Error	(02/14) Define Guarantee score	Define the score for the criterion 02 “Quality”, Sub-criterion 14 “Guarantee”.
DEFINE_INNOVATIONS_SCORE	Importing: • Supplier, • Purchasing Organization, • Main Criterion, • Sub-criterion, • Evaluation Method Exporting: • Score Exceptions: • Error	(02/15) Define Innovations score	Define the score for the criterion 02 “Quality”, Sub-criterion 15 “Innovations”.
DEFINE_DELIV_EVALUATION_SCORE	Importing: • Supplier, • Purchasing Organization, • Main Criterion, • Sub-criterion, • Evaluation Method Exporting: • Score Exceptions: • Error	(05/11) Define delivery evaluation score	Define the score for the criterion 05 “Service”, Sub-criterion 11 “Delivery evaluation”.
DEFINE_BATCH_FLEXIBILITY_SCORE	Importing: • Supplier, • Purchasing Organization, • Main Criterion, • Sub-criterion, • Evaluation Method Exporting: • Score Exceptions: • Error	(05/12) Define Batch flexibility score	Define the score for the criterion 05 “Service”, Sub-criterion 12 “Batch flexibility”.
DEFINE_SUPPLIER_ADAPT_SCORE	Importing: • Supplier, • Purchasing Organization, • Main Criterion, • Sub-criterion,	(05/13) Define Supplier adaptation score	Define the score for the criterion 05 “Service”, Sub-criterion 13 “Supplier flexibility”.

	• Evaluation Method Exporting: • Score Exceptions: • Error		
DEFINE_POLLUTION_CONTROL_SCORE	Importing: • Supplier, • Purchasing Organization, • Main Criterion, • Sub-criterion, • Evaluation Method Exporting: • Score Exceptions: • Error	(99/01) Define pollution control score	Define the score for the criterion 99 “Green Criteria”, Sub-criterion 01 “Pollution control”.
DEFINE_ENV_MNG_PROD_LVL_SCORE	Importing: • Supplier, • Purchasing Organization, • Main Criterion, • Sub-criterion, • Evaluation Method Exporting: • Score Exceptions: • Error	(99/02) Define environment management score (product level)	Define the score for the criterion 99 “Green Criteria”, Sub-criterion 02 “Environmental management (product level)”.
DEFINE_ENV_MNG_SUP_LVL_SCORE	Importing: • Supplier, • Purchasing Organization, • Main Criterion, • Sub-criterion, • Evaluation Method Exporting: • Score Exceptions: • Error	(99/03) Define environment management score (supplier level)	Define the score for the criterion 99 “Green Criteria”, Sub-criterion 03 “Environmental management (general level)”.
DEFINE_GREEN_COMPETENCE_SCORE	Importing: • Supplier, • Purchasing Organization, • Main Criterion, • Sub-criterion, • Evaluation Method Exporting: • Score Exceptions: • Error	(99/04) Define Green Competencies Score (ISO Certification)	Define the score for the criterion 99 “Green Criteria”, Sub-criterion 04 “Green competencies”.

CONSTRUCTOR	Importing: • Supplier, • Purchasing Organization, • Main Criterion, • Sub-criterion, • Evaluation Method Exceptions: • Error	Constructor	This method is called during the creating of the Class object. Initialize required attributes insight.
CALCULATE_COLUMN_NUBMERS	Importing: • Table of any type Exporting: • The number of columns	Calculate column numbers	The method calculates the number of columns of the input table.
GET_PURCHASING_MATERIAL	Exceptions: • ZCX_MSG	Get Purchasing Material	The method takes the material for the evaluation from the table ZTSUPEV_MARC “Data base for evaluated materials”.
GET_VALUE_SCORE	Importing: • Custom float number table • Custom float value to be scored • Flag: is it beneficial parameter? Exporting: • The score of custom float number	Calculate the score of the value	This method is a part of PROMETEE II logic. Accepts the non-normalized value table, returns the normalized value.
FILL_CALCULATION_MATRIX	Importing: • Suppliers’ evaluation normalized table • Numbers of parameters to calculate • Weights structure • Weights key column number Exporting: • Suppliers calculation matrix Exceptions: • ZCX_MSG	Generate calculation matrix	This method is a part of PROMETEE II logic. Accepts the normalized data table, returns the calculation matrix.
FILL_SUPPLIERS_SCORE	Importing: • Suppliers calculation matrix Exporting: • Suppliers score	Fill Suppliers score	This method is a part of PROMETEE II logic. Fills the table of suppliers scores.
FILL_EVALUATION_VALUES	Importing: • Suppliers raw values table • Number of key fields • Beneficial / non-beneficial • Weights key column number Exporting:	Fill evaluation values	This method is a part of PROMETEE II logic. Accepts the non-normalized values table, returns the normalized table.

	• Suppliers calculation matrix Exceptions: ZCX_MSG		
GET_WEIGHTS_STRUCTURE	Importing: • Weighting Table name Exporting: • Weights structure Exceptions: • ZCX_MSG	Get weight structure	This method is a part of PROMETEE II logic. Returns the weighting values by the weighting table name.
CREATE_CERTIFICATION_TABLE	Importing: • Information Category Exporting: • Certification data Changing • Additional BP info	Create table for certificates	This method is a part of PROMETEE II logic. Transform the Certification values to the normalized numerical data.
CREATE_DYNAMIC_TABLE	Importing: • Information Category • Information Type • Field Name • Data Type in ABAP Dictionary • Internal Length in Bytes • Number of Decimal Places Exporting: • Dynamic table • Certificates table	Create dynamic table for certificates	This method is a part of PROMETEE II logic. Creates a dynamic table for certificates
GET_CURRENT_CURRENCY	-	Get current currency	This method is a part of PROMETEE II logic. It’s used to transform the suppliers currencies.



Class/Interface ZCL_CUSTOM_SUPPL_ASSESSMENT Implemented / Active

Attribute	Level	Visibility	R...	Typing	Associated Type	Description	Initial Value
MO	Static Attribute	Public	□	Type Ref To	ZCL_SUPPL_EVALUATION_ABSTRACT	Main class ofject	
MS_INPUT_DATA	Instance Attribute	Protected	□	Type	TS_INPUT_DATA	Input data	
MC_MAIN_LOG_OBJ	Constant	Protected	□	Type	BALOBJ_D		ZPRCC_LOG_OBJECT
MV_LOG_HANDLE	Instance Attribute	Protected	□	Type	BALLOGHNDL	Application Log: Log Handle	
MS_CRITERIA_DEF	Static Attribute	Protected	□	Type	ZSUPEV_OBJDEF	Criteria definition	
MT_SUBCRITERIA_DEF	Instance Attribute	Protected	□				
MV_LOG_EXTNUMBER	Instance Attribute	Protected	□	Type	BALNREXT	Application Log: External ID	
MT_SUPPL_FOR_EVAL	Instance Attribute	Protected	□				
MS_MATNR_WERKS	Instance Attribute	Protected	□	Type	TY_S_MATKEY_WERKS		
GT_DYN_TABLE	Instance Attribute	Protected	□	Type Ref To	DATA		
			□	Type			

Figure 44: The executing class ZCL_CUSTOM_SUPPL_ASSESSMENT attributes list

Blue records – inherited attributes, **Gray records** – native class attributes

Attribute	Type	Fields	Use
MS_MATNR_WERKS	Structure	Material Plant	The evaluated material and its plant level.
GT_DYN_TABLE	Structure	ANY	Initial structure.

```
  METHOD define_price_behavior_score.
    TYPES:
      BEGIN OF ty_prices,
        supplier        TYPE lifnr,
        current_price   TYPE zde_value_type,
        previous_price  TYPE zde_value_type,
        effective_price TYPE zde_value_type,
      END OF ty_prices,

      BEGIN OF ty_price_behav_values,
        supplier         TYPE lifnr,
        pct_price_change TYPE zde_value_type,
      END OF ty_price_behav_values.


    DATA:
      ls_env_mng_w          TYPE ztb_price_behv_w,
      lv_first_day          TYPE sy-datum,
      lv_last_day           TYPE sy-datum,
      lv_valid_price_per    TYPE i,
      ls_prices             TYPE ty_prices,
      lt_prices             TYPE TABLE OF ty_prices,
      ls_price_behav_values TYPE ty_price_behav_values,
      lt_price_behav_values TYPE TABLE OF ty_price_behav_values,
      lt_suppl_ev           TYPE TABLE OF ty_price_behav_values,
      lt_suppl_calc         TYPE TABLE OF ty_suppl_calc,
      ls_suppliers_score    TYPE ty_suppliers_score,
      lt_suppliers_score    TYPE TABLE OF ty_suppliers_score.


    get_current_currency( ).

    CALL FUNCTION 'FIRST_AND_LAST_DAY_IN_YEAR_GET'
      EXPORTING
        i_gjahr     = CONV t009b-bdatj( sy-datum(4) - 1 )
        i_periv     = 'Z1'
      IMPORTING
        e_first_day = lv_first_day
        e_last_day  = lv_last_day.

    SELECT a~lifnr, a~datab, a~datbi, kp~kbetr, kp~konwa
      INTO TABLE @DATA(lt_price_cond)
      FROM a018 AS a
        JOIN konp AS kp ON a~knumh = kp~knumh
      FOR ALL ENTRIES IN @mt_suppl_for_eval
      WHERE a~lifnr =  @mt_suppl_for_eval-lifnr
        AND a~ekorg =  @iv_xekorg
        AND a~datbi >= @lv_first_day
        AND a~matnr =  @ms_matnr_werks-matnr.

    IF sy-subrc NE 0.
      MESSAGE w016(zprc_sup_ev) INTO DATA(lv_message).
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.

    SORT lt_price_cond BY lifnr ASCENDING
                          datbi DESCENDING.

    LOOP AT mt_suppl_for_eval INTO DATA(ls_suppl_for_eval).
      CHECK line_exists( lt_price_cond[ lifnr = ls_suppl_for_eval-lifnr ] ).

      ls_prices-supplier = ls_suppl_for_eval-lifnr.

      LOOP AT lt_price_cond ASSIGNING FIELD-SYMBOL(<wa_price_cond>) WHERE lifnr =  ls_suppl_for_eval-
lifnr.
        IF <wa_price_cond>-konwa <> sy-waers.
          CALL FUNCTION 'CONVERT_TO_LOCAL_CURRENCY'
            EXPORTING
              client           = sy-mandt
              date             = sy-datum
              foreign_amount   = <wa_price_cond>-kbetr
              foreign_currency = <wa_price_cond>-konwa
              local_currency   = sy-waers
              type_of_rate     = 'M'
            IMPORTING
              local_amount     = <wa_price_cond>-kbetr.

          <wa_price_cond>-konwa = sy-waers.
        ENDIF.

        IF <wa_price_cond>-datab <= sy-datum AND sy-datum <= <wa_price_cond>-datbi .
          ls_prices-current_price = <wa_price_cond>-kbetr.
          CONTINUE.
        ENDIF.

        lv_valid_price_per = lv_valid_price_per + CONV i( <wa_price_cond>-datbi - <wa_price_cond>-
datab + 1 ).
        ls_prices-effective_price = ls_prices-effective_price +
                                    <wa_price_cond>-kbetr * CONV i( <wa_price_cond>-
datbi - <wa_price_cond>-datab + 1 ).

      ENDLOOP.

      IF ls_prices-current_price IS INITIAL.
        CLEAR ls_prices.
        CONTINUE.
      ENDIF.

      "effective price
      ls_prices-effective_price = ls_prices-effective_price / lv_valid_price_per.

      APPEND ls_prices TO lt_prices.
      CLEAR: lv_valid_price_per, ls_prices.

      ls_suppliers_score-supplier = ls_suppl_for_eval-lifnr.
      APPEND ls_suppliers_score TO lt_suppliers_score.

    ENDLOOP.

    LOOP AT lt_prices INTO ls_prices.
      ls_price_behav_values-supplier = ls_prices-supplier.
      IF ls_prices-current_price   IS NOT INITIAL AND ls_prices-effective_price IS NOT INITIAL.
        ls_price_behav_values-pct_price_change = ( ls_prices-current_price - ls_prices-
effective_price ) / ls_prices-effective_price * 100.
      ELSEIF  ls_prices-effective_price IS INITIAL.
        ls_price_behav_values-pct_price_change = 0.
      ENDIF.
      APPEND ls_price_behav_values TO lt_price_behav_values.
    ENDLOOP.

    fill_evaluation_values(
      EXPORTING
        it_values_raw = lt_price_behav_values
        iv_key_fields_no = 1
        "is_env_mng_values_param_types = ls_env_mng_values_param_types
      IMPORTING
        et_suppl_ev = lt_suppl_ev
    ).

    IF mt_subcriteria_def[ tkrit = iv_xtkrit ]-zz_weight_table IS NOT INITIAL.
      TRY .
          get_weights_structure(
            EXPORTING
              iv_table_name_w = mt_subcriteria_def[ tkrit = iv_xtkrit ]-zz_weight_table
            IMPORTING
              es_weights = ls_env_mng_w
          ).
        CATCH zcx_msg INTO DATA(lo_msg).
          RAISE error.
      ENDTRY.
    ENDIF.

    fill_calculation_matrix(
      EXPORTING
        it_suppl_ev = lt_suppl_ev
        iv_param_col_no = 1
        is_weight = ls_env_mng_w
        iv_weight_key_no = 2
      IMPORTING
        et_suppl_calc = lt_suppl_calc
     ).

    fill_suppliers_score(
      EXPORTING
        it_suppl_calc      = lt_suppl_calc
      CHANGING
        ct_suppliers_score = lt_suppliers_score
    ).
    IF line_exists( lt_suppliers_score[ supplier = iv_xlifnr ] ).
      ev_xbeurt = CONV elbp-beurt( lt_suppliers_score[ supplier = iv_xlifnr ]-evaluation ).
    ELSE.
      ev_xbeurt = 0.
    ENDIF.
  ENDMETHOD.
```

```
METHOD define_return_ratio_score.

  TYPES:
    BEGIN OF ty_return_pct_values,
      supplier   TYPE lifnr,
      return_pct TYPE zde_value_type,
    END OF ty_return_pct_values.

  DATA:
    lv_input_qty         TYPE zde_value_type,
    lv_ret_qty           TYPE zde_value_type,
    ls_return_pct_values TYPE ty_return_pct_values,
    lt_return_pct_values TYPE TABLE OF ty_return_pct_values,
    lt_suppl_ev          TYPE TABLE OF ty_return_pct_values,
    lt_suppl_calc        TYPE TABLE OF ty_suppl_calc,
    ls_suppliers_score   TYPE ty_suppliers_score,
    lt_suppliers_score   TYPE TABLE OF ty_suppliers_score.


  SELECT lifnr, bwart, menge
    INTO TABLE @DATA(lt_mat_docs)
    FROM mseg
    FOR ALL ENTRIES IN @mt_suppl_for_eval
    WHERE lifnr   = @mt_suppl_for_eval-lifnr
      AND ( bwart = @zprcc_bwart-101 OR bwart = @zprcc_bwart-161 ).


  LOOP AT mt_suppl_for_eval INTO DATA(ls_suppl_for_eval).

    ls_suppliers_score-supplier = ls_suppl_for_eval-lifnr.
    APPEND ls_suppliers_score TO lt_suppliers_score.

    IF NOT line_exists( lt_mat_docs[ lifnr = ls_suppl_for_eval-lifnr ] ).
      ls_return_pct_values-supplier = ls_suppl_for_eval-lifnr.
      ls_return_pct_values-return_pct = 0.
      APPEND ls_return_pct_values TO lt_return_pct_values.
      CONTINUE.
    ENDIF.

  ENDLOOP.

  LOOP AT lt_mat_docs INTO DATA(ls_mat_docs)
                      GROUP BY ( lifnr  = ls_mat_docs-lifnr ) ASCENDING
                      REFERENCE INTO DATA(lg_lifnr).
      LOOP AT GROUP lg_lifnr INTO DATA(ls_lifnr).
        IF ls_lifnr-bwart = zprcc_bwart-101.
          lv_input_qty = lv_input_qty + ls_lifnr-menge.
        ELSEIF ls_lifnr-bwart = zprcc_bwart-161.
          lv_ret_qty = lv_ret_qty + ls_lifnr-menge.
        ENDIF.
      ENDLOOP.
      ls_return_pct_values-supplier = lg_lifnr->lifnr.
      TRY .
          ls_return_pct_values-return_pct = lv_ret_qty / lv_input_qty * 100.
        CATCH cx_sy_zerodivide.
          ls_return_pct_values-return_pct = 0.
      ENDTRY.
      APPEND ls_return_pct_values TO lt_return_pct_values.

      CLEAR: lv_input_qty, lv_ret_qty, ls_return_pct_values.

    ENDLOOP.

    fill_evaluation_values(
      EXPORTING
        it_values_raw = lt_return_pct_values
        iv_key_fields_no = 1
        "is_env_mng_values_param_types = ls_env_mng_values_param_types
      IMPORTING
        et_suppl_ev = lt_suppl_ev
    ).


    fill_calculation_matrix(
      EXPORTING
        it_suppl_ev = lt_suppl_ev
        iv_param_col_no = 1
      IMPORTING
        et_suppl_calc = lt_suppl_calc
     ).

    fill_suppliers_score(
      EXPORTING
        it_suppl_calc      = lt_suppl_calc
      CHANGING
        ct_suppliers_score = lt_suppliers_score
    ).

    IF line_exists( lt_suppliers_score[ supplier = iv_xlifnr ] ).
      ev_xbeurt = CONV elbp-beurt( lt_suppliers_score[ supplier = iv_xlifnr ]-
evaluation ).
    ELSE.
      ev_xbeurt = 0.
    ENDIF.
  ENDMETHOD.

  METHOD define_quality_certif_score.

    DATA:
      lv_message         TYPE string,
      lt_bp_data_raw     TYPE TABLE OF ts_bp_data_raw,
      lt_suppl_calc      TYPE TABLE OF ty_suppl_calc,
      ls_suppliers_score TYPE ty_suppliers_score,
      lt_suppliers_score TYPE TABLE OF ty_suppliers_score.

    FIELD-SYMBOLS:
       <gfs_dyn_table> TYPE STANDARD TABLE.


    SELECT *
      INTO CORRESPONDING FIELDS OF TABLE @lt_bp_data_raw
      FROM bp3100 AS bp
        JOIN but000 AS but ON bp~partner = but~partner
        JOIN cvi_vend_link AS link ON but~partner_guid = link~partner_guid
      FOR ALL ENTRIES IN @mt_suppl_for_eval
      WHERE link~vendor = @mt_suppl_for_eval-lifnr
        AND bp~addtype  = @zprcc_bp3100_add_type-91.

    IF sy-subrc NE 0.
      MESSAGE w017(zprc_sup_ev) INTO lv_message.
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.
    SORT lt_bp_data_raw BY vendor DESCENDING.

    IF NOT line_exists( lt_bp_data_raw[ vendor = iv_xlifnr ] ).
      MESSAGE e015(zprc_sup_ev) INTO lv_message WITH iv_xlifnr.       "General env
ironmental management data for the supplier is not found.
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.

    LOOP AT mt_suppl_for_eval INTO DATA(ls_suppl_for_eval).
      CHECK line_exists( lt_bp_data_raw[ vendor = ls_suppl_for_eval-lifnr ] ).
      ls_suppliers_score-supplier = ls_suppl_for_eval-lifnr.
      APPEND ls_suppliers_score TO lt_suppliers_score.
    ENDLOOP.

    "----------------------------------------------------------------------------

    create_dynamic_table(
      EXPORTING
        iv_addtype = zprcc_bp3100_add_type-91
      IMPORTING
        et_dynamic_table = gt_dyn_table
     ).

    ASSIGN gt_dyn_table->* TO <gfs_dyn_table>.

    create_certification_table(
      EXPORTING
        iv_addtype     = zprcc_bp3100_add_type-91
      IMPORTING
        et_certif_data = <gfs_dyn_table>
      CHANGING
        ct_bp_data_raw = lt_bp_data_raw
     ).

    CALL METHOD me->calculate_column_nubmers
      EXPORTING
```

```
        it_any_table         = <gfs_dyn_table>
      IMPORTING
        ev_number_of_columns = DATA(lv_number_of_columns).

    fill_calculation_matrix(
      EXPORTING
        it_suppl_ev = <gfs_dyn_table>
        iv_param_col_no = lv_number_of_columns - 2
        "is_weight = ls_env_mng_w
        "iv_weight_key_no = 2
      IMPORTING
        et_suppl_calc = lt_suppl_calc
     ).

    fill_suppliers_score(
      EXPORTING
        it_suppl_calc      = lt_suppl_calc
      CHANGING
        ct_suppliers_score = lt_suppliers_score
    ).

    IF line_exists( lt_suppliers_score[ supplier = iv_xlifnr ] ).
      ev_xbeurt = CONV elbp-beurt( lt_suppliers_score[ supplier = iv_xlifnr ]-
evaluation ).
    ELSE.
      ev_xbeurt = 0.
    ENDIF.
  ENDMETHOD.

  METHOD define_finance_score.

    TYPES:
      BEGIN OF ty_finance_values,
        supplier      TYPE lifnr,
        current_ratio TYPE zde_value_type,
        creit_rating  TYPE zde_value_type,
        payment_time  TYPE zde_value_type,
      END OF ty_finance_values,

      BEGIN OF ty_finance_val_type,
        current_ratio TYPE abap_bool,
        creit_rating  TYPE abap_bool,
        payment_time  TYPE abap_bool,
      END OF ty_finance_val_type.

    DATA:
      lt_bp_raw           TYPE TABLE OF ts_bp_data_raw_2,
      ls_finance_values   TYPE ty_finance_values,
      lt_finance_values   TYPE TABLE OF ty_finance_values,
      ls_finance_val_type TYPE ty_finance_val_type,
      lt_suppl_ev         TYPE TABLE OF ty_finance_values,
      ls_env_mng_w        TYPE ztb_finance_w,
      lt_suppl_calc       TYPE TABLE OF ty_suppl_calc,
      ls_suppliers_score  TYPE ty_suppliers_score,
      lt_suppliers_score  TYPE TABLE OF ty_suppliers_score.

    SELECT *
      INTO CORRESPONDING FIELDS OF TABLE @lt_bp_raw
      FROM bp3100 AS bp
```

```
        JOIN but000 AS but ON bp~partner = but~partner
        JOIN cvi_vend_link AS link ON but~partner_guid = link~partner_guid
        LEFT JOIN ztb_creit_rat_w AS w ON bp~ct_field = w~zz_cred_rat
      FOR ALL ENTRIES IN @mt_suppl_for_eval
      WHERE link~vendor = @mt_suppl_for_eval-lifnr
        AND bp~addtype     = @zprcc_bp3100_add_type-93.

    IF sy-subrc NE 0.
      MESSAGE w018(zprc_sup_ev) INTO DATA(lv_message).
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.

    LOOP AT mt_suppl_for_eval INTO DATA(ls_suppl_for_eval).
      IF NOT line_exists( lt_bp_raw[ vendor = ls_suppl_for_eval-lifnr ] ).
        ls_finance_values-supplier = ls_suppl_for_eval-lifnr.
        ls_finance_values-current_ratio = 0.
        ls_finance_values-creit_rating = 22.
        ls_finance_values-payment_time = 0.
      ELSE.
        ls_finance_values-supplier = ls_suppl_for_eval-lifnr.
        ls_finance_values-
current_ratio = CONV zde_value_type( lt_bp_raw[ vendor = ls_suppl_for_eval-
lifnr data_type = zprcc_tp22_data_type-01 ]-text ).
        ls_finance_values-
creit_rating  = CONV zde_value_type( lt_bp_raw[ vendor = ls_suppl_for_eval-
lifnr data_type = zprcc_tp22_data_type-02 ]-zz_cred_rat_w ).
        ls_finance_values-
payment_time  = CONV zde_value_type( lt_bp_raw[ vendor = ls_suppl_for_eval-
lifnr data_type = zprcc_tp22_data_type-03 ]-text ).
      ENDIF.
      APPEND ls_finance_values TO lt_finance_values.

      ls_suppliers_score-supplier = ls_suppl_for_eval-lifnr.
      APPEND ls_suppliers_score TO lt_suppliers_score.
    ENDLOOP.

    ls_finance_val_type-current_ratio = abap_true.
    ls_finance_val_type-creit_rating  = abap_false.
    ls_finance_val_type-payment_time  = abap_true.

    fill_evaluation_values(
      EXPORTING
        it_values_raw = lt_finance_values
        iv_key_fields_no = 1
        is_env_mng_values_param_types = ls_finance_val_type
      IMPORTING
        et_suppl_ev = lt_suppl_ev
    ).


    TRY .
        get_weights_structure(
          EXPORTING
            "iv_table_name_w = 'ZTB_ENV_MNG_W'
            iv_table_name_w = mt_subcriteria_def[ tkrit = iv_xtkrit ]-
zz_weight_table
          IMPORTING
```

```
            es_weights = ls_env_mng_w
        ).
      CATCH zcx_msg INTO DATA(lo_msg).
        RAISE error.
    ENDTRY.

    fill_calculation_matrix(
      EXPORTING
        it_suppl_ev = lt_suppl_ev
        iv_param_col_no = 3
        is_weight = ls_env_mng_w
        iv_weight_key_no = 2
      IMPORTING
        et_suppl_calc = lt_suppl_calc
     ).

    fill_suppliers_score(
      EXPORTING
        it_suppl_calc      = lt_suppl_calc
      CHANGING
        ct_suppliers_score = lt_suppliers_score
    ).

    IF line_exists( lt_suppliers_score[ supplier = iv_xlifnr ] ).
      ev_xbeurt = CONV elbp-beurt( lt_suppliers_score[ supplier = iv_xlifnr ]-
evaluation ).
    ELSE.
      ev_xbeurt = 0.
    ENDIF.
  ENDMETHOD.

  METHOD define_guarantee_score.

    TYPES:
      BEGIN OF ty_guarantee_values,
        supplier       TYPE lifnr,
        eplif_garantee TYPE zde_value_type,
        response_time  TYPE zde_value_type,
      END OF ty_guarantee_values,

      BEGIN OF ty_guarantee_values_type,
        eplif_garantee TYPE abap_bool,
        response_time  TYPE abap_bool,
      END OF ty_guarantee_values_type.

    DATA:
      ls_guarantee_values       TYPE ty_guarantee_values,
      lt_guarantee_values       TYPE TABLE OF ty_guarantee_values,
      ls_guarantee_values_typre TYPE ty_guarantee_values_type,
      lt_suppl_ev               TYPE TABLE OF ty_guarantee_values,
      ls_guarantee_w            TYPE ztb_guarantee_w,
      lt_suppl_calc             TYPE TABLE OF ty_suppl_calc,
      ls_suppliers_score        TYPE ty_suppliers_score,
      lt_suppliers_score        TYPE TABLE OF ty_suppliers_score.


    SELECT h~lifnr, h~aedat, i~zz_eplif_garantee, i~zz_response_time
      FROM ekko AS h
        JOIN ekpo AS i ON h~ebeln = i~ebeln
      FOR ALL ENTRIES IN @mt_suppl_for_eval
      WHERE i~matnr    =  @ms_matnr_werks-matnr
        AND i~werks    =  @ms_matnr_werks-werks
        AND h~lifnr    =  @mt_suppl_for_eval-lifnr
        AND h~bstyp    =  @zprcc_ekko_bstyp-ReqQuot
        AND i~agdat    >= @sy-datum
      INTO TABLE @DATA(lt_quotations).

    IF sy-subrc NE 0.
      MESSAGE w008(zprc_sup_ev) INTO DATA(lv_message).
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.

    SORT lt_quotations BY lifnr aedat DESCENDING.
    DELETE ADJACENT DUPLICATES FROM lt_quotations COMPARING lifnr.

    LOOP AT lt_quotations INTO DATA(ls_quotations).
      ls_guarantee_values-supplier       = ls_quotations-lifnr.
      ls_guarantee_values-eplif_garantee = CONV zde_value_type( ls_quotations-
zz_eplif_garantee ).
      ls_guarantee_values-response_time  = CONV zde_value_type( ls_quotations-
zz_response_time ).
      APPEND ls_guarantee_values TO lt_guarantee_values.

      ls_suppliers_score-supplier = ls_quotations-lifnr.
      APPEND ls_suppliers_score TO lt_suppliers_score.
    ENDLOOP.

    ls_guarantee_values_typre-eplif_garantee = abap_true.
    ls_guarantee_values_typre-response_time = abap_false.

    fill_evaluation_values(
      EXPORTING
        it_values_raw = lt_guarantee_values
        iv_key_fields_no = 1
        is_env_mng_values_param_types = ls_guarantee_values_typre
      IMPORTING
        et_suppl_ev = lt_suppl_ev
    ).

    TRY .
        get_weights_structure(
          EXPORTING
            iv_table_name_w = mt_subcriteria_def[ tkrit = iv_xtkrit ]-
zz_weight_table
          IMPORTING
            es_weights = ls_guarantee_w
        ).
      CATCH zcx_msg INTO DATA(lo_msg).
        RAISE error.
    ENDTRY.

    fill_calculation_matrix(
      EXPORTING
        it_suppl_ev = lt_suppl_ev
        iv_param_col_no = 2
        is_weight = ls_guarantee_w
        iv_weight_key_no = 2
      IMPORTING
        et_suppl_calc = lt_suppl_calc
     ).

    fill_suppliers_score(
      EXPORTING
        it_suppl_calc      = lt_suppl_calc
      CHANGING
        ct_suppliers_score = lt_suppliers_score
    ).

    IF line_exists( lt_suppliers_score[ supplier = iv_xlifnr ] ).
      ev_xbeurt = CONV elbp-beurt( lt_suppliers_score[ supplier = iv_xlifnr ]-
evaluation ).
    ELSE.
      ev_xbeurt = 0.
    ENDIF.
  ENDMETHOD.

  METHOD define_innovations_score.

    TYPES:
      BEGIN OF ty_innovation_values,
        supplier            TYPE lifnr,
        patents_no_1_year   TYPE zde_value_type,
        patents_no_10_years TYPE zde_value_type,
      END OF ty_innovation_values,

      BEGIN OF ty_innovation_param_types,
        patents_no_1_year   TYPE abap_bool,
        patents_no_10_years TYPE abap_bool,
      END OF ty_innovation_param_types.

    DATA:
      lt_bp_raw                 TYPE TABLE OF ts_bp_data_raw,
      ls_innovation_values      TYPE ty_innovation_values,
      lt_innovation_values      TYPE TABLE OF ty_innovation_values,
      ls_innovation_param_types TYPE ty_innovation_param_types,
      lt_suppl_ev               TYPE TABLE OF ty_innovation_values,
      ls_innovation_w           TYPE ztb_innovation_w,
      lt_suppl_calc             TYPE TABLE OF ty_suppl_calc,
      ls_suppliers_score        TYPE ty_suppliers_score,
      lt_suppliers_score        TYPE TABLE OF ty_suppliers_score.

    SELECT *
      INTO CORRESPONDING FIELDS OF TABLE @lt_bp_raw
      FROM bp3100 AS bp
        JOIN but000 AS but ON bp~partner = but~partner
        JOIN cvi_vend_link AS link ON but~partner_guid = link~partner_guid
      FOR ALL ENTRIES IN @mt_suppl_for_eval
      WHERE link~vendor = @mt_suppl_for_eval-lifnr
        AND bp~addtype     = @zprcc_bp3100_add_type-92.

    IF sy-subrc NE 0.
      MESSAGE w019(zprc_sup_ev) INTO DATA(lv_message).
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.

    SORT lt_bp_raw BY vendor DESCENDING.

    IF NOT line_exists( lt_bp_raw[ vendor = iv_xlifnr ] ).
      MESSAGE e012(zprc_sup_ev) INTO lv_message WITH iv_xlifnr.      "General env
ironmental management data for the supplier is not found.
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.

    LOOP AT mt_suppl_for_eval INTO DATA(ls_suppl_for_eval).
      CHECK line_exists( lt_bp_raw[ vendor = ls_suppl_for_eval-lifnr ] ).

      ls_innovation_values-supplier = ls_suppl_for_eval-lifnr.
      ls_innovation_values-
patents_no_1_year   = CONV zde_value_type( lt_bp_raw[ vendor = ls_suppl_for_eval-
lifnr data_type = zprcc_tp22_data_type-01 ]-text ).
      ls_innovation_values-
patents_no_10_years = CONV zde_value_type( lt_bp_raw[ vendor = ls_suppl_for_eval-
lifnr data_type = zprcc_tp22_data_type-02 ]-text ).
      APPEND ls_innovation_values TO lt_innovation_values.

      ls_suppliers_score-supplier = ls_suppl_for_eval-lifnr.
      APPEND ls_suppliers_score TO lt_suppliers_score.
    ENDLOOP.

    ls_innovation_param_types-patents_no_1_year = abap_true.
    ls_innovation_param_types-patents_no_10_years = abap_true.

    fill_evaluation_values(
      EXPORTING
        it_values_raw = lt_innovation_values
        iv_key_fields_no = 1
        is_env_mng_values_param_types = ls_innovation_param_types
      IMPORTING
        et_suppl_ev = lt_suppl_ev
    ).

    TRY .
        get_weights_structure(
          EXPORTING
            iv_table_name_w = mt_subcriteria_def[ tkrit = iv_xtkrit ]-
zz_weight_table
          IMPORTING
            es_weights = ls_innovation_w
        ).
      CATCH zcx_msg INTO DATA(lo_msg).
        RAISE error.
    ENDTRY.

    fill_calculation_matrix(
      EXPORTING
        it_suppl_ev = lt_suppl_ev
        iv_param_col_no = 2
        is_weight = ls_innovation_w
        iv_weight_key_no = 2
      IMPORTING
```

```
        et_suppl_calc = lt_suppl_calc
     ).

    fill_suppliers_score(
      EXPORTING
        it_suppl_calc      = lt_suppl_calc
      CHANGING
        ct_suppliers_score = lt_suppliers_score
    ).

    IF line_exists( lt_suppliers_score[ supplier = iv_xlifnr ] ).
      ev_xbeurt = CONV elbp-beurt( lt_suppliers_score[ supplier = iv_xlifnr ]-
evaluation ).
    ELSE.
      ev_xbeurt = 0.
    ENDIF.
  ENDMETHOD.

  METHOD define_deliv_evaluation_score.

    TYPES:
      BEGIN OF ty_deliv_ev_values,
        supplier        TYPE lifnr,
        lead_time       TYPE zde_value_type,
        pct_del_on_time TYPE zde_value_type,
      END OF ty_deliv_ev_values,

      BEGIN OF ty_deliv_ev_type,
        lead_time       TYPE abap_bool,
        pct_del_on_time TYPE abap_bool,
      END OF ty_deliv_ev_type.

    DATA:
      lv_max_date        TYPE ekko-aedat,
      lv_qty             TYPE zde_value_type,
      lv_lead_time       TYPE zde_value_type,
      ls_deliv_ev_values TYPE ty_deliv_ev_values,
      lt_deliv_ev_values TYPE TABLE OF ty_deliv_ev_values,
      ls_deliv_ev_type   TYPE ty_deliv_ev_type,
      lt_suppl_ev        TYPE TABLE OF ty_deliv_ev_values,
      lt_suppl_calc      TYPE TABLE OF ty_suppl_calc,
      ls_suppliers_score TYPE ty_suppliers_score,
      lt_suppliers_score TYPE TABLE OF ty_suppliers_score.

    SELECT h~ebeln, h~lifnr, h~aedat, i~plifz, i~ktmng, e~slfdt, e~menge
      FROM ekko AS h
        JOIN ekpo AS i ON h~ebeln = i~ebeln
        JOIN eket AS e ON i~ebeln = e~ebeln AND i~ebelp = e~ebelp
      FOR ALL ENTRIES IN @mt_suppl_for_eval
      WHERE i~matnr   =  @ms_matnr_werks-matnr
        AND i~werks   =  @ms_matnr_werks-werks
        AND h~lifnr   =  @mt_suppl_for_eval-lifnr
        AND h~bstyp   =  @zprcc_ekko_bstyp-ReqQuot
        AND i~agdat   >= @sy-datum
        AND i~loekz   = ''
      INTO TABLE @DATA(lt_quotations).

    IF sy-subrc NE 0.
      MESSAGE w008(zprc_sup_ev) INTO DATA(lv_message).
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.

    SORT lt_quotations BY lifnr ASCENDING
                          aedat DESCENDING
                          slfdt ASCENDING.

    DELETE ADJACENT DUPLICATES FROM lt_quotations COMPARING lifnr.

    LOOP AT lt_quotations INTO DATA(ls_quotation).
      ls_deliv_ev_values-supplier = ls_quotation-lifnr.
      ls_deliv_ev_values-lead_time = ls_quotation-plifz.
      TRY .
          ls_deliv_ev_values-pct_del_on_time = ls_quotation-menge / ls_quotation-
ktmng * 100.
        CATCH cx_sy_zerodivide.
          ls_deliv_ev_values-pct_del_on_time = 0.
      ENDTRY.
      APPEND ls_deliv_ev_values TO lt_deliv_ev_values.

      ls_suppliers_score-supplier = ls_quotation-lifnr.
      APPEND ls_suppliers_score TO lt_suppliers_score.
    ENDLOOP.

    "check whether a supplier quotation exist
    IF NOT line_exists( lt_quotations[ lifnr = iv_xlifnr ] ).
      MESSAGE e009(zprc_sup_ev) INTO lv_message WITH iv_xlifnr.       "Quatations
for the supplier are not found.
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.

    ls_deliv_ev_type-lead_time = abap_false.
    ls_deliv_ev_type-pct_del_on_time = abap_true.

    fill_evaluation_values(
      EXPORTING
        it_values_raw = lt_deliv_ev_values
        iv_key_fields_no = 1
        is_env_mng_values_param_types = ls_deliv_ev_type
      IMPORTING
        et_suppl_ev = lt_suppl_ev
    ).

    fill_calculation_matrix(
      EXPORTING
        it_suppl_ev = lt_suppl_ev
        iv_param_col_no = 2
        "is_weight = ls_env_mng_w
        "iv_weight_key_no = 2
      IMPORTING
        et_suppl_calc = lt_suppl_calc
     ).

    fill_suppliers_score(
      EXPORTING
```

```
        it_suppl_calc      = lt_suppl_calc
      CHANGING
        ct_suppliers_score = lt_suppliers_score
    ).

    IF line_exists( lt_suppliers_score[ supplier = iv_xlifnr ] ).
      ev_xbeurt = CONV elbp-beurt( lt_suppliers_score[ supplier = iv_xlifnr ]-
evaluation ).
    ELSE.
      ev_xbeurt = 0.
    ENDIF.
  ENDMETHOD.

  METHOD define_batch_flexibility_score.

    TYPES:
      BEGIN OF ty_batch_flex_values,
        supplier       TYPE lifnr,
        charg_min_size TYPE zde_value_type,
        charg_max_size TYPE zde_value_type,
      END OF ty_batch_flex_values,

      BEGIN OF ty_batch_flex_type,
        charg_min_size TYPE abap_bool,
        charg_max_size TYPE abap_bool,
      END OF ty_batch_flex_type.

    DATA:
      lv_max_date          TYPE ekko-aedat,
      lv_qty               TYPE zde_value_type,
      lv_lead_time         TYPE zde_value_type,
      ls_batch_flex_values TYPE ty_batch_flex_values,
      lt_batch_flex_values TYPE TABLE OF ty_batch_flex_values,
      ls_batch_flex_type   TYPE ty_batch_flex_type,
      lt_suppl_ev          TYPE TABLE OF ty_batch_flex_values,
      lt_suppl_calc        TYPE TABLE OF ty_suppl_calc,
      ls_suppliers_score   TYPE ty_suppliers_score,
      lt_suppliers_score   TYPE TABLE OF ty_suppliers_score.

    SELECT h~ebeln, h~lifnr, h~aedat, i~zz_charg_min_size, i~zz_charg_max_size
      FROM ekko AS h
        JOIN ekpo AS i ON h~ebeln = i~ebeln
      FOR ALL ENTRIES IN @mt_suppl_for_eval
      WHERE i~matnr    =  @ms_matnr_werks-matnr
        AND i~werks    =  @ms_matnr_werks-werks
        AND h~lifnr    =  @mt_suppl_for_eval-lifnr
        AND h~bstyp    =  @zprcc_ekko_bstyp-ReqQuot
        AND i~agdat    >= @sy-datum
        AND i~loekz    = ''
      INTO TABLE @DATA(lt_quotations).

    IF sy-subrc NE 0.
      MESSAGE w008(zprc_sup_ev) INTO DATA(lv_message).
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.

    SORT lt_quotations BY lifnr aedat DESCENDING.
    DELETE ADJACENT DUPLICATES FROM lt_quotations COMPARING lifnr.

    "check whether a supplier quotation exist
    IF NOT line_exists( lt_quotations[ lifnr = iv_xlifnr ] ).
      MESSAGE e009(zprc_sup_ev) INTO lv_message WITH iv_xlifnr.       "Quatations
for the supplier are not found.
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.

    LOOP AT lt_quotations INTO DATA(ls_quotation).
      ls_batch_flex_values-supplier = ls_quotation-lifnr.
      ls_batch_flex_values-charg_min_size = ls_quotation-zz_charg_min_size.
      ls_batch_flex_values-charg_max_size = ls_quotation-zz_charg_max_size.
      APPEND ls_batch_flex_values TO lt_batch_flex_values.

      ls_suppliers_score-supplier = ls_quotation-lifnr.
      APPEND ls_suppliers_score TO lt_suppliers_score.
    ENDLOOP.

    ls_batch_flex_type-charg_min_size = abap_false.
    ls_batch_flex_type-charg_max_size = abap_true.

    fill_evaluation_values(
      EXPORTING
        it_values_raw = lt_batch_flex_values
        iv_key_fields_no = 1
        is_env_mng_values_param_types = ls_batch_flex_type
      IMPORTING
        et_suppl_ev = lt_suppl_ev
    ).

    fill_calculation_matrix(
      EXPORTING
        it_suppl_ev = lt_suppl_ev
        iv_param_col_no = 2
        "is_weight = ls_env_mng_w
        "iv_weight_key_no = 2
      IMPORTING
        et_suppl_calc = lt_suppl_calc
     ).

    fill_suppliers_score(
      EXPORTING
        it_suppl_calc      = lt_suppl_calc
      CHANGING
        ct_suppliers_score = lt_suppliers_score
    ).

    IF line_exists( lt_suppliers_score[ supplier = iv_xlifnr ] ).
      ev_xbeurt = CONV elbp-beurt( lt_suppliers_score[ supplier = iv_xlifnr ]-
evaluation ).
    ELSE.
      ev_xbeurt = 0.
    ENDIF.
  ENDMETHOD.
  METHOD define_supplier_adapt_score.
```

```
TYPES:
  BEGIN OF ty_adaptivity_values,
    supplier        TYPE lifnr,
    charg_spec_size TYPE zde_value_type,
  END OF ty_adaptivity_values,

  BEGIN OF ty_adaptivity_type,
    charg_spec_size TYPE abap_bool,
  END OF ty_adaptivity_type.

DATA:
  ls_adaptivity_values TYPE ty_adaptivity_values,
  lt_adaptivity_values TYPE TABLE OF ty_adaptivity_values,
  ls_adaptivity_type   TYPE ty_adaptivity_type,
  lt_suppl_ev          TYPE TABLE OF ty_adaptivity_values,
  lt_suppl_calc        TYPE TABLE OF ty_suppl_calc,
  ls_suppliers_score   TYPE ty_suppliers_score,
  lt_suppliers_score   TYPE TABLE OF ty_suppliers_score.


SELECT h~ebeln, h~lifnr, h~aedat, i~zz_charg_spec_size
  FROM ekko AS h
    JOIN ekpo AS i ON h~ebeln = i~ebeln
  FOR ALL ENTRIES IN @mt_suppl_for_eval
  WHERE i~matnr   =  @ms_matnr_werks-matnr
    AND i~werks   =  @ms_matnr_werks-werks
    AND h~lifnr   =  @mt_suppl_for_eval-lifnr
    AND h~bstyp   =  @zprcc_ekko_bstyp-ReqQuot
    AND i~agdat   >= @sy-datum
    AND i~loekz   = ''
  INTO TABLE @DATA(lt_quotations).

IF sy-subrc NE 0.
  MESSAGE w008(zprc_sup_ev) INTO DATA(lv_message).
  add_message_to_log( ).
  ev_xbeurt = 0.
  RETURN.
ENDIF.

SORT lt_quotations BY lifnr aedat DESCENDING.
DELETE ADJACENT DUPLICATES FROM lt_quotations COMPARING lifnr.

"check whether a supplier quotation exist
IF NOT line_exists( lt_quotations[ lifnr = iv_xlifnr ] ).
  MESSAGE e009(zprc_sup_ev) INTO lv_message WITH iv_xlifnr.       "Quatations
for the supplier are not found.
  add_message_to_log( ).
  ev_xbeurt = 0.
  RETURN.
ENDIF.

LOOP AT lt_quotations INTO DATA(ls_quotation).
  ls_adaptivity_values-supplier = ls_quotation-lifnr.
  ls_adaptivity_values-charg_spec_size = ls_quotation-zz_charg_spec_size.
  APPEND ls_adaptivity_values TO lt_adaptivity_values.

  ls_suppliers_score-supplier = ls_quotation-lifnr.
  APPEND ls_suppliers_score TO lt_suppliers_score.
ENDLOOP.
```

```
    ls_adaptivity_type-charg_spec_size = abap_true.

    fill_evaluation_values(
      EXPORTING
        it_values_raw = lt_adaptivity_values
        iv_key_fields_no = 1
        is_env_mng_values_param_types = ls_adaptivity_type
      IMPORTING
        et_suppl_ev = lt_suppl_ev
    ).

    fill_calculation_matrix(
      EXPORTING
        it_suppl_ev = lt_suppl_ev
        iv_param_col_no = 1
      IMPORTING
        et_suppl_calc = lt_suppl_calc
     ).

    fill_suppliers_score(
      EXPORTING
        it_suppl_calc      = lt_suppl_calc
      CHANGING
        ct_suppliers_score = lt_suppliers_score
    ).

    IF line_exists( lt_suppliers_score[ supplier = iv_xlifnr ] ).
      ev_xbeurt = CONV elbp-beurt( lt_suppliers_score[ supplier = iv_xlifnr ]-
evaluation ).
    ELSE.
      ev_xbeurt = 0.
    ENDIF.
  ENDMETHOD.

  METHOD define_pollution_control_score.

    TYPES:
      BEGIN OF ty_polut_contr_type,
        carbon_footprint TYPE abap_bool,
        waste_water      TYPE abap_bool,
        solid_wastes     TYPE abap_bool,
        energy_consump   TYPE abap_bool,
        harmful_mat      TYPE abap_bool,
        evaluation       TYPE abap_bool,
      END OF ty_polut_contr_type.

    DATA:
      lv_message             TYPE string,
      ls_polut_contr_values TYPE ty_suppl_ev,
      lt_polut_contr_values TYPE TABLE OF ty_suppl_ev,
      lt_suppl_ev            TYPE TABLE OF ty_suppl_ev,
      lt_suppl_calc          TYPE TABLE OF ty_suppl_calc,
      ls_suppliers_score     TYPE ty_suppliers_score,
      lt_suppliers_score     TYPE TABLE OF ty_suppliers_score.


    "find appropriate Quotations
    SELECT h~ebeln, h~lifnr, h~aedat, i~zz_carbon_footprint, i~zz_waste_water, i~z
```

```
z_solid_wastes, i~zz_energy_consump, i~zz_harmful_mat
      FROM ekko AS h
        JOIN ekpo AS i ON h~ebeln = i~ebeln
      FOR ALL ENTRIES IN @mt_suppl_for_eval
      WHERE i~matnr    =  @ms_matnr_werks-matnr
        AND i~werks    =  @ms_matnr_werks-werks
        AND h~lifnr    =  @mt_suppl_for_eval-lifnr
        AND h~bstyp    =  @zprcc_ekko_bstyp-ReqQuot
        AND i~agdat    >= @sy-datum
        AND i~loekz    =  ''
      INTO TABLE @DATA(lt_quotations).

    IF sy-subrc NE 0.
      MESSAGE w008(zprc_sup_ev) INTO lv_message.
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.

    SORT lt_quotations BY lifnr aedat DESCENDING.
    DELETE ADJACENT DUPLICATES FROM lt_quotations COMPARING lifnr.

    "check whether a supplier quotation exist
    IF NOT line_exists( lt_quotations[ lifnr = iv_xlifnr ] ).
      MESSAGE e009(zprc_sup_ev) INTO lv_message WITH iv_xlifnr.        "Quatations
for the supplier are not found.
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.

    LOOP AT lt_quotations INTO DATA(ls_quotation).
      ls_polut_contr_values-supplier            = ls_quotation-lifnr.
      ls_polut_contr_values-carbon_footprint = ls_quotation-zz_carbon_footprint.
      ls_polut_contr_values-waste_water      = ls_quotation-zz_waste_water.
      ls_polut_contr_values-solid_wastes     = ls_quotation-zz_solid_wastes.
      ls_polut_contr_values-energy_consump   = ls_quotation-zz_energy_consump.
      ls_polut_contr_values-harmful_mat      = ls_quotation-zz_harmful_mat.
      APPEND ls_polut_contr_values TO lt_polut_contr_values.

      ls_suppliers_score-supplier = ls_quotation-lifnr.
      APPEND ls_suppliers_score TO lt_suppliers_score.
    ENDLOOP.

    fill_evaluation_values(
      EXPORTING
        it_values_raw = lt_polut_contr_values
        iv_key_fields_no = 1
        "is_env_mng_values_param_types = ls_batch_flex_type
      IMPORTING
        et_suppl_ev = lt_suppl_ev
    ).

    fill_calculation_matrix(
      EXPORTING
        it_suppl_ev = lt_suppl_ev
        iv_param_col_no = 5
        "is_weight = ls_env_mng_w
        "iv_weight_key_no = 2
      IMPORTING
        et_suppl_calc = lt_suppl_calc
     ).

    fill_suppliers_score(
      EXPORTING
        it_suppl_calc      = lt_suppl_calc
      CHANGING
        ct_suppliers_score = lt_suppliers_score
    ).

    IF line_exists( lt_suppliers_score[ supplier = iv_xlifnr ] ).
      ev_xbeurt = CONV elbp-beurt( lt_suppliers_score[ supplier = iv_xlifnr ]-
evaluation ).
    ELSE.
      ev_xbeurt = 0.
    ENDIF.
  ENDMETHOD.


  METHOD define_env_mng_prod_lvl_score.
    TYPES:
      BEGIN OF ty_suppl_ev_99_02,
        supplier         TYPE lifnr,
        cost_comp_dispos TYPE zde_value_type,
        green_pack       TYPE zde_value_type,
      END OF ty_suppl_ev_99_02,

      BEGIN OF ty_weight_99_02,
        cost_comp_dispos_w TYPE zde_value_type,
        green_pack_w       TYPE zde_value_type,
      END OF ty_weight_99_02.

    DATA:
      lv_message         TYPE string,
      lv_alpha           TYPE zde_value_type,
      lv_betta           TYPE zde_value_type,
      lv_value           TYPE zde_value_type,
      lt_values          TYPE zde_value_type_table,
      ls_suppl_ev        TYPE ty_suppl_ev_99_02,
      lt_suppl_ev        TYPE TABLE OF ty_suppl_ev_99_02,
      ls_weight_99_02    TYPE ty_weight_99_02,
      ls_suppliers_score TYPE ty_suppliers_score,
      lt_suppliers_score TYPE TABLE OF ty_suppliers_score.


    DATA:
      ls_suppl_calc TYPE ty_suppl_calc,
      lt_suppl_calc TYPE TABLE OF ty_suppl_calc.

    FIELD-SYMBOLS:
      <main> TYPE zde_value_type,
      <sub>  TYPE zde_value_type.


    SELECT h~lifnr, h~aedat, i~zz_cost_comp_dispos, i~zz_green_pack
      FROM ekko AS h
        JOIN ekpo AS i ON h~ebeln = i~ebeln
      FOR ALL ENTRIES IN @mt_suppl_for_eval
      WHERE i~matnr   =  @ms_matnr_werks-matnr
        AND i~werks   =  @ms_matnr_werks-werks
        AND h~lifnr   =  @mt_suppl_for_eval-lifnr
        AND h~bstyp   =  @zprcc_ekko_bstyp-ReqQuot
        AND i~agdat   >= @sy-datum
      INTO TABLE @DATA(lt_quotations).

    IF sy-subrc NE 0.
      MESSAGE w008(zprc_sup_ev) INTO lv_message.
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.

    SORT lt_quotations BY lifnr aedat DESCENDING.
    DELETE ADJACENT DUPLICATES FROM lt_quotations COMPARING lifnr.

    "check whether a supplier quotation exist
    IF NOT line_exists( lt_quotations[ lifnr = iv_xlifnr ] ).
      MESSAGE e009(zprc_sup_ev) INTO lv_message WITH iv_xlifnr.       "Quatations for the sup
plier are not found.
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.

    TRY.
        zcl_tvarvc=>read_param(
          EXPORTING
            iv_var_name = CONV #( zprcc_sup_eval_99_02_alpha )
          IMPORTING
            ev_value    = ls_weight_99_02-cost_comp_dispos_w ).

        zcl_tvarvc=>read_param(
          EXPORTING
             iv_var_name = CONV #( zprcc_sup_eval_99_02_betta )
          IMPORTING
            ev_value    = ls_weight_99_02-green_pack_w ).

      CATCH zcx_msg.
        MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
          WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4 INTO lv_message.
        add_message_to_log( ).
        ev_xbeurt = 0.
        RETURN.
    ENDTRY.

    CALL METHOD me->calculate_column_nubmers
      EXPORTING
        it_any_table         = lt_quotations
      IMPORTING
        ev_number_of_columns = DATA(lv_number_of_columns).

    LOOP AT lt_quotations INTO DATA(ls_quotation).
      CLEAR lv_value.
      lv_value = CONV #( ls_quotation-zz_cost_comp_dispos ).
      APPEND lv_value TO lt_values.

      ls_suppliers_score-supplier = ls_quotation-lifnr.
      APPEND ls_suppliers_score TO lt_suppliers_score.
    ENDLOOP.


    "LOOP AT lt_quotations INTO ls_quotation.
    LOOP AT lt_quotations INTO ls_quotation.

      IF line_exists( mt_suppl_for_eval[ lifnr = ls_quotation-lifnr ] ).

        ls_suppl_ev-supplier = ls_quotation-lifnr.

        get_value_score(
          EXPORTING
            iv_value = CONV #( ls_quotation-zz_cost_comp_dispos )
            it_values = CONV #( lt_values )
          IMPORTING
            ev_score = DATA(lv_score) ).

        ls_suppl_ev-cost_comp_dispos = CONV #( lv_score ).
        ls_suppl_ev-green_pack = COND #( WHEN ls_quotation-zz_green_pack  =  abap_true  THEN  1
ELSE 0 ).
        APPEND ls_suppl_ev TO lt_suppl_ev.

      ENDIF.
    ENDLOOP.

    fill_calculation_matrix(
      EXPORTING
        it_suppl_ev = lt_suppl_ev
        iv_param_col_no = 2
        is_weight = ls_weight_99_02
      IMPORTING
        et_suppl_calc = lt_suppl_calc
     ).

    fill_suppliers_score(
      EXPORTING
        it_suppl_calc      = lt_suppl_calc
      CHANGING
        ct_suppliers_score = lt_suppliers_score
    ).

    IF line_exists( lt_suppliers_score[ supplier = iv_xlifnr ] ).
      ev_xbeurt = CONV elbp-beurt( lt_suppliers_score[ supplier = iv_xlifnr ]-evaluation ).
    ELSE.
      ev_xbeurt = 0.
    ENDIF.
  ENDMETHOD.


  METHOD define_env_mng_sup_lvl_score.

    TYPES:
      BEGIN OF ty_env_mng_values,
        supplier         TYPE lifnr,
        pct_green_cust   TYPE zde_value_type,
        eco_low_compl_no TYPE zde_value_type,
        pct_prof_on_env  TYPE zde_value_type,
        corp_social_resp TYPE zde_value_type,
        green_proc_plan  TYPE zde_value_type,
        alt_imp_nat_res  TYPE zde_value_type,
      END OF ty_env_mng_values,

      BEGIN OF ty_param_types,
        pct_green_cust   TYPE abap_bool,
        eco_low_compl_no TYPE abap_bool,
        pct_prof_on_env  TYPE abap_bool,
        corp_social_resp TYPE abap_bool,
        green_proc_plan  TYPE abap_bool,
        alt_imp_nat_res  TYPE abap_bool,
      END OF ty_param_types.

    DATA:
      lv_message                    TYPE string,
      lt_env_mng_values_raw         TYPE TABLE OF ts_bp_data_raw,
      ls_env_mng_values_param_types TYPE ty_param_types,
      ls_env_mng_values             TYPE ty_env_mng_values,
      lt_env_mng_values             TYPE TABLE OF ty_env_mng_values,
      ls_env_mng_w                  TYPE ztb_env_mng_w,
      lt_suppl_ev                   TYPE TABLE OF ty_env_mng_values,
      lt_suppl_calc                 TYPE TABLE OF ty_suppl_calc,
      ls_suppliers_score            TYPE ty_suppliers_score,
      lt_suppliers_score            TYPE TABLE OF ty_suppliers_score.

    SELECT *
      INTO CORRESPONDING FIELDS OF TABLE @lt_env_mng_values_raw
      FROM bp3100 AS bp
        JOIN but000 AS but ON bp~partner = but~partner
        JOIN cvi_vend_link AS link ON but~partner_guid = link~partner_guid
      FOR ALL ENTRIES IN @mt_suppl_for_eval
      WHERE link~vendor = @mt_suppl_for_eval-lifnr
        AND bp~addtype     = @zprcc_bp3100_add_type-94.

    IF sy-subrc NE 0.
      MESSAGE w011(zprc_sup_ev) INTO lv_message.
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.

    SORT lt_env_mng_values_raw BY vendor DESCENDING.
    "DELETE ADJACENT DUPLICATES FROM lt_env_mng_values_raw COMPARING vendor.

    "check whether a supplier quotation exist
    IF NOT line_exists( lt_env_mng_values_raw[ vendor = iv_xlifnr ] ).
      MESSAGE e012(zprc_sup_ev) INTO lv_message WITH iv_xlifnr.       "General env
ironmental management data for the supplier is not found.
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.

    LOOP AT mt_suppl_for_eval INTO DATA(ls_suppl_for_eval).
      CHECK line_exists( lt_env_mng_values_raw[ vendor = ls_suppl_for_eval-
lifnr ] ).
      ls_env_mng_values-supplier = ls_suppl_for_eval-lifnr.
      ls_env_mng_values-
pct_green_cust   = CONV #( lt_env_mng_values_raw[ vendor = ls_suppl_for_eval-
lifnr data_type = '01' ]-text ).
      ls_env_mng_values-
eco_low_compl_no = CONV #( lt_env_mng_values_raw[ vendor = ls_suppl_for_eval-
lifnr data_type = '02' ]-text ).
      ls_env_mng_values-
pct_prof_on_env  = CONV #( lt_env_mng_values_raw[ vendor = ls_suppl_for_eval-
lifnr data_type = '03' ]-text ).
      ls_env_mng_values-
corp_social_resp = CONV #( COND #( WHEN lt_env_mng_values_raw[ vendor = ls_suppl_f
or_eval-lifnr data_type = '04' ]-xfeld  = abap_true THEN 1 ELSE 0 ) ).
      ls_env_mng_values-
green_proc_plan  = CONV #( COND #( WHEN lt_env_mng_values_raw[ vendor = ls_suppl_f
or_eval-lifnr data_type = '05' ]-xfeld  = abap_true THEN 1 ELSE 0 ) ).
      ls_env_mng_values-
alt_imp_nat_res  = CONV #( COND #( WHEN lt_env_mng_values_raw[ vendor = ls_suppl_f
or_eval-lifnr data_type = '06' ]-xfeld  = abap_true THEN 1 ELSE 0 ) ).
      APPEND ls_env_mng_values TO lt_env_mng_values.

      ls_suppliers_score-supplier = ls_suppl_for_eval-lifnr.
      APPEND ls_suppliers_score TO lt_suppliers_score.

    ENDLOOP.

    ls_env_mng_values_param_types-pct_green_cust   = abap_true.
    ls_env_mng_values_param_types-eco_low_compl_no = abap_false.
    ls_env_mng_values_param_types-pct_prof_on_env  = abap_true.
    ls_env_mng_values_param_types-corp_social_resp = abap_true.
    ls_env_mng_values_param_types-green_proc_plan  = abap_true.
    ls_env_mng_values_param_types-alt_imp_nat_res  = abap_true.

    fill_evaluation_values(
      EXPORTING
        it_values_raw = lt_env_mng_values
        iv_key_fields_no = 1
        is_env_mng_values_param_types = ls_env_mng_values_param_types
      IMPORTING
        et_suppl_ev = lt_suppl_ev
    ).

    TRY .
        get_weights_structure(
          EXPORTING
            iv_table_name_w = mt_subcriteria_def[ tkrit = iv_xtkrit ]-
zz_weight_table
          IMPORTING
            es_weights = ls_env_mng_w
        ).
      CATCH zcx_msg INTO DATA(lo_msg).
        RAISE error.
    ENDTRY.

    fill_calculation_matrix(
      EXPORTING
        it_suppl_ev = lt_suppl_ev
        iv_param_col_no = 6
        is_weight = ls_env_mng_w
        iv_weight_key_no = 2
      IMPORTING
        et_suppl_calc = lt_suppl_calc
```

```
     ).

    fill_suppliers_score(
      EXPORTING
        it_suppl_calc      = lt_suppl_calc
      CHANGING
        ct_suppliers_score = lt_suppliers_score
    ).

    IF line_exists( lt_suppliers_score[ supplier = iv_xlifnr ] ).
      ev_xbeurt = CONV elbp-beurt( lt_suppliers_score[ supplier = iv_xlifnr ]-
evaluation ).
    ELSE.
      ev_xbeurt = 0.
    ENDIF.
  ENDMETHOD.


  METHOD define_green_competence_score.

    TYPES:
      BEGIN OF ty_iso_cert_values,
        supplier         TYPE lifnr,
        pct_green_cust   TYPE zde_value_type,
        eco_low_compl_no TYPE zde_value_type,
        pct_prof_on_env  TYPE zde_value_type,
        corp_social_resp TYPE zde_value_type,
        green_proc_plan  TYPE zde_value_type,
        alt_imp_nat_res  TYPE zde_value_type,
      END OF ty_iso_cert_values.

    DATA:
      lv_message         TYPE string,
      lt_bp_data_raw     TYPE TABLE OF ts_bp_data_raw,
      "ls_iso_cert_values TYPE ty_env_mng_values,
      lt_iso_cert_values TYPE lvc_t_fcat,
      lt_suppl_calc      TYPE TABLE OF ty_suppl_calc,
      ls_suppliers_score TYPE ty_suppliers_score,
      lt_suppliers_score TYPE TABLE OF ty_suppliers_score.

    FIELD-SYMBOLS:
       <gfs_dyn_table> TYPE STANDARD TABLE.


    SELECT *
      INTO CORRESPONDING FIELDS OF TABLE @lt_bp_data_raw
      FROM bp3100 AS bp
        JOIN but000 AS but ON bp~partner = but~partner
        JOIN cvi_vend_link AS link ON but~partner_guid = link~partner_guid
      FOR ALL ENTRIES IN @mt_suppl_for_eval
      WHERE link~vendor = @mt_suppl_for_eval-lifnr
        AND bp~addtype     = @zprcc_bp3100_add_type-95.

    IF sy-subrc NE 0.
      MESSAGE w014(zprc_sup_ev) INTO lv_message.
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.
    SORT lt_bp_data_raw BY vendor DESCENDING.

    IF NOT line_exists( lt_bp_data_raw[ vendor = iv_xlifnr ] ).
      MESSAGE e015(zprc_sup_ev) INTO lv_message WITH iv_xlifnr.       "General env
ironmental management data for the supplier is not found.
      add_message_to_log( ).
      ev_xbeurt = 0.
      RETURN.
    ENDIF.

    LOOP AT mt_suppl_for_eval INTO DATA(ls_suppl_for_eval).
      CHECK line_exists( lt_bp_data_raw[ vendor = ls_suppl_for_eval-lifnr ] ).
      ls_suppliers_score-supplier = ls_suppl_for_eval-lifnr.
      APPEND ls_suppliers_score TO lt_suppliers_score.
    ENDLOOP.

    "----------------------------------------------------------------------------

    create_dynamic_table(
      EXPORTING
        iv_addtype = zprcc_bp3100_add_type-95
      IMPORTING
        et_dynamic_table = gt_dyn_table
     ).

    ASSIGN gt_dyn_table->* TO <gfs_dyn_table>.

    create_certification_table(
      EXPORTING
        iv_addtype     = zprcc_bp3100_add_type-95
      IMPORTING
        et_certif_data = <gfs_dyn_table>
      CHANGING
        ct_bp_data_raw = lt_bp_data_raw
     ).

    fill_calculation_matrix(
      EXPORTING
        it_suppl_ev = <gfs_dyn_table>
        iv_param_col_no = 10
        "is_weight = ls_env_mng_w
        "iv_weight_key_no = 2
      IMPORTING
        et_suppl_calc = lt_suppl_calc
     ).

    fill_suppliers_score(
      EXPORTING
        it_suppl_calc      = lt_suppl_calc
      CHANGING
        ct_suppliers_score = lt_suppliers_score
    ).

    IF line_exists( lt_suppliers_score[ supplier = iv_xlifnr ] ).
      ev_xbeurt = CONV elbp-beurt( lt_suppliers_score[ supplier = iv_xlifnr ]-
evaluation ).
    ELSE.
      ev_xbeurt = 0.
    ENDIF.
  ENDMETHOD.


  METHOD constructor.
    DATA: lv_message TYPE string.

    super->constructor(
      EXPORTING
        iv_xlifnr = iv_xlifnr
        iv_xekorg = iv_xekorg
        iv_xhkrit = iv_xhkrit
        iv_xtkrit = iv_xtkrit
        iv_xkrtyp = iv_xkrtyp
      EXCEPTIONS
        error     = 1
        OTHERS    = 2
    ).

    IF sy-subrc NE 0.
      IF sy-msgid IS INITIAL.
        MESSAGE e002(zprc_sup_ev) INTO lv_message.
      ENDIF.
      RAISE error.
    ENDIF.

    init(
      EXCEPTIONS
        error  = 1
        OTHERS = 2  ).

    IF sy-subrc NE 0.
      IF sy-msgid IS INITIAL.
        MESSAGE e002(zprc_sup_ev) INTO lv_message.
      ENDIF.
      RAISE error.
    ENDIF.
  ENDMETHOD.


  METHOD calculate_column_nubmers.
    DATA: lt_mara         TYPE TABLE OF mara,
          lo_table_descr  TYPE REF TO cl_abap_tabledescr,
          lo_struct_descr TYPE REF TO cl_abap_structdescr.

    TRY.
*     Use RTTI services to describe table variable
        lo_table_descr ?= cl_abap_tabledescr=>describe_by_data( p_data = it_any_ta
ble ).
*     Use RTTI services to describe table structure
        lo_struct_descr ?= lo_table_descr->get_table_line_type( ).

*     Count number of columns in structure
        ev_number_of_columns = lines( lo_struct_descr->components ).

      CATCH cx_sy_move_cast_error.
        ev_number_of_columns = 0.
    ENDTRY.

  ENDMETHOD.
```

```
METHOD get_purchasing_material.
  DATA:
    lv_message TYPE string.

  SELECT SINGLE matnr, werks FROM ztsupev_marc
    INTO CORRESPONDING FIELDS OF @ms_matnr_werks
    WHERE ekorg = @ms_criteria_def-ekorg
      AND zz_mat_to_eval = @abap_true.

  IF sy-subrc NE 0.
    MESSAGE e007(zprc_sup_ev) WITH ms_criteria_def-ekorg INTO lv_message.
    RAISE EXCEPTION TYPE zcx_msg EXPORTING ms_syst = sy.
  ENDIF.
ENDMETHOD.


METHOD get_value_score.
  CHECK it_values IS NOT INITIAL.

  DATA(lt_values) = it_values.
  SORT lt_values DESCENDING.
  DATA(lv_max_value) = lt_values[ 1 ].

  DATA(lv_min_value) = lt_values[ lines( it_values ) ].

  IF iv_is_beneficial EQ abap_false.
    TRY .
        ev_score = ( lv_max_value - iv_value ) / ( lv_max_value - lv_min_value )
      CATCH cx_sy_zerodivide.
        ev_score = 0.
    ENDTRY.
  ELSE.
    TRY .
        ev_score = ( iv_value - lv_min_value ) / ( lv_max_value - lv_min_value )
      CATCH cx_sy_zerodivide.
        ev_score = 0.
    ENDTRY.
  ENDIF.
ENDMETHOD.


METHOD fill_calculation_matrix.

  DATA:
    ls_suppl_calc TYPE ty_suppl_calc.

  FIELD-SYMBOLS:
    <main>          TYPE zde_value_type,
    <sub>           TYPE zde_value_type,
    <weight>        TYPE zde_value_type,
    <supplier_main> TYPE any,
    <supplier_sub>  TYPE any.

  DATA:
    lt_components TYPE abap_compdescr_tab,
    lv_weight_sum TYPE zde_value_type.

  DATA:
      l_ref   TYPE REF TO cl_abap_tabledescr,
      l_dref  TYPE REF TO cl_abap_structdescr,
      text    TYPE string,
      lo_cast TYPE REF TO cx_sy_move_cast_error.
    DATA wa TYPE abap_compdescr.

    TRY.
        l_ref ?= cl_abap_typedescr=>describe_by_data( it_suppl_ev ).
        l_dref ?= l_ref->get_table_line_type( ).
        lt_components = l_dref->components.
      CATCH cx_sy_move_cast_error INTO lo_cast.
        text = lo_cast->get_text( ).
    ENDTRY.

    "-----------------------------------------------------------------------------
---------------

    TYPE-POOLS: abap.

    DATA: ls_components TYPE abap_compdescr.
    DATA: lo_strucdescr TYPE REF TO cl_abap_structdescr.
    DATA: lt_weight_components TYPE abap_compdescr_tab.

    IF is_weight IS NOT INITIAL.
      lo_strucdescr ?= cl_abap_typedescr=>describe_by_data( is_weight ).
      lt_weight_components = lo_strucdescr->components.
    ENDIF.

    LOOP AT lt_weight_components INTO DATA(ls_weight_component).
      CHECK sy-tabix > iv_weight_key_no.
      ASSIGN COMPONENT ls_weight_component-
name OF STRUCTURE is_weight TO <weight>.
      lv_weight_sum = lv_weight_sum + <weight>.
    ENDLOOP.
    UNASSIGN <weight>.
    "-----------------------------------------------------------------------------
---------------

    LOOP AT it_suppl_ev ASSIGNING FIELD-SYMBOL(<wa_main_sup>).
      LOOP AT it_suppl_ev ASSIGNING FIELD-SYMBOL(<wa_sub_sup>).
        ASSIGN COMPONENT 'SUPPLIER' OF STRUCTURE <wa_main_sup> TO <supplier_main>.
        ASSIGN COMPONENT 'SUPPLIER' OF STRUCTURE <wa_sub_sup> TO <supplier_sub>.
        IF <supplier_main> = <supplier_sub>.
          CONTINUE.
        ENDIF.
        ls_suppl_calc-main_supplier = <supplier_main>.
        ls_suppl_calc-sub_supplier  = <supplier_sub>.
        ls_suppl_calc-calculation   = 0.

        DO iv_param_col_no TIMES.
          DATA(lv_comp_name) = lt_components[ sy-index + 1 ]-name.
          UNASSIGN <main>.
          UNASSIGN <sub>.
          ASSIGN COMPONENT lv_comp_name OF STRUCTURE <wa_main_sup> TO <main>.
          ASSIGN COMPONENT lv_comp_name OF STRUCTURE <wa_sub_sup> TO <sub>.
          DATA(lv_calculation) = COND zde_value_type( WHEN <main> - <sub> > 0 THEN
 <main> - <sub> ).
          IF is_weight IS NOT INITIAL.
            DATA(lv_comp_name_w) = lt_weight_components[ sy-
index + iv_weight_key_no ]-name.
            ASSIGN COMPONENT lv_comp_name_w OF STRUCTURE is_weight TO <weight>.
            lv_calculation = lv_calculation * <weight> / lv_weight_sum.
          ELSE.
            lv_calculation = CONV #( lv_calculation / iv_param_col_no ).
          ENDIF.
          "ls_suppl_calc-calculation = ls_suppl_calc-
calculation + COND zde_value_type( WHEN <main> - <sub> > 0 THEN <main> - <sub> ).
          ls_suppl_calc-calculation = ls_suppl_calc-calculation + lv_calculation.
        ENDDO.
        APPEND ls_suppl_calc TO et_suppl_calc.
      ENDLOOP.
    ENDLOOP.
  ENDMETHOD.


  METHOD fill_suppliers_score.

    DATA:
      lv_leaving_flow  TYPE zde_value_type,
      lv_entering_flow TYPE zde_value_type.

    LOOP AT ct_suppliers_score ASSIGNING FIELD-SYMBOL(<entry>).
      CLEAR: lv_leaving_flow, lv_entering_flow.
      LOOP AT it_suppl_calc INTO DATA(ls_suppl_calc).
        IF ls_suppl_calc-main_supplier = <entry>-supplier.
          lv_leaving_flow = lv_leaving_flow + ls_suppl_calc-calculation.
        ELSEIF ls_suppl_calc-sub_supplier = <entry>-supplier.
          lv_entering_flow = lv_entering_flow + ls_suppl_calc-calculation.
        ENDIF.
      ENDLOOP.

      <entry>-evaluation = lv_leaving_flow - lv_entering_flow.

    ENDLOOP.

    SELECT SINGLE FROM @ct_suppliers_score AS a FIELDS MAX( a~evaluation ) AS max_
evaluation, MIN( a~evaluation ) AS min_evaluation
       INTO @DATA(ls_max_min).

    IF ls_max_min-min_evaluation < 0.
      LOOP AT ct_suppliers_score ASSIGNING <entry>.
        DATA(lv_delta) = CONV zde_value_type( ls_max_min-
max_evaluation - ls_max_min-min_evaluation ).
        <entry>-evaluation = ( CONV zde_value_type( <entry>-
evaluation + abs( ls_max_min-min_evaluation ) ) / lv_delta ) * 100.
      ENDLOOP.
    ENDIF.


  ENDMETHOD.


  METHOD fill_evaluation_values.

    DATA: lt_components TYPE abap_compdescr_tab,
          lv_value      TYPE zde_value_type,
          lt_values     TYPE zde_value_type_table.

    DATA:
      l_ref   TYPE REF TO cl_abap_tabledescr,
      l_dref  TYPE REF TO cl_abap_structdescr,
      text    TYPE string,
      lo_cast TYPE REF TO cx_sy_move_cast_error.

    FIELD-SYMBOLS:
      <value> TYPE zde_value_type,
      <type>  TYPE abap_bool.


    TRY.
        l_ref ?= cl_abap_typedescr=>describe_by_data( it_values_raw ).
        l_dref ?= l_ref->get_table_line_type( ).
        lt_components = l_dref->components.
      CATCH cx_sy_move_cast_error INTO lo_cast.
        text = lo_cast->get_text( ).
    ENDTRY.

    "-----------------------------------------------------------------------------
----

    TYPE-POOLS: abap.

    DATA: ls_components TYPE abap_compdescr.
    DATA: lo_strucdescr TYPE REF TO cl_abap_structdescr.
    DATA: lt_param_types_comp TYPE abap_compdescr_tab.

    IF is_env_mng_values_param_types IS NOT INITIAL.
      lo_strucdescr ?= cl_abap_typedescr=>describe_by_data( is_env_mng_values_para
m_types ).
      lt_param_types_comp = lo_strucdescr->components.
    ENDIF.

    "-----------------------------------------------------------------------------
----

    et_suppl_ev = it_values_raw.

    CALL METHOD me->calculate_column_nubmers
      EXPORTING
        it_any_table         = it_values_raw
      IMPORTING
        ev_number_of_columns = DATA(lv_number_of_columns).

    DO lv_number_of_columns - iv_key_fields_no TIMES.
      DATA(lv_comp_num) = sy-index.
      CLEAR lt_values[].

      TRY.
          ASSIGN COMPONENT lt_param_types_comp[ lv_comp_num ]-
name OF STRUCTURE is_env_mng_values_param_types TO <type>.
          DATA(lv_beneficial) = <type>.
        CATCH cx_sy_itab_line_not_found.
          lv_beneficial = abap_false.
      ENDTRY.


      LOOP AT it_values_raw ASSIGNING FIELD-SYMBOL(<wa_values_raw>).
        ASSIGN COMPONENT lt_components[ lv_comp_num + iv_key_fields_no ]-
```

```
name OF STRUCTURE <wa_values_raw> TO <value>.
        lv_value = <value>.
        APPEND lv_value TO lt_values.
      ENDLOOP.

      LOOP AT et_suppl_ev ASSIGNING <wa_values_raw>.
        ASSIGN COMPONENT lt_components[ lv_comp_num + iv_key_fields_no ]-
name OF STRUCTURE <wa_values_raw> TO <value>.
        get_value_score(
          EXPORTING
            iv_value          = <value>
            it_values         = lt_values
            iv_is_beneficial = lv_beneficial
          IMPORTING
            ev_score          = DATA(lv_score) ).

        <value> = lv_score.

      ENDLOOP.
    ENDDO.
  ENDMETHOD.


  METHOD get_weights_structure.

    IF iv_table_name_w IS NOT INITIAL.
      SELECT SINGLE *
        INTO es_weights
        FROM (iv_table_name_w)
          WHERE ekorg = ms_criteria_def-ekorg.

    ELSE.
      MESSAGE e013(zprc_sup_ev) INTO DATA(lv_message) WITH ms_input_data-
xhkrit ms_input_data-xhkrit.
      add_message_to_log( ).
      RAISE EXCEPTION TYPE zcx_msg EXPORTING ms_syst = sy.
    ENDIF.
  ENDMETHOD.


  METHOD create_certification_table.
    DATA :
      gw_line  TYPE REF TO data,
      gw_line1 TYPE REF TO data.
*
    FIELD-SYMBOLS:
      <gfs_line>      TYPE any,
      <gfs_dyn_table> TYPE STANDARD TABLE,
      <fs1>           TYPE any,
      <fs_other>      TYPE any,
      <table>         TYPE ANY TABLE.

    LOOP AT ct_bp_data_raw ASSIGNING FIELD-SYMBOL(<wa_bp_data_raw>).
      REPLACE ALL OCCURRENCES OF ` ` IN <wa_bp_data_raw>-ct_field WITH '_'.
    ENDLOOP.

    create_dynamic_table(
      EXPORTING
        iv_addtype = iv_addtype
      IMPORTING
```

```
        et_dynamic_table = gt_dyn_table
        et_certificates  = DATA(lt_certificates)
     ).

    ASSIGN gt_dyn_table->* TO <gfs_dyn_table>.
    ASSIGN et_certif_data TO <table>.

* Create dynamic work area for the dynamic table
    CREATE DATA gw_line LIKE LINE OF <gfs_dyn_table>.
    ASSIGN gw_line->* TO <gfs_line>.

    LOOP AT mt_suppl_for_eval INTO DATA(ls_suppl_for_eval).
      CHECK line_exists( ct_bp_data_raw[ vendor = ls_suppl_for_eval-lifnr ] ).
      ASSIGN COMPONENT 'SUPPLIER' OF STRUCTURE <gfs_line> TO <fs1>.
      <fs1> = ls_suppl_for_eval-lifnr.

      LOOP AT lt_certificates INTO DATA(ls_sertificate).
        "DATA(lv_comp_no) = sy-tabix + 1.
        IF line_exists( ct_bp_data_raw[ vendor = ls_suppl_for_eval-
lifnr ct_field = ls_sertificate ] ).
          ASSIGN COMPONENT ls_sertificate OF STRUCTURE <gfs_line> TO <fs_other>.
          <fs_other> = 1.
        ELSE.
          ASSIGN COMPONENT ls_sertificate OF STRUCTURE <gfs_line> TO <fs_other>.
          <fs_other> = 0.
        ENDIF.
      ENDLOOP.
      APPEND <gfs_line> TO <gfs_dyn_table>.
      CLEAR: <gfs_line>.
    ENDLOOP.

    et_certif_data = <gfs_dyn_table>.

  ENDMETHOD.


  METHOD create_dynamic_table.
    DATA:
      gw_dyn_fcat TYPE lvc_s_fcat,
      gt_dyn_fcat TYPE lvc_t_fcat.

    DATA : gv_pos TYPE i.
    DATA : fname TYPE string.

* Declaring the first column - SUPPLIER
    gv_pos = gv_pos + 1.

    gw_dyn_fcat-fieldname = iv_key_field_name.               " Field Name
    "gw_dyn_fcat-outputlen = 10.                       " Output Length
    gw_dyn_fcat-domname = 'zde_value_type'.
    gw_dyn_fcat-tabname   = 'LT_CERTIFICATES'.        " Local Table Name
    gw_dyn_fcat-coltext   = 'SUPPLIER'.               " Header text for the column
    gw_dyn_fcat-col_pos   = gv_pos.                   " Column position
    gw_dyn_fcat-
key = 'X'.                            " Key attribute is set for the field vend.
    APPEND gw_dyn_fcat TO gt_dyn_fcat.
    CLEAR gw_dyn_fcat.

    SELECT ct_field
      FROM tp22
      INTO TABLE @DATA(lt_certificates)
      WHERE addtype = @iv_addtype
        AND data_type = @iv_data_type.

    LOOP AT lt_certificates ASSIGNING FIELD-SYMBOL(<wa_sertificate>).
      "INTO DATA(ls_sertificate).
      DATA(lv_sert_name) = CONV string( <wa_sertificate>-ct_field ).
      REPLACE ALL OCCURRENCES OF ` ` IN <wa_sertificate>-ct_field WITH '_'.
      REPLACE ALL OCCURRENCES OF `/` IN <wa_sertificate>-ct_field WITH '_'.
      REPLACE ALL OCCURRENCES OF `\` IN <wa_sertificate>-ct_field WITH '_'.
      gv_pos = gv_pos + 1.
      gw_dyn_fcat-fieldname = <wa_sertificate>-ct_field.
      gw_dyn_fcat-tabname   = 'LT_CERTIFICATES'.
      gw_dyn_fcat-coltext   = <wa_sertificate>-ct_field.
      "gw_dyn_fcat-outputlen = 13.
      gw_dyn_fcat-datatype  = iv_fields_datatype.
      gw_dyn_fcat-intlen = iv_fields_intlen.
      gw_dyn_fcat-decimals = iv_fields_decimals.
      "gw_dyn_fcat-dd_roll  = 'zde_value_type'.
      "gw_dyn_fcat-domname = 'zde_value_type'.
      gw_dyn_fcat-rollname   = 'zde_value_type'.
      gw_dyn_fcat-col_pos   = gv_pos.
      APPEND gw_dyn_fcat TO gt_dyn_fcat.
      CLEAR gw_dyn_fcat.
    ENDLOOP.

    DATA :
      "gt_dyn_table TYPE REF TO data,
      gt_dyn_table TYPE REF TO data,
      gw_line      TYPE REF TO data,
      gw_line1     TYPE REF TO data.

    FIELD-SYMBOLS:
      <gfs_line>      TYPE any,
      <gfs_line1>     TYPE any,
      <gfs_dyn_table> TYPE STANDARD TABLE,
      <fs1>           TYPE any,
      <fs_other>      TYPE any,
      <table>         TYPE ANY TABLE.

    CALL METHOD cl_alv_table_create=>create_dynamic_table
      EXPORTING
        i_style_table             = 'X'
        it_fieldcatalog           = gt_dyn_fcat
      IMPORTING
        ep_table                  = gt_dyn_table
      EXCEPTIONS
        generate_subpool_dir_full = 1
        OTHERS                    = 2.

    ASSIGN gt_dyn_table->* TO <gfs_dyn_table>.
    et_dynamic_table = gt_dyn_table.
    et_certificates = lt_certificates.
  ENDMETHOD.


  METHOD get_current_currency.
```

```
  SELECT SINGLE *
    INTO @DATA(ls_t024e)
    FROM t024e
      WHERE ekorg EQ @ms_input_data-xekorg.

  IF sy-subrc EQ 0 AND ls_t024e-bukrs NE space.
    SELECT SINGLE *
      INTO @DATA(ls_t001)
      FROM t001
        WHERE bukrs EQ @ls_t024e-bukrs.

    IF sy-subrc EQ 0.
      sy-waers = ls_t001-waers.
    ENDIF.

  ELSE.
    SELECT SINGLE *
      INTO @DATA(ls_t000)
      FROM t000
        WHERE mandt EQ @sy-mandt.

    IF sy-subrc EQ 0.
      sy-waers = ls_t000-mwaer.
    ENDIF.
  ENDIF.
ENDMETHOD.
```