

Research Article

A Multicriteria Approach to Support Task Allocation in Projects of Distributed Software Development

Marum Simão Filho  ^{1,2}, **Plácido R. Pinheiro**  ¹, **Adriano B. Albuquerque**  ¹,
Régis P. S. Simão  ², **Raimundo S. N. Azevedo**  ², and **Luciano C. Nunes**  ^{1,2}

¹*Graduate Program in Applied Informatics, University of Fortaleza, Brazil*

²*Information Systems, 7 de Setembro University Center, Brazil*

Correspondence should be addressed to Marum Simão Filho; marumsimao@gmail.com

Received 29 November 2018; Accepted 7 March 2019; Published 1 April 2019

Guest Editor: Marin Lujak

Copyright © 2019 Marum Simão Filho et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Organizations are increasingly investing in Distributed Software Development (DSD) over the years. A typical decision-making problem in the distributed scenario consists of deciding which team should be allocated each task. That decision takes into account a relative degree of subjectivity. That setting is suitable for applying Verbal Decision Analysis (VDA). This paper introduces an approach to support the allocation of tasks to distributed units in DSD projects, structured on the hybridisation of methods of Verbal Decision Analysis for classification and rank ordering applied to influencing factors and executing units. Firstly, a review of the literature was conducted aiming to identify the approaches to support the allocation of tasks in DSD contexts. Then, an approach was developed by applying VDA-based methods for classification and ordering. Bibliographic research and the application of surveys with professionals allowed identifying and characterising the main elements that influence task assignment in DSD projects. Afterwards, experiences were carried out in five real-world companies. In the end, the proposed approach has been submitted to the evaluation by the professionals of the participating companies and by some project management experts. The proposed approach comprises a workflow containing responsible actors and descriptions of the activities. Automated tools are also employed in automating the implementation of the approach. After applying the approach in five companies, task assignment recommendations are presented in groups for each company, according to the task type, i.e., requirements, architecture, coding, and testing, ranging from the most to the least preferable office. Results of the experiences and evaluations held during this work present evidence that the proposed approach is flexible, adaptable, and easy to understand and to use. Moreover, it helps to reduce decision subjectivity and to think of new aspects, supporting the task allocation process in DSD.

1. Introduction

Software development can be structured in several activities, such as requirements, design, architecture, and programming. Companies that have a mature and organized development process can direct activities to be developed in different locations, taking advantage of the skills of local teams.

A typical decision-making problem in the distributed scenario consists of deciding which team should be allocated each task. The specific literature mentions some methodologies and models that address the decision problem on the division of tasks in distributed projects using diverse strategies. Some approaches consist of simple tools whereas

others consist of complex systems. Some solutions define simple procedures while others comprise whole processes. The granularity of the object to be allocated also varied widely, from single tasks to full stages (sets of tasks).

However, few models use multicriteria approaches and, when they do, they usually focus on quantitative or numerical aspects, often requiring a heavy mathematical framework. Furthermore, only a small set of criteria was usually used to decide on the allocation of tasks, especially the job costs. Team skills and experience, and cultural issues, among other relevant factors, have been often neglected [1]. In practice, what happens is that, most times, the professional responsible for the project takes into account only their experience and

some knowledge about the project and the teams to decide how the tasks will be distributed, that is, in a subjective and restricted way.

Situations like these bring some degree of subjectivity to the decision-making procedure. This scenario is suitable for Verbal Decision Analysis, which is a methodology that employs qualitative analysis to solve multicriteria problems [2]; that is, the methods that integrate the VDA framework address the subjectivity of the criteria. In VDA methods most decision-making problems can be represented in a qualitative way. In addition, the methods use the verbal representation of the problems to carry out the decision process [3].

Considering the given scenario, in this paper we introduce DiVA, an approach to support the assignment of tasks to distributed teams Distributed Software Development Projects, structured on the hybridisation of methods of Verbal Decision Analysis for classification and rank ordering, applied to influencing factors and executing units. As an executing unit, we understand individuals, teams, departments, offices, or any other entity that develops the work itself.

The remainder of the work is organized as described next. In Section 2, some theoretical reference necessary to support the understanding of the research is provided. Section 3 presents the main works found in the literature related to the subject of this paper. Section 4 introduces and describes the proposed task allocation approach to DSD projects. In Section 5, we report the experiences of using the proposed approach performed in real companies, as well as providing the results, evaluations and limitations of the work. Section 6 contains the conclusions and some suggestions for further works.

2. Theoretical Reference

2.1. Task Allocation in Distributed Software Development. Organizing tasks is one of the main assignments of a project manager. He or she is responsible for determining which tasks are essential to the completion of each step of the process and for allocating those tasks to the team members of different backgrounds and experience. Workload, expertise, maturity, and profile of each team member are criteria commonly taken into account in the task allocation process.

In this research, we adopt the PMBOK (Body of Knowledge in Project Management [4]) definition of “task”, which is a responsibility that can be allocated within the project management plan in such a way that the designated resource is subject to the obligation to perform the task requirements. PMBOK also defines task assignment in software development as an activity that settles the way in which the tasks will be carried out and how and which professionals will be assigned the tasks, considering the imposed restrictions [4]. These tasks set out what should be done to achieve not only the project goals but also the organization’s objectives [5].

When we identify tasks in distributed projects, we usually think of the constraints, whether technical, social, or professional, related to assigning tasks to distributed groups. In other words, the task allocation planning should consider people’s characteristics (such as personal skills, for example) as well as the characteristics of relationships

among distributed teams (such as cultural differences and communication issues). In addition, the characteristics of the project or the product to be developed, such as complexity, dependencies, and abilities necessary for its development, cannot be ignored. [1].

The task allocation is a crucial step for project management, mainly if it is a distributed project. The allocation of tasks to distributed teams should simultaneously match the right task with the right team in the correct order while defining the way to carry out the tasks. Moreover, assuming that every project deals with limited resources, such as people and time, to accurately plan, project managers must take into account (a) the attributes of the teams, (b) the features of the products that will be build, (c) the properties of the tasks and the relationships between them, and, finally, (d) the organization’s objectives [5]. Thus, it is critical to map the main aspects that influence the assignment of tasks in distributed projects. We will call such aspects “influencing factors”, and we will deal with them later.

2.2. Verbal Decision Analysis. Decision-making is an everyday activity in people’s and companies’ lives. To make an assertive decision, we evaluate a problem by judging a set of characteristics or attributes; i.e., our analysis addresses many factors, which we call criteria. Decisions can have a considerable impact on the business context, as choosing the wrong alternative can lead to loss of time, money, and resources, causing detrimental effects for the company. Such management decisions usually take into account several factors. For these cases, it is advisable to adopt methodologies to support the decision-making process.

The decision-making process involving the analysis of alternatives from various points of view is called multicriteria decision analysis. There are many methodologies that support multicriteria decision analysis [6]. Such methodologies help to increase the confidence of the decision-maker as they promote the generation of knowledge about the decision-making context [7].

Given this perspective, Verbal Decision Analysis (VDA) is an approach that employs qualitative analysis for solving multicriteria problems [2]. In VDA, the problems assume a verbal representation. The VDA methods are applied for classifying and rank ordering the alternatives. Some examples of VDA’s classification methods are ORCLASS [3], SAC [8], DIFCLASS [9], and CYCLE [9]. Examples of VDA’s ordering methods are PACOM [3], ARACE [10], STEPZAPROS, ZAPROS-LM, ZAPROS III, and ZAPROS III-i [11].

There are many studies describing the application of VDA methods in real-world problems. Some of them are provided next. Machado et al. [12] used VDA to help the selection of specific CMMI practices. Mendes et al. [13] and Tamanini et al. [14] applied VDA in digital TV scenario. Machado [15] developed a hybrid model of VDA for selecting project management approaches and, in [16], Machado et al. employed VDA for helping to choose the best SCRUM practices. Tamanini et al. [17] created a VDA-based model to cashew chestnut industrialization process. Tamanini et al. ([18, 19]) and Castro et al. ([20–22]) carried out studies using VDA to support the diagnosis of the Alzheimer’s

disease. Gomes et al. [23] applied VDA in the marketing business. Barbosa et al. [24] described the use of VDA to prioritize software requirements. Finally, Simão Filho et al. [25] employed VDA in project portfolio ranking.

ORCLASS (ORDinal CLASSification) is a method of the VDA framework that aims to classify the alternatives in a set. For this, it is necessary that these alternatives be classified by the decision-maker in a small number of decision groups or classes, often two. The first group will contain preferable alternatives, and the second group will include nonpreferable alternatives [3]. In addition, a set of criteria and their respective values should be established to allow the evaluation of alternatives.

Some advantages are associated with the ORCLASS method, which we highlight below [3]. First, the comparison of the values is performed verbally since the preferences elicitation is performed in the native language of the decision-maker. Second, the classification rule is organized according to the presentation of the most significant combinations of criteria for the decision-maker. Such strategy decreases the number of questions that the decision-maker would be required to answer during the preferences elicitation. Third, it is possible to check the preferences consistency of the decision-maker. Finally, it is possible to formulate a verbal explanation for the decisions from the classification rules obtained. Such advantages motivated its choice for this study.

In order to make the application of the ORCLASS method easy, the tool ORCLASSWEB was developed and then described by Machado et al. [26]. It can be accessed through the link <http://www2.unifor.br/OrclassWeb>. ORCLASSWEB aims to make the process of alternatives comparison automatic, providing a result for the problem to the decision-maker, following the ORCLASS method specification. Another factor that favoured the choice of ORCLASS for this study was the existence of an automated tool to run the model.

The ZAPROS III-i method also belongs to the VDA framework. It integrates the ZAPROS family, and its function is to rank alternatives to problems with a small set of criteria and criteria values and a high number of alternatives. The ZAPROS III-i method adopts the same idea of criteria and criteria values employed by the ORCLASS method to assess the alternatives. The method uses values representing the distances between judgments on the scale of the same criteria or between two criteria to elicit the preferences of the decision-maker. A scale of preferences is then built in such a way that allows the alternatives comparison.

The main advantages of the ZAPROS method are emphasised next [27]. First, during the preferences elicitation stage, the model exposes issues that are apprehensible to the decision-maker, according to the criteria values. This process has psychological validity since it accepts the limitations of the human information processing system. Such property consists of the most significant benefit of the method. Besides, the method can detect contradictory inputs from the decision-maker and request a solution for them. Finally, the ZAPROS provides all qualitative comparison information in a language that is natural to the decision-maker. Such advantages influenced its choice for this study.

To facilitate the application of the ZAPROS III-i method and to allow it to be executed consistently, the ARANAÚ tool was implemented ([28–30]). Although initially ARANAÚ has been built for the ZAPROS III method, an updated version for the ZAPROS III-i method was used in this work. As in the previous method, the existence of an automated tool to run the model contributed to its choice.

3. Related Work

Today, more and more companies are conducting projects in a distributed way. Assigning tasks is a crucial activity for projects in general, especially in a distributed environment. Much research on the distribution of tasks in DSD has been conducted recently in order to clarify this issue, its difficulties, and its challenges. In this perspective, some approaches, methodologies, and models for distributing tasks considering remote teams were created.

Aiming to identify the related works to the present research, we have developed a systematic review of the literature, which is described in detail in [31]. Our main objectives in conducting the systematic review were (i) to discover which approaches were used to allocate tasks in projects of DSD; (ii) to know whether the approach was based on qualitative multicriteria decision-making methods; (iii) to find out if the approach was based on VDA methods; and (iv) to identify if the study mentioned some factor or criterion that influenced the assignment of tasks in DSD scenarios.

To guide the literature review, based on the objectives cited above, we formulated a series of questions, for which we seek answers in the studies researched. Table 1 shows the research questions. “PQ” identifies primary questions, and “SQ” identify secondary questions.

As a result of the literature review, some relevant papers were selected, which are briefly described next. Some of the most relevant works are those of Lamersdorf and his colleagues. In [1], Lamersdorf et al. analysed different approaches to the distribution of responsibilities. The analysis included various approaches in the areas of distributed development, distributed generation, and distributed systems. As a conclusion, they pointed to the Bokhari algorithm as a high applicability model in DSD. Lamersdorf and Münch [32] presented TAMRI (Task Allocation based on Multiple cRIteria), an approach structured on the Bokhari algorithm and Bayesian networks that adopts various criteria and influences factors to assist the systematic assignment of tasks in DSD projects.

Ruano-Mayoral et al. [33], in turn, proposed a methodological framework to distribute work packages to professionals in global software development (GSD) projects. The proposed approach was composed of two stages: definition and validation. The paper also provided the results of the application of the proposed approach. In [34], Almeida et al. introduced McDSDS (Multicriteria Model for Planning Distributed Software Development Projects with Scrum). The proposed model involves a multicriteria decision applied to planning and fine-tuning project plans. It was based on cognitive mapping and MACBETH (Measuring Attractiveness by a Categorical Based Evaluation Technique) [6]. In [35],

TABLE 1: Codes used in the data consolidation table.

Question Id.	Description
PQ1	What approach was used or suggested for task assignment in DSD projects?
PQ2	Is the approach based on qualitative multi-criteria methods of decision-making?
PQ3	Is the approach based on any method of Verbal Decision Analysis framework?
PQ4	Does the study mention any factor or criterion that influences the assignment of tasks in distributed software environments?
SQ1	Does the approach describe a process?
SQ2	Does the approach have any automated support tool?
SQ3	Does the approach apply influencing factors extracted from the specific literature?
SQ4	Can the approach be instantiated from an organisation standard?
SQ5	Does the approach reuse knowledge?
SQ6	Does the approach presuppose consensus meeting?
SQ7	Does the approach presuppose feedback of results?
SQ8	Does the approach allow using a subset of the whole process?
SQ9	Does the approach allow modifying the main parameters (allocation object and executing unit)?
SQ10	Has the approach been applied in real-world situations?
SQ11	Has the approach been evaluated by users or professionals?
SQ12	Has the approach been evaluated by knowledge area experts?

Almeida et al. reported the application of the McDSDS in the selection of DSD Scrum-based project plans, whose chances of success are higher.

Pedras et al. ([36, 37]) introduced DIMANAGER, a tool integrated to DiSEN environment (Distributed Software Engineering Environment) ([38, 39]) whose goal is to assist the choice of team members according to their skills, knowledge, and availability. DIMANAGER used Fuzzy Logic to quantify the identified criteria.

Another model was developed by Setamanit et al. ([40–42]), who described an approach that identifies dynamic and discrete factors of a DSD domain and produces data regarding the dedicated resources, productivity, coordination, and communication throughout the project. By using the model, it is possible to configure several remote sites and their procedures involving several steps and assignment policies, enabling make the best choice to the assignment systematics.

Prikladnicki et al., in their turn, introduced MuNDDoS ([43, 44]) a Reference Framework for DSD. The authors created a project allotment flow that selects projects to be conducted in each remote office, taking into account the allocation strategy established by the company.

In [45], Jalote and Jain described a 24-hour Development Method that adopts a Directed Acyclic Graph (DAG). In such a DAG, the vertices symbolise the tasks, and the edges denote the current operational dependencies. Besides, the method involves three sets of resources, denoting professionals from each remote office related to each fraction of 8 hours of work. By using the critical path model, the method performs the attribution aiming to shorten the duration of the project.

Mullick et al. [46] introduced the Global Studio Project, a distribution model designed to optimise the work allocation.

In such a model, groups of professionals from several universities worldwide execute distributed development methods and techniques. All development work is divided into related work packages along with the background needed to perform them and their temporal dependencies. From this data and the professionals' knowledge, the packs are allocated manually.

In [47], Mak and Kruchten developed NextMove, an approach structured on object-oriented process modelling and project management practices to support the coordination and task assignment for remote professionals in an agile development scenario. NextMove applies the AHP (Analytic Hierarchy Process) approach to indicate the most suitable professional to receive a task according to their abilities.

Barcus and Montibeller [48] presented a real case study wherewith they constructed a multicriteria approach to assist the distributed tasks assignment decision for a global software company. The approach was formulated with the support of several software development project managers, combining decision conferencing and value assessment of several factors. The approach encompasses software engineering factors as well as strategic and non-strategic issues such as professionals' motivation and possibilities for education.

Finally, dos Santos et al. [49] proposed a recommendation framework for assigning development teams in distributed projects of software product lines. The framework consists of an activity flow that, in the end, indicates the best recommendation for the allocation. They reported the application of the framework in real situations.

After analysing the works in detail, we elaborated Table 2, which provides an overview of the papers as well as the answers to the research questions. Cells marked with “√”

TABLE 2: Mapping of approaches and research questions.

Approach	PQ1	PQ2	PQ3	PQ4	SQ1	SQ2	SQ3	SQ4	SQ5	SQ6	SQ7	SQ8	SQ9	SQ10	SQ11	SQ12
TAMRI [32, 58]	An approach structured on the Bolckhari algorithm and Bayesian networks and adopts various criteria and influence factors to assist the systematic assignment of tasks in DSD projects.	-	-	✓	✓	✓	✓	✓	-	-	-	✓	✓	✓	✓	✓
Ruano-Mayoral's Methodological Framework [33, 50]	A methodological framework to distribute work packages to professionals in GSD projects.	-	-	✓	✓	✓	✓	✓	-	-	-	✓	✓	✓	✓	✓
McDSDS [34, 35, 71]	A multi-criteria decision approach for planning DSD projects with Scrum.	✓	-	✓	-	✓	✓	✓	-	-	-	✓	-	-	-	-
DIMANAGER [36-39]	A tool integrated to DISEN environment to assist the choice of team members. It uses Fuzzy Logic as its basis.	-	-	✓	-	✓	✓	✓	-	-	-	✓	✓	✓	✓	✓
GSD Simulation Model [40-42]	An approach that identifies dynamic and discrete factors of a DSD domain and produces data regarding the involved resources, productivity, coordination, and communication throughout the project.	-	-	✓	-	✓	✓	✓	-	✓	✓	✓	✓	-	-	-
24 hours Development Model [45]	A method that adopts a DAG Directed Acyclic Graph) and uses the critical path model. It performs the attributions aiming to shorten the duration of the project.	-	-	✓	-	-	✓	✓	-	-	-	✓	-	-	-	-
MuNDDoS [43, 44]	A reference model for DSD, which involves a project allotment flow that selects projects to be conducted in each remote office, taking into account the allocation strategy established by the company.	-	-	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	-	-	-

TABLE 2: Continued.

Approach	PQ1	PQ2	PQ3	PQ4	SQ1	SQ2	SQ3	SQ4	SQ5	SQ6	SQ7	SQ8	SQ9	SQ10	SQ11	SQ12
Global Studio Project [46]	A distribution model designed to optimize the work allocation. All development work is divided into related work packages along with the background needed to perform them and their temporal dependencies.	-	-	✓	-	✓	✓	-	-	-	-	✓	✓	-	-	-
NextMove [47]	An approach structured on object-oriented process modelling and project management practices to support the coordination and task assignment for remote professionals in an agile development scenario. It uses AHP model as its basis.	-	-	✓	-	✓	✓	✓	✓	-	✓	-	-	-	-	-
Barcus and Montibeller [48]	A multi-criteria approach to assist the distributed tasks assignment decision, which combines decision conferencing and value assessment of several factors.	✓	-	✓	-	✓	✓	✓	✓	-	✓	-	✓	-	-	✓
Recommendation Framework for allocation of dev teams in dist. proj. SPL [49]	A framework that gives suggestions for team allocation in distributed Software Product Lines projects.	-	-	✓	-	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-

indicate that a positive answer to the corresponding question was found in the investigated works. Cells filled with “-” denote that we could not conclude anything about the research question in the works consulted. It is worth emphasising that our evaluation considered the information that was published and accessible throughout this research. Also, the author made his interpretations when he did not find explicit information in the texts read.

From these studies, we have noted some variation in the granularity of the task to be allocated. Some approaches consider individual tasks, such as [38, 45, 47, 48], while others focus on groups or packages of tasks ([33, 48]), or even entire projects, such as [43].

We also detected variations about to the target of the assignment. Some authors have explored the allocation of tasks to individuals [38, 45, 47], others consider teams, offices, or branches ([1, 34, 40, 43, 48]).

Some works have elaborated abstract models, such as reference models and frameworks ([43, 50]), while others implemented software tools from the proposed models ([36]).

It is worth noting that considering the works analysed, only three presented approaches based on multicriteria decision-making methods ([34, 47, 48]).

Nevertheless, only two studies used qualitative analysis as a basis, ([34, 48]), and none considered Verbal Decision Analysis methods. Besides the above-listed studies, other works did not present approaches but referred to some factors or criteria that influence or affect the allocation of tasks in DSD. Among them, we highlight [51–57].

Considering the fifteen objective questions assessed in this study (three primary questions, PQ2 to PQ4, and twelve secondary questions), PQ4 and SP3 questions, which concerned the influencing factors for task assignment in DSD projects, were successful for all the investigated approaches. Meanwhile, only one approach addressed the question SQ6 (consensus meeting), and no approach satisfied the question P3 (structured on some VDA method). Table 3 details the performance of the issues investigated.

From this analysis, we have noted that some works researched have developed multicriteria models, but generally considering the quantitative or mathematical approach. Moreover, in most cases, a small number of elements were applied to guide the attribution of tasks such as labour expenses, mainly. Thus, we could identify gaps in the set of existing approaches for task allocation in DSD that allowed us to drive some research to use multicriteria methods based on Verbal Decision Analysis. Moreover, many influencing factors were identified in the DSD task allocation decision process, which may be useful for future works. Besides, several aspects have been analysed in the existing approaches that have enabled us to think of some desirable features for a new approach in this field, such as the issues addressed in the secondary questions.

3.1. Influencing Factors for Task Allocation in DSD. After the analysis described in the previous subsection, it was possible to establish the main factors that impact or influence the allocation of tasks in DSD projects. This work was mainly

based on the studies of Lamersdorf et al., such as [1, 58, 59]. They carried out literature studies regarding the distributed scenario and explored the criteria or factors that affect the activity of task allocation in GSD. Other works were also used as a basis for composing the relation of influencing factors in the allocation of tasks used in this approach, such as [5, 51–57]. In the end, we have selected the fifteen most cited factors in these works.

Table 4 lists the fifteen factors selected. These factors will be an important part of the approach presented in this paper.

4. DiVA—An Approach to Support the Allocation of Software Development Tasks to Distributed Teams Based on Verbal Decision Analysis

In this section, we introduce DiVA, an approach to support the allocation of software development tasks to distributed teams based on Verbal Decision Analysis. The approach is based on Verbal Decision Analysis methods for classification and ordering. For the reasons discussed previously, the methods adopted are ORCLASS for classification and ZAPROS III-i for ordination. These methods work with some common concepts and have already proven successful in their hybridisation, as shown in previous works such as [16, 60]. Preliminary works on VDA application to the problem of task allocation in DSD context can be found in [61–67].

The approach is divided into four main stages, namely, context characterisation, ranking of the influencing factors, ranking of the executing units, and evaluation and decision. Context characterisation stage is responsible for the first definitions of the basic concepts, such as tasks, criteria, criteria values, and preference groups. In the second stage, ranking of the influencing factors, the VDA methods are employed to classify and order the influencing factors considering the selected criteria and criteria values, taking into account the types of tasks chosen. In the third stage, ranking of the executing units, the preferable influencing factors identified in the previous stage are now converted into criteria for evaluating the possible destination of the tasks, which may be offices, departments, or teams. Finally, in the last stage, evaluation and decision, restrictions, conditions, and limiting factors are identified, and the results are evaluated taking these issues into account. After the evaluation, the decision is made according to the process of each organization. Figure 1 shows a procedural view of DiVA with actors responsible for each activity. The process shows the activities grouped in their respective stages. Note that depending on the type of data the company already has, it may choose to start the process from stage 2 or stage 3, not necessarily from stage 1. Likewise, depending on the outcome of the evaluation of the data and the decisions made, it is possible to feedback the approach, returning to stage 1, 2, or 3. The company can also choose to instantiate the flow once each cycle (semiannual, annual), and then the company can only calibrate the model from the data feedback. A more detailed description of DiVA stages and activities is provided next.

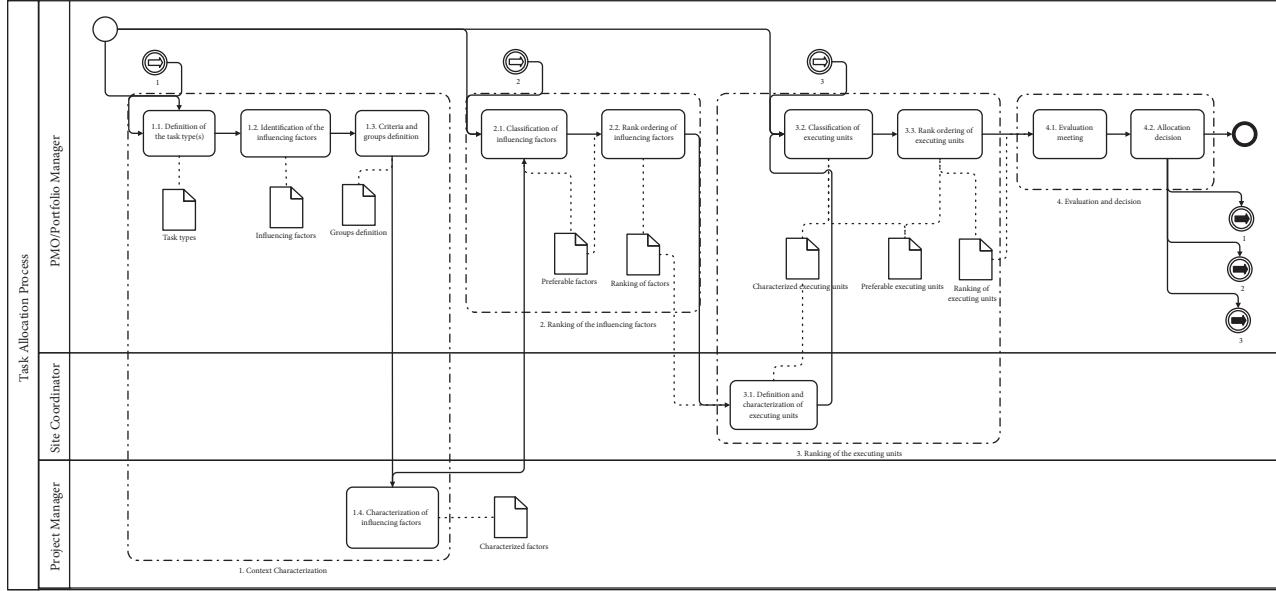


FIGURE 1: The process of the DiVA approach.

TABLE 3: Total approaches that scored for each question.

Q. Id.	Description	Count
PQ4	Does the study mention any factor or criterion that influences the assignment of tasks in distributed software environments?	11
SQ3	Does the approach apply influencing factors extracted from the specific literature?	11
SQ4	Can the approach be instantiated from an organisation standard?	8
SQ9	Does the approach allow modifying the main parameters (allocation object and executing unit)?	8
SQ2	Does the approach have any automated support tool?	6
SQ5	Does the approach reuse knowledge?	6
SQ10	Has the approach been applied in real-world situations?	6
SQ1	Does the approach describe a process?	5
SQ7	Does the approach presuppose feedback of results?	5
SQ8	Does the approach allow using a subset of the whole process?	5
SQ12	Has the approach been evaluated by knowledge area experts?	4
SQ11	Has the approach been evaluated by users or professionals?	3
PQ2	Is the approach based on qualitative multi-criteria methods of decision-making?	2
SQ6	Does the approach presuppose consensus meeting?	1
PQ3	Is the approach based on any method of Verbal Decision Analysis framework?	0

(1) Context Characterisation: This stage comprises four activities, which are detailed below:

(1.1) Definition of the Task Type(s): Define the work object of the approach, i.e., which tasks, type of

tasks, or groups of tasks that will be analysed for allocation.

(1.2) Identification of the Influencing Factors: Identify which factors influence the distribution of tasks in the context being analysed.

TABLE 4: Influencing factors on task allocation in DSD projects.

ID	Factors	Description
Factor1	Technical skills	Knowledge and ability on techniques, programming languages, frameworks, tools, APIs, necessary for the professionals to carry out the task.
Factor2	Knowledge in business	Knowledge of the teams about the area, domain or business of the clients.
Factor3	Project manager maturity	Experience, background, and maturity of project managers in their profession
Factor4	Proximity to the customer	The office whose team will perform the task is positioned geographically close to the client.
Factor5	Low turnover rate	The turnover rate of office employees is low, that is, typically in the branch, there are few changes in teams.
Factor6	Team availability	Teams seek to be free, available to perform tasks.
Factor7	Team maturity	Teams are mature / experienced concerning the task being performed.
Factor8	Team personal trust	Managers and colleagues believe and trust in themselves and each other.
Factor9	Same time zone	The various teams associated with the task execution work in the same time zone.
Factor10	Cultural similarities	The various teams associated with the task execution share the same cultural aspects.
Factor11	Team willingness	Teams are motivated, excited and interested in the work.
Factor12	Low labour cost	The cost of professionals in the office is low/attractive.
Factor13	Maturity in the process	The teams are mature and experienced concerning the software development process adopted.
Factor14	Language fluency	The teams have fluency in the foreign languages commonly used in the office.
Factor15	Good communication infrastructure	The office has a good communication infrastructure (speed, availability, redundancy, among others).

- (1.3) Definition of Criteria and Preference Groups: Determine the criteria and their respective values to be used to assess the influencing factors in task assignment. Moreover, define the preference groups to categorize the influencing factors. The groups usually defined are (i) the preferable and (ii) the nonpreferable factors.
- (1.4) Characterisation of Influencing Factors: Characterise the influencing factors according to the criteria and criteria values, by task group or task. This means that the company must evaluate the factors against the criteria chosen.
- (2) Ranking of the Influencing Factors: This stage consists of two activities, which are described next:
- (2.1) Classification of Influencing Factors: Use the ORCLASS method to classify the influencing factors identified in Activity (1.2) into the preference groups defined in Activity (1.3). The preferable influencing factors, properly characterised (as a result of Activity (1.4)), will be

- processed by the ORCLASS method in this step. The influence factors classified as nonpreferable should be discarded.
- (2.2) Rank Ordering of Influencing Factors: Use the ZAPROS III-i method to rank order the preferable influencing factors (those from the group (i) resulting from the previous Activity (2.1)), using the same factors characterisation generated in the Activity (1.4).
- (3) Ranking of the Executing Units (Individuals/Offices/Teams): This stage is composed of three activities, which are detailed next:
- (3.1) Definition and characterisation of the executing units: After identifying the main influencing factors in task assignment in DSD projects, the approach now considers the influencing factors as criteria and the executing units (company offices or teams) as the alternatives for the VDA methods. In this activity, relations must be

- established between the influencing factors and the executing units; that is, the company's professionals should evaluate the offices, teams, or other executing units regarding the influencing factors.
- (3.2) Classification of the Executing Units: Use the ORCLASS method to classify the executing units (offices/teams) into preference groups (defined in the Activity (3.1)), usually two: (i) the preferable and (ii) the nonpreferable ones. As in the previous stage, nonpreferable executing units should be discarded.
 - (3.3) Rank Ordering of the Executing Units: Use the ZAPROS III-i method to rank order the preferable executing units (those from the group (i) resulting from the previous Activity (3.2)), using the same units characterisation generated in the Activity (3.1).
 - (4) Evaluation and Decision: This stage consists of two activities, which are described below:

- (4.1) Evaluation Meeting: The portfolio management team or equivalent meets to discuss the results of the approach, assessing any restrictions that may exist. Results may change as a consequence of this activity.
- (4.2) Allocation Decision: Finally, decisions about task allocation are made, and tasks are assigned to the responsible teams according to the organization's process or procedure.

The next section reports on the experiences of use carried out with the objective of evaluating the proposed approach.

5. Some Experiences with the Proposed Approach

5.1. Preparation. To evaluate DiVA and obtain indications about how it supports organizations in the allocation of tasks in DSD, we conducted some experiences. The application of DiVA was carried out in five real-world companies. After conducting the experiences of use, we chose to evaluate the application of the approach qualitatively, according to the stakeholders' perception, collected through questionnaires. Note that the study scenarios can be classified as *in vivo* [68], that is, a validation carried out in real development environments with the supervision or participation of researchers and professionals.

The purpose of the experiences of use was, for each company, to establish rankings of the most recommended executing units to assign a particular type of software development task. For this scenario, some assumptions have been made:

- (i) We considered the companies' offices as the executing units, that is, the entities that would perform the tasks themselves;

- (ii) Four different types of tasks were taken into account, namely: requirements-related tasks, architecture/design-related tasks, implementation-related tasks, and testing-related tasks.

Due to time restrictions of the companies' professionals and to generate a knowledge base to be used in future instantiations of the approach, we chose to structure the application of the approach considering two phases. The first phase, which covered the stages "Characterisation of the Context" and "Ranking of the Influencing Factors", was common to all participating companies. At the end of the first phase, the approach generated four ordered lists of influencing factors, i.e., one list for each task type (requirements, architecture/design, and implementation/coding and testing).

The second phase involved the stage of "Ranking of the executing units" for the participating companies. In this case, the application was individualized per company. The stage "Evolution and Decision" was not carried out as part of these experiences of use due to the obvious difficulties of putting into practice within the companies. Table 5 shows the details of the strategy for the application of the approach.

We sought to select companies with different profiles, with subsidiaries in various cities, and even in several countries. The size of the companies ranged from medium to large. Figure 2 shows the geographic distribution of the companies' offices involved in the research.

5.2. Execution

5.2.1. Context Characterisation. In order to perform Stages 1 and 2, the influencing factors previously identified and exposed in Table 4 served as alternatives for the classification and ordering VDA methods (ORCLASS and ZAPROS III-i).

For this activity, we followed the PMBOK's concepts [4] to help us choose the criteria because it has great acceptance in the software industry worldwide. Therefore, for our research, we decided to adopt the iron triangle, which is composed of time, cost, and quality, as the criteria for measuring success in project management. The factors were then characterised according to following criteria: time, cost, and quality of the task. These criteria have three possible values (high gain, some gain, and no gain) with the following interpretations (the adopted methods do not accept negative values):

- (i) High gain: The existence or presence of the factor causes a very positive influence on the time/cost/quality of the task to be performed.
- (ii) Some gain: The existence or presence of the factor provokes some positive influence on the time/cost/quality of the task to be performed.
- (iii) No gain: The existence or presence of the factor produces no influence on the time/cost/quality of the task to be performed.

To build the characterisation vectors for the alternatives of our problem, we needed to find out the impact of influencing factors regarding time, cost, and quality in tasks in DSD. For



FIGURE 2: Geographic distribution of the companies' offices.

TABLE 5: Strategy for application of the approach.

Stages	Activities	Strategy
(1) Context Characterisation	(1.1) Definition of tasks types (1.2) Identification of influencing factors (1.3) Criteria and groups definition (1.4) Characterisation of influencing factors	Performed once for all participating companies. The data will be shared between companies.
(2) Ranking of the influencing factors	(2.1) Classification of influencing factors into preference groups (2.2) Rank ordering of influencing factors	
(3) Ranking of the executing units	(3.1) Definition and characterisation of offices (3.2) Classification of offices into preference groups (3.3) Rank ordering of offices	Individual per company.

this, we conducted a survey with 111 professionals who played the role of software project manager in several organizations spread throughout Brazil. The survey questions look like this:

(1) *Factor.* Technical skills – knowledge and ability on techniques, programming languages, frameworks, tools, APIs, necessary for the professionals to carry out the task.

(1.1) Requirements-Related Tasks.

Criteria\Values	1. High gain	2. Some gain	3. No gain
A. Quality of the task	A1. ()	A2. ()	A3. ()
B. Time of the task	B1. ()	B2. ()	B3. ()
C. Cost of the task	C1. ()	C2. ()	C3. ()

After conducting the survey, we were able to characterise the alternatives (in this stage, the influencing factors), which served as inputs for the VDA methods. An example of alternatives characterisation can be seen in Table 6, where the cells in bold represent the most voted criteria values and, therefore, the ones that were considered in the final characterisation vector (note the final vector column). The numbers in the cells represent the total votes each criterion value received for that factor.

5.2.2. *Ranking of the Influencing Factors.* After defining the criteria and criteria values, alternatives (factors) and preference groups, we moved to the Stage of Ranking of the Influencing Factors. As explained, first, the ORCLASS method was applied with the support of the ORCLASSWEB tool, following the steps listed below:

- (1) Definition of the criteria and criteria values;
- (2) Definition of the alternatives;
- (3) Construction of the classification rule; and
- (4) Results generation.

We performed this stage once for each of the four task types determined initially. First, we introduced the problem's criteria into the ORCLASSWEB. In this step, we specified the criteria's names and their possible values. Next, we entered the problem's alternatives into the ORCLASSWEB. We provided the alternatives' names and their representations in terms of criteria values, considering the criteria established in the previous step (values in the column "final vector" in Table 6). The ORCLASSWEB then builds the classification rules by asking questions to the decision-maker. At that time, we got the support of five project management experts with more than ten years of experience in software development and that have a master's degree. They all lived in Fortaleza (Brazil) and worked for different organizations. They played the role of

TABLE 6: Example of alternatives characterisation for requirements tasks.

Criteria/alternatives	Quality of the task			Time of the task			Cost of the task			Final vector
	A1	A2	A3	B1	B2	B3	C1	C2	C3	
Factor1	61	40	10	43	56	12	39	48	24	A1B2C2
Factor2	97	10	4	81	27	3	75	33	3	A1B1C1
Factor3	52	55	4	56	49	6	51	50	10	A2B1C1
	...									

decision-makers and answered the questions of the method to classify the alternatives. The series of responses build the classification rule, which allows separating the alternatives into groups. Finally, the ORCLASSWEB classified all the alternatives and provided the results. For example, in our case, for requirement tasks, the A1B2C2 alternative fell in the group I while the A2B2C2 alternative was classified in group II.

At the end of ORCLASSWEB application, the factors were distributed between group I (the preferential factors) and group II (the nonpreferable factors) for each type of task (see Table 7). Factors from group I are the most relevant ones that project managers should take into account when assigning tasks in DSD projects, whereas factors from group II are the least relevant ones and should be less considered or even not considered when attributing tasks in DSD environment.

Once the preferable factors are identified by applying the ORCLASS method, we advanced to the step of ordering in the Stage of Ranking of the Influencing Factors. In this step, we employed the ZAPROS III-i method to order the preferable factors so as to establish a ranking of them. As already explained, the least preferable factors should be discarded in order to reduce our workspace. Like the previous step, this one was performed once for each of the four defined types of task.

As explained previously, the ZAPROS III-i method was performed with the support of the ARANAÚ software, following the steps described below:

- (1) Definition of the criteria and criteria values;
- (2) Elicitation of preferences;
- (3) Definition of the alternatives; and
- (4) Results generation.

Initially, we inserted the criteria into the ARANAÚ. Next, the same group of five project management experts from the previous step, in the role of decision-makers, decided the preferences in order to formulate the scale of preferences. The procedure happens in two steps: preferences elicitation for quality variation of the same criterion and preferences elicitation between pairs of criteria. After the preference scale has been formulated, the alternatives of the problem must be defined. The alternatives of the problem under study are the factors that constitute the group I, i.e., the factors that should be preferentially regarded in the allocation of tasks in DSD projects.

Once the data were entered and all the questions of the model were answered, ARANAU executed the process according to ZAPROS III-i, generating the outputs at the

end, that is, an ordered list of the most relevant factors that the project managers should think about when distributing tasks in DSD projects. In our study, once we performed the model for all task types, ARANAÚ produced four ordered scales. The tool shows the alternatives of the problem, its representations in terms of criteria values and their rankings.

Note that, among the fifteen factors initially considered, for the tasks related to requirements, eleven factors were classified in group I; that is, they were indicated as preferable, whereas another four factors were classified in group 2, that is, not preferable. The reasoning is the same for all other types of tasks. Table 8 shows the classification summary after the application of the ORCLASS method. The rank column indicates the position of the influencing factor on the preference scale generated by the model. For example, for requirement-related tasks, Factors 2, 4, 5, 6, 7, 11, 13, 14, and 15 were ranked at position 1; after all, they all have the best representation, i.e., A1B1C1.

The ARANAÚ also draw a graph showing the dominance relations among the alternatives. For all task types, we created the corresponding dominance graphs, which are exhibited in Figure 3. The arrow direction indicates dominance. It is important to emphasise that all such knowledge produced at this stage of the experiences of use can be reused in companies that intend to use the approach proposed in this research. Thus, a company may decide not to perform stage 1, Context Characterisation, and adopt the characterisation of the influencing factors generated by this work. Likewise, a company may also choose not to perform Stage 2, Ranking of the Influencing Factors, and thus make use of the ranking of influencing factors generated in this work. We understand that this ranking of influencing factors constitutes one of the significant contributions of this work since it is the result of very comprehensive research, which had the participation of more than a hundred professionals from the project management area in Brazil.

5.2.3. Ranking of the Executing Units. Once we completed the stages that are common for all the companies involved and have selected the main influencing factors by task type, we moved to the individualized stage per company (Stage 3). In this step, the companies' offices served as alternatives for the VDA classification (ORCLASS) and ordering (ZAPROS III-i) methods. The set of the preferable influencing factors resulting from Stage 2 served as criteria to evaluate the offices.

For this study, we defined that each criterion could be evaluated as excellent, good, or regular. Thus, these were the criteria values adopted for this process stage. Once the criteria and their values have been defined, we proceeded to

TABLE 7: Results of ORCLASS application separated by task types and preference groups.

		Group I - The Preferable Alternatives			Group II - The Non-preferable Alternatives		
		Criteria Values	Factors	Criteria Values	Factors	Criteria Values	Factors
Requirements	A1B1C1	Factor2 - Knowledge in business	A2B2C2	Factor8 - Team personal trust	Factor3 - Project manager maturity	Factor2 - Knowledge in business	Factor8 - Team personal trust
		Factor4 - Proximity to the customer	A3B3C1	Factor10 - Cultural similarities	Factor4 - Proximity to the customer	Factor10 - Cultural similarities	Factor12 - Low labour cost
		Factor5 - Low turnover rate		Factor12 - Low labour cost			
		Factor6 - Team availability					
		Factor7 - Team maturity					
	A2B1C1 A1B1C2 A1B2C2	Factor11 - Team willingness					
		Factor13 - Maturity in the process					
		Factor14 - Language fluency					
		Factor15 - Good communic. infra.					
		Factor3 - Project manager maturity					
Architecture	A1B1C1	Factor9 - Same time zone					
		Factor1 - Technical skills					
		Factor4 - Technical skills					
		Factor5 - Low turnover rate					
		Factor7 - Team maturity	A2B2C2	Factor3 - Project manager maturity	Factor4 - Proximity to the customer	Factor2 - Knowledge in business	Factor8 - Team personal trust
	A1B1C2 A1B2C2	Factor6 - Team availability		Factor8 - Team personal trust	Factor9 - Same time zone	Factor3 - Project manager maturity	Factor4 - Proximity to the customer
		Factor11 - Team willingness		Factor10 - Cultural similarities	Factor12 - Low labour cost	Factor8 - Team personal trust	Factor9 - Same time zone
		Factor13 - Maturity in the process	A2B2C3	Factor11 - Team willingness	Factor12 - Low labour cost	Factor9 - Same time zone	Factor10 - Cultural similarities
		Factor14 - Language fluency	A3B3C1	Factor15 - Good communic. infra.	Factor13 - Maturity in the process	Factor10 - Cultural similarities	Factor12 - Low labour cost
		Factor15 - Good communic. infra.					
Implementation	A1B1C1	Factors		Criteria Values	Factors	Criteria Values	Factors
		Factor1 - Technical expertise		Factor3 - Project manager maturity	Factor4 - Proximity to the customer	Factor2 - Knowledge in business	Factor8 - Team personal trust
		Factor5 - Low turnover rate	A2B2C2	Factor8 - Team personal trust	Factor9 - Same time zone	Factor3 - Project manager maturity	Factor4 - Proximity to the customer
		Factor7 - Team maturity		Factor9 - Same time zone	Factor10 - Cultural similarities	Factor8 - Team personal trust	Factor9 - Same time zone
		Factor6 - Team availability		Factor12 - Low labour cost	Factor13 - Maturity in the process	Factor9 - Same time zone	Factor10 - Cultural similarities
	A1B1C2 A1B2C2	Factor11 - Team willingness		Factor15 - Good communic. infra.	Factor14 - Language fluency	Factor11 - Team willingness	Factor12 - Low labour cost
		Factor13 - Maturity in the process		Factor15 - Good communic. infra.	Factor14 - Language fluency	Factor13 - Maturity in the process	Factor14 - Language fluency
		Factor14 - Language fluency		Factor15 - Good communic. infra.	Factor14 - Language fluency	Factor15 - Good communic. infra.	Factor15 - Good communic. infra.
		Factor15 - Good communic. infra.					
Tests	A1B1C1	Criteria Values	Factors	Criteria Values	Factors	Criteria Values	Factors
		Factor5 - Low turnover rate	A2B2C2	Factor8 - Team personal trust	Factor3 - Project manager maturity	Factor2 - Knowledge in business	Factor8 - Team personal trust
	A1B1C2 A1B2C2	Factor7 - Maturity of the team		Factor9 - Same time zone	Factor4 - Proximity to the customer	Factor3 - Project manager maturity	Factor4 - Proximity to the customer
		Factor5 - Good communic. infra.		Factor10 - Cultural similarities	Factor11 - Team willingness	Factor5 - Good communic. infra.	Factor6 - Team availability

TABLE 8: Detailed results of ARANAÚ application for each type of tasks.

	Rank	Alternative	Quality	Representation	Time	Cost
Requirements	1	Factor2 - Knowledge of business	A1. High gain	B1. High gain	C1. High gain	
	1	Factor4 - Proximity to the customer	A1. High gain	B1. High gain	C1. High gain	
	1	Factor5 - Low turnover rate	A1. High gain	B1. High gain	C1. High gain	
	1	Factor6 - Availability	A1. High gain	B1. High gain	C1. High gain	
	1	Factor7 - Maturity of the team	A1. High gain	B1. High gain	C1. High gain	
	1	Factor11 - Willingness at site	A1. High gain	B1. High gain	C1. High gain	
	1	Factor13 - Maturity in the process	A1. High gain	B1. High gain	C1. High gain	
	1	Factor14 - Language fluency	A1. High gain	B1. High gain	C1. High gain	
	1	Factor15 - Good communication infrastructure	A1. High gain	B1. High gain	C1. High gain	
	2	Factor9 - Same time zone	A1. High gain	B1. High gain	C2. Some gain	
Architecture	3	Factor3 - Maturity of the project manager	A2. Some gain	B1. High gain	C1. High gain	
	4	Factor1 - Technical skills	A1. High gain	B2. Some gain	C2. Some gain	
	Rank	Alternative	Quality	Representation	Time	Cost
Architecture	1	Factor1 - Technical skills	A1. High gain	B1. High gain	C1. High gain	
	1	Factor5 - Low turnover rate	A1. High gain	B1. High gain	C1. High gain	
	1	Factor7 - Maturity of the team	A1. High gain	B1. High gain	C1. High gain	
	2	Factor6 - Availability	A1. High gain	B1. High gain	C2. Some gain	
	2	Factor11 - Willingness at site	A1. High gain	B1. High gain	C2. Some gain	
	2	Factor13 - Maturity in the process	A1. High gain	B1. High gain	C2. Some gain	
	2	Factor14 - Language fluency	A1. High gain	B1. High gain	C2. Some gain	
	2	Factor15 - Good communication infrastructure	A1. High gain	B1. High gain	C2. Some gain	
	Rank	Alternative	Quality	Representation	Time	Cost
Implementation	1	Factor1 - Technical expertise	A1. High gain	B1. High gain	C1. High gain	
	1	Factor5 - Low turnover rate	A1. High gain	B1. High gain	C1. High gain	
	1	Factor6 - Availability	A1. High gain	B1. High gain	C1. High gain	
	1	Factor7 - Site maturity	A1. High gain	B1. High gain	C1. High gain	
	1	Factor11 - Willingness at site	A1. High gain	B1. High gain	C1. High gain	
	1	Factor13 - Maturity in the process	A1. High gain	B1. High gain	C1. High gain	
	1	Factor15 - Good communication infrastructure	A1. High gain	B1. High gain	C1. High gain	
	2	Factor14 - Language fluency	A1. High gain	B1. High gain	C2. Some gain	
	3	Factor2 - Expertise in business	A1. High gain	B2. Some gain	C2. Some gain	
	Rank	Alternative	Quality	Representation	Time	Cost
Tests	1	Factor5 - Low turnover rate	A1. High gain	B1. High gain	C1. High gain	
	1	Factor7 - Maturity of the team	A1. High gain	B1. High gain	C1. High gain	
	1	Factor15 - Good communication infrastructure	A1. High gain	B1. High gain	C1. High gain	
	2	Factor6 - Availability	A1. High gain	B1. High gain	C2. Some gain	
	2	Factor11 - Willingness at site	A1. High gain	B1. High gain	C2. Some gain	
	2	Factor13 - Maturity in the process	A1. High gain	B1. High gain	C2. Some gain	
	2	Factor14 - Language fluency	A1. High gain	B1. High gain	C2. Some gain	
	3	Factor1 - Technical skills	A1. High gain	B2. Some gain	C2. Some gain	
	3	Factor2 - Knowledge of business	A1. High gain	B2. Some gain	C2. Some gain	
	3	Factor4 - Proximity to the customer	A1. High gain	B2. Some gain	C2. Some gain	

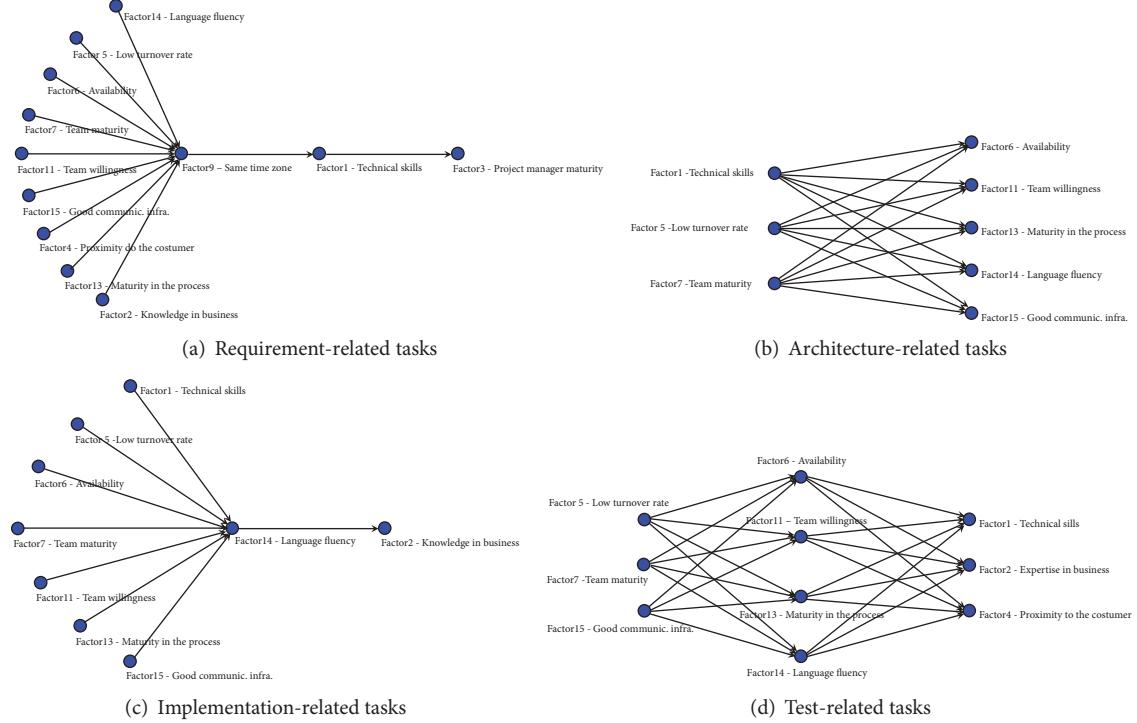


FIGURE 3: Dominance relations among the alternatives by tasks type.

the characterisation of the offices as regards the influencing factors, we mean; considering the influencing factors as criteria and the company offices as alternatives, we wanted to know how each company office was evaluated against the influencing factors. That characterisation will be useful for the next steps. The companies' professionals were then asked to rate all influencing factors for the offices of the company he or she works for, according to the following questions template. We replicated the questions to the other offices and factors.

(1) *Office Located in São Paulo.*

Influencing Factor	Excellent	Good	Regular
Factor 1 – Technical skills: knowledge and ability on techniques, programming languages, frameworks, tools, APIs, necessary for the professionals to carry out the task.	()	()	()
Factor 2 – Knowledge in business: Knowledge of the teams about the area, domain or business of the clients.	()	()	()
Factor 3 ...			

The survey continues with the other offices.

After characterising the offices, we ran ORCLASS and ZAPROS III-I methods to get the ordered lists of recommended offices for each task type. Figure 4 shows the

ORCLASSWEB screen for assembling the classification rule for the ORCLASS method. In this step, PMO staff, portfolio managers, or both should play the role of decision-maker and answer the questions posed by the model in order to classify the alternatives.

They should respond "Yes" or "No" for all questions. The series of answers contribute to assembling the classification rule. Thus, it is possible to divide the alternatives (influencing factors) into preference groups. A positive response classifies that combination into the preferable group. Otherwise, the combination goes to the nonpreferable group. Thereby, the classification rule will be completed according to the decision-makers selections. At last, the ORCLASSWEB software processes the complete classification of the alternatives.

Figure 5 shows the ARANAÚ interface for the preferences elicitation between pairs of criteria as part of ZAPROS III-I method application. In this scenario, the decision-maker should choose the best alternative, considering that the other values are at their best or worst degree. The choices set the scale of preferences among the criteria.

5.3. Results. In the end, after applying DiVA for the five companies, we obtained the results (see Table 9). A detailed description of these experiences of use can be found in [69].

5.4. Evaluation of the Proposed Approach. To evaluate the DiVA approach, we performed two types of analysis. The first evaluation sought to analyse the approach from the perspective of the five project management experts. The other analysis aimed to evaluate the approach by the companies'

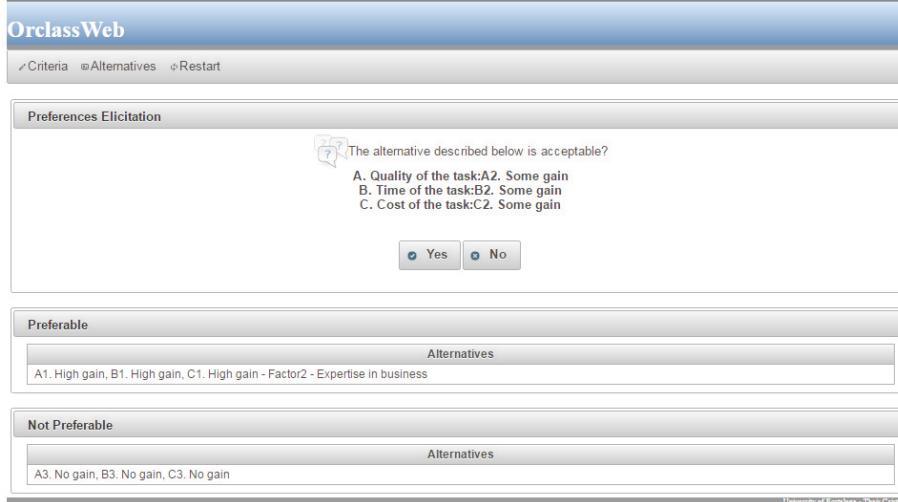


FIGURE 4: ORCLASSWEB's interface for construction of the classification rule.

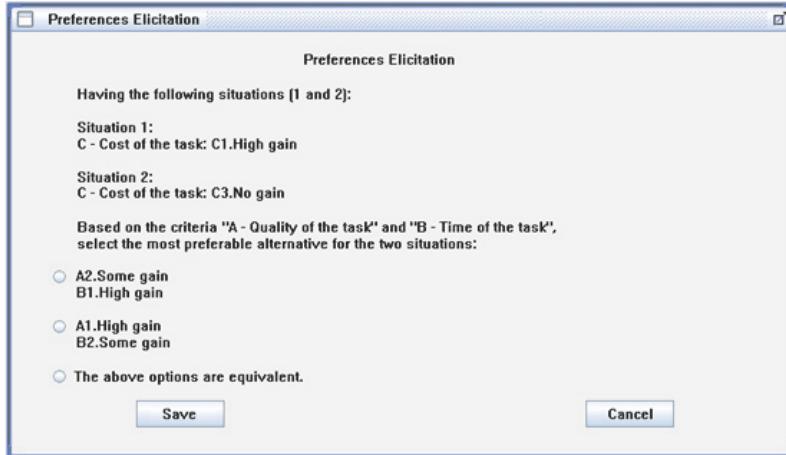


FIGURE 5: ARANAÚ's interface for structuring the problem scale of preferences between pairs of criteria.

professionals who participated in the experiences of use. Thirteen professionals from the companies participated in the experiences of use. However, three of them abstained from collaborating with the evaluation questionnaire. Table 10 summarizes the responses of both evaluations, which are briefly discussed in Section 5.5.

5.5. Discussion. In both evaluations, most respondents agreed that the approach is easy to understand and to use. Furthermore, most of them also agreed that the approach helps reduce subjectivity and helps think of new aspects. Finally, most participants evaluated that it helps in task allocation in DSD. As points of attention, the assessment of the adaptability/extensibility and of the effort to use did not allow drawing precise conclusions.

Analysing the proposed approach based on the experiences of use conducted and the evaluations carried out, we have the following about DiVA:

- (i) It describes a process, providing a visual flowchart, which makes easy its understanding;
- (ii) It is supported by automated tools (ORCLASSWEB and ARANAÚ), facilitating its application;
- (iii) It uses influence factors extracted from the specialized literature, bringing global knowledge to the process;
- (iv) It can be instantiated from an organization standard; the experiences of use carried out followed this concept; that is, in the beginning, a common structure was defined, and, then, each company entered with its respective data. In the same way, this can be done within an organization, creating a common base and then customizing for each project.

TABLE 9: Results after applying DiVA.

Company	Order	Requirements related tasks	Architecture/Design related tasks	Implementation related tasks	Testing related tasks
1	1	Fortaleza (BR)	São Paulo (BR)	São Paulo (BR)	São Paulo (BR)
	2	São Paulo (BR)	Fortaleza (BR)	Fortaleza (BR)	Fortaleza (BR)
	3	Quixadá (BR)	Quixadá (BR)	Quixadá (BR)	Quixadá (BR)
2	1	Recife (BR)	Natal (BR)	Natal (BR)	Recife (BR)
	2	Fortaleza (BR)	Recife (BR)	Fortaleza (BR)	Natal (BR)
	3	Natal (BR)	Fortaleza (BR)	Recife (BR)	Fortaleza (BR)
3	1	São Paulo (BR)	São Paulo (BR)	São Paulo (BR)	São Paulo (BR)
	2	Rio de Janeiro (BR)	Rio de Janeiro (BR)	Rio de Janeiro (BR)	Rio de Janeiro (BR)
	3	Porto Alegre (BR)	Porto Alegre (BR)	Porto Alegre (BR)	Porto Alegre (BR)
	4	Fortaleza (BR)	Fortaleza (BR)	Fortaleza (BR)	Fortaleza (BR)
4	1	New York (USA)	Buenos Aires (ARG)	New York (USA)	New York (USA)
	2	Buenos Aires (ARG)	Fortaleza (BR)	Buenos Aires (ARG)	Buenos Aires (ARG)
	3	Santiago (CHI)	New York (USA)	Santiago (CHI)	São Paulo (BR)
	4	Fortaleza (BR)	São Paulo (BR)	Fortaleza (BR)	Santiago (CHI)
	5	São Paulo (BR)	Santiago (CHI)	São Paulo (BR)	Fortaleza (BR)
5	1	Aracaju (BR)	Aracaju (BR)	Aracaju (BR)	Aracaju (BR)
	2	Juiz de Fora (BR)	Campina Grande (BR)	Juiz de Fora (BR)	Jaguariúna (BR)
	3	Curitiba (BR)	Juiz de Fora (BR)	Campina Grande (BR)	Curitiba (BR)
	4	Brasília (BR)	Rio de Janeiro (BR)	Jaguariúna (BR)	Juiz de Fora (BR)
	5	Jaguariúna (BR)	Curitiba (BR)	Curitiba (BR)	Campina Grande (BR)
	6	Campina Grande (BR)	Jaguariúna (BR)	Rio de Janeiro (BR)	Rio de Janeiro (BR)
	7	Belém (BR)	Porto Alegre (BR)	Belém (BR)	Belém (BR)

- (v) It reuses knowledge because it allows the use of knowledge from other sources (factors, criteria, relations, among others); besides, it is possible to feedback the approach with the generated knowledge itself;
- (vi) It presupposes consensus meetings since it provides for meetings between the company's specialists to analyse and evaluate the recommendations suggested by the approach;
- (vii) It presupposes feedback of results as its flow provides for feedback of information and results to calibrate the model and improve decisions;
- (viii) It proposes to be flexible and adaptable, since it has a modular structure, which makes it possible to use

of a subset of the whole process, besides allowing the modification of the main parameters (object of allocation and executing unit).

In addition, DiVA was applied in real-world situations, since the experiences of use were conducted with the collaboration of five software companies with distributed structure, and both knowledge area specialists and business professionals evaluated it.

5.6. Threats and Limitations

5.6.1. *Threats to Validity.* Possible threats were identified during the execution of the experiences of use. The first point that may have influenced the results of the experiences of use

TABLE 10: Evaluation of the approach by the project management experts (PME) and by the companies' professionals (CP).

	Strongly disagree PME	CP	PME	Disagree CP	PME	Neither agree nor disagree PME	CP	PME	Agree CP	PME	CP	Strongly agree PME	CP
RQ1. Easy to understand	-	1	-	1	-	1	1	5	7	-	-	-	-
RQ2. Easy to use	-	1	-	1	1	1	1	2	6	2	1	2	1
RQ3. Adaptable/extensible	-	-	-	-	1	1	5	1	4	3	1	3	1
RQ4. Low effort to use	-	1	2	2	2	4	-	-	3	2	1	2	1
RQ5. Helps in reducing subjectivity	-	-	1	-	-	-	3	8	8	2	1	2	1
RQ6. Helps in thinking of new aspects	-	-	1	-	1	1	3	5	5	2	1	3	1
RQ7. Helps in task allocation in DSD	-	-	-	-	3	1	1	7	4	-	-	-	-

is the fact that the author of the work was the manager of the process executed in the experiences of use. Prior knowledge of the objectives proposed for DiVA may have had a positive influence on the obtained results. However, it was not possible to adopt an action capable of mitigating this bias, since ideally some professional from each company who had not been involved in this study should have executed DiVA. This was not possible due to staff restrictions and employees' workload in the companies. Thus, the directors requested that the author of the research conducted the experiences of use.

For obvious reasons of time restriction of companies' employees, it was not possible to perform the entire approach within the companies. Thus, we have chosen to structure the application of the approach in two phases. This may have caused some confusion to the companies' professionals for not having participated in the selection of the influencing factors. We tried to mitigate this bias by explaining the whole approach in the questionnaires applied to professionals. The analysis of the results obtained with the application of DiVA by the author of this research can also be considered a threat. To mitigate that threat, the analysis of the results obtained with the execution of the DiVA and the evaluations made by companies' professionals and project management experts were discussed by two other researchers and by two project managers who were not involved with the research. Moreover, the analysis, although subjective, has taken into consideration only the answers regarding the agreement or not with certain statements related to the objectives of the research and can be easily verified from the results presented in the tables throughout the text of the paper.

Finally, another threat of this study concerns the generalization of the results. Once it has been performed in a limited way in a restricted group of companies, it is not allowed to generalize the results. It is only possible to observe indications about how the application of DiVA supports organizations in the task allocation process.

5.6.2. Limitations. The user experiences have been developed partly in collaboration with the professionals of the participating companies. Better results could have been obtained if performed entirely by the companies' professionals. It was also not possible to obtain information on the time, cost, and effort of applying DiVA, due to the circumstances reported. Furthermore, another important limitation perceived was the lack of an integrated automated tool. Although there are specific automated tools for the VDA methods adopted, an integrated tool comprising the entire process would likely bring more benefits to the decision-making process as well as better results in approach evaluation.

6. Conclusion and Future Works

The software industry has a high level of competition, which requires IT companies to constantly search for innovative models of management, service supply and software development. On the other hand, the IT companies market, more than in any segment, is global; that is, customers can be spread all over the world. An increasing trend is to create subsidiaries around the world and deliver activities to them to benefit

from the best features of each location and thus achieve the gains that the distributed model can provide. However, that solution presents challenges because many issues should be taken into consideration when distributing tasks to remote teams. The context exposed here is a typical decision-making problem that can be structured on multiple criteria. This problem can be tackled by using Verbal Decision Analysis.

In that scenario, this research introduced DiVA, an approach to support the allocation of tasks in DSD projects, based on the hybridisation of methods of Verbal Decision Analysis for classification and ordering applied to influencing factors and executing units. In this way, we intend to aid organizations in task allocation activity, which as presented throughout this work is very critical and involves many difficulties.

Aligned with this objective, a review of the literature was conducted to find out the approaches that support the allocation of tasks in DSD environment. Thus, it was possible to obtain an overview of the existing solutions that proposed to tackle the problem of task assignment in DSD. The literature review also allowed identifying some gaps that were addressed in the proposed approach, named DiVA. For example, the results of the review showed that few solutions proposed by other studies were based on multicriteria decision-making methods and even fewer studies were based on qualitative analysis. Furthermore, no approach based on Verbal Decision Analysis methods was identified.

Thus, for the development of the approach, we applied a hybrid model based on ORCLASS (for classification) and ZAPROS III-i (for ordering) methods, both from Verbal Decision Analysis framework, firstly regarding the factors that influence the task attribution in DSD, and, later, concerning the executing units (teams, offices, or branches). Bibliographic research and application of surveys with professionals allowed identifying and characterising the main factors that influence task assignment in DSD projects.

Experiences were carried out with the collaboration of professionals from five real-world companies that have a distributed structure. For that, tasks were gathered based on their type, that is, requirements, architecture/design, implementation, and testing. In the end, DiVA has been submitted to the evaluation by the professionals of the companies that participated in the experiences of use. In addition, DiVA has also been evaluated by project management experts with extensive experience in the knowledge field being studied.

Several conclusions can be drawn from the use of the approach, among which we can mention: it is supported by automated tools, it uses influence factors extracted from the specialized literature, and it proposes to be flexible and adaptable, since it is possible to use a subset of the whole process, besides allowing the modification of the main parameters (object of allocation and executing unit). Considering the evaluations of the approach, the results provided evidence that the approach is easy to understand and to use and helps to reduce the decision subjectivity and to think of new aspects.

According to [70], in research based on the Design Science Research (DSR), the research method adopted in this work, we seek to find a solution to a particular problem. This solution should not necessarily be optimal but at

least satisfactory for the context it proposes. In addition, it is accepted that an actual problem and the artefacts that promote satisfactory solutions to it can share common characteristics that allow its generalization to a determinate category of problems. In this context, based on the results of the evaluations carried out by project management experts from different organizations and by professionals from the companies participating in the experiences of use, it is considered that DiVA approach can support software organizations in allocating tasks in DSD projects.

6.1. Future Works. The application of DiVA in real contexts and the evaluation of the results of the approach by professionals of the companies participating in the experiences of use, besides the evaluation of the approach by the specialists, allowed identifying some opportunities for improvement, which we suggest as future works (i) to develop an automated suite aiming the integration of the ORCLASSWEB and ARANAÚ tools in order to make its application easy and minimize the effort to adopt the approach; (ii) to hybridise other classification methods (NORCLASS or SAC) with other ordering methods (PACOM, SNOD, ORCON, or ARACE) to compose variants of DiVA; (iii) to create a knowledge base on factors of influence that would enable companies to select the most convenient factors depending on the context; (iv) to provide a guide or tutorial with information on applying the approach, describing “step by step” the execution of each activity; (v) to apply DiVA considering a reduced set of influencing factors, chosen as more appropriate for each company; (vi) to adapt DiVA to support the allocation of tasks for individual projects; and, finally, (vii) to adapt DiVA to consider dependency relations between tasks.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The first author is thankful for the support given by the 7 de Setembro University Center during this project. The authors would like to thank the Edson Queiroz Foundation/University of Fortaleza for all the support. The authors would also like to thank the 11th Iberian Conference on Information Systems and Technologies (CISTI) for having published one of the embryonic articles of their research. The second author is grateful to the National Counsel of Technological and Scientific Development (CNPq) via Grants #305844/2011-3.

References

- [1] A. Lamersdorf, J. Münch, and D. Rombach, “Towards a multi-criteria development distribution model: an analysis of existing task distribution approaches,” in *Proceedings of the 3rd IEEE International Conference on Global Software Engineering (ICGSE '08)*, pp. 109–118, Bangalore, India, August 2008.
- [2] O. I. Larichev and R. V. Brown, “Numerical and verbal decision analysis: Comparison on practical cases,” *Journal of Multi-Criteria Decision Analysis*, vol. 9, no. 6, pp. 263–273, 2000.
- [3] O. I. Larichev and H. M. Moshkovitch, *Verbal Decision Analysis for Unstructured Problems*, Kluwer Academic Publishers, Boston, 2013.
- [4] Project Management Institute, *A Guide to Project Management Knowledge (PMBOK Guide)*, 2013.
- [5] A. B. Marques, J. R. Carvalho, R. Rodrigues, T. Conte, R. Prikladnicki, and S. Marczak, “An ontology for task allocation to teams in distributed software development,” in *Proceedings of the IEEE 8th International Conference on Global Software Engineering (ICGSE '13)*, pp. 21–30, Bari, Italy, August 2013.
- [6] C. A. Bana e Costa, R. Sanchez-Lopez, J. C. Vansnick, and J. M. De Corte, “Introducción a MACBETH,” in *Análisis Multicriterio para la Toma de Decisiones: Métodos y Aplicaciones*, Mexico, 2011.
- [7] C. Evangelou, N. Karacapilidis, and O. A. Khaled, “Interweaving knowledge management, argumentation and decision making in a collaborative setting: the KAD ontology model,” *International Journal of Knowledge and Learning*, vol. 1, no. 1-2, pp. 130–145, 2005.
- [8] O. I. Larichev, A. V. Kortnev, and D. Y. Kochin, “Decision support system for classification of a finite set of multicriteria alternatives,” *Decision Support Systems*, vol. 33, no. 1, pp. 13–21, 2002.
- [9] H. M. Moshkovitch and A. I. Mechitov, “Verbal decision analysis: foundations and trends,” *Advances in Decision Sciences*, vol. 2013, Article ID 697072, 9 pages, 2013.
- [10] D. P. Oleynikov, L. N. Butenko, H. Moshkovitch, and A. Mechitov, “ARACE – A New Method for Verbal Decision Analysis,” *International journal of information technology & decision making*, vol. 14, no. 1, pp. 115–140, 2015.
- [11] O. I. Larichev, “Ranking multicriteria alternatives: the method ZAPROS III,” *European Journal of Operational Research*, vol. 131, no. 3, pp. 550–558, 2001.
- [12] T. C. S. Machado, P. R. Pinheiro, A. B. Albuquerque, and M. M. L. de Lima, “Applying verbal decision analysis in selecting specific practices of CMMI,” in *Proceedings of the International Conference on Rough Sets and Knowledge Technology*, vol. 7414, pp. 215–221, Springer Berlin Heidelberg, 2012.
- [13] M. Mendes, A. L. Carvalho, E. Furtado, and P. Pinheiro, “A co-evolutionary interaction design of digital TV applications based on verbal decision analysis of user experiences,” *International Journal of Digital Culture and Electronic Tourism*, vol. 1, no. 4, p. 312, 2009.
- [14] I. Tamanini, T. C. Sampaio Machado, M. S. Mendes, A. L. Carvalho, M. E. S. Furtado, and P. R. Pinheiro, “A model for mobile television applications based on verbal decision analysis,” in *Proceedings of International Conference on Systems, Advances in Computer and Information Sciences and Engineering*, pp. 399–404, Springer, Dordrecht, Netherlands, 2008.
- [15] T. Machado, *Towards aided by multicriteria support methods and software development: a hybrid model of verbal decision analysis*

- for selecting approaches of project management*, University of Fortaleza, 2012.
- [16] T. C. S. Machado, P. R. Pinheiro, and I. Tamanini, "Project management aided by verbal decision analysis approaches: a case study for the selection of the best SCRUM practices," *International Transactions in Operational Research*, 2014.
 - [17] I. Tamanini, A. L. Carvalho, A. K. Castro, and P. R. Pinheiro, "A novel multicriteria model applied to cashew chestnut industrialization process," *Advances in Intelligent and Soft Computing*, vol. 58, pp. 243–252, 2009.
 - [18] I. Tamanini, A. K. de Castro, P. R. Pinheiro, and M. C. Pinheiro, "Towards an applied multicriteria model to the diagnosis of Alzheimer's disease: a neuroimaging study case," in *Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS '09)*, pp. 652–656, Shanghai, China, 2009.
 - [19] I. Tamanini, A. K. de Castro, P. R. Pinheiro, and M. C. D. Pinheiro, "Verbal decision analysis applied on the optimization of alzheimer's disease diagnosis: a case study based on neuroimaging," *Advances in Experimental Medicine and Biology*, vol. 696, pp. 555–564, 2011.
 - [20] A. K. Araújo de, P. R. Castro, and M. C. Dantas Pinheiro, "Applying a Decision Making Model in the Early Diagnosis of Alzheimer's Disease," in *Rough Sets and Knowledge Technology*, pp. 149–156, Springer, Berlin, Germany.
 - [21] P. R. Pinheiro, A. K. Castro, and M. C. Pinheiro, "A multicriteria model applied in the diagnosis of alzheimer's disease: a bayesian network," in *Proceedings of the IEEE 11th International Conference on Computational Science and Engineering (CSE '08)*, pp. 15–22, 2008.
 - [22] A. K. Castro, P. R. Pinheiro, M. C. Pinheiro, and I. Tamanini, "Applied hybrid model in the neuropsychological diagnosis of the Alzheimer's disease: a decision making study case," *International Journal of Social and Humanistic Computing*, vol. 1, no. 3, p. 331, 2010.
 - [23] L. F. A. M. Gomes, H. Moshkovich, and A. Torres, "Marketing decisions in small businesses: how verbal decision analysis can help," *International Journal of Management and Decision Making*, vol. 11, no. 1, pp. 19–36, 2010.
 - [24] P. A. Barbosa, P. R. Pinheiro, F. R. de Vasconcelos Silveira, and M. Simão Filho, "Applying verbal analysis of decision to prioritize software requirement considering the stability of the requirement," in *Proceedings of the Computer Science Online Conference (CSOC '17)*, vol. 575, pp. 416–426, Springer International Publishing.
 - [25] M. Simão Filho, P. R. Pinheiro, and U. R. P. Gomes, "Project portfolio prioritization aided by verbal decision analysis," in *Proceedings of the 13th Iberian Conference on Information Systems and Technologies, CISTI '18*, pp. 1–6, Spain, 2018.
 - [26] T. C. S. Machado, P. R. Pinheiro, and I. Tamanini, "OrclassWeb: a tool based on the classification methodology ORCLASS from verbal decision analysis framework," *Mathematical Problems in Engineering*, vol. 2014, Article ID 238168, 11 pages, 2014.
 - [27] L. Ustinovich and D. Kochin, "Verbal decision analysis methods for determining the efficiency of investments in construction," *European Journal of Environmental and Civil Engineering*, vol. 5, no. 1, pp. 35–46, 2004.
 - [28] I. Tamanini and P. R. Pinheiro, "Reducing incomparability in multicriteria decision analysis: an extension of the ZAPROS method," *Pesquisa Operacional*, vol. 31, no. 2, pp. 251–270, 2011.
 - [29] I. Tamanini and P. Pinheiro, "Applying a new approach methodology with ZAPROS," *XL Brazilian Symposium on Operational Research*, pp. 914–925, 2008.
 - [30] I. Tamanini, *Improving the ZAPROS method considering the incomparability cases*, Master Thesis, University of Fortaleza, 2010.
 - [31] M. Simão Filho, P. R. Pinheiro, A. B. Albuquerque, and J. J. Rodrigues, "Task allocation in distributed software development: a systematic literature review," *Complexity*, vol. 2018, Article ID 6071718, 13 pages, 2018.
 - [32] A. Lamersdorf and J. Münch, "TAMRI: A tool for supporting task distribution in global software development projects," in *Proceedings of the 4th IEEE International Conference on Global Software Engineering, ICGSE '09*, pp. 322–327, Ireland, 2009.
 - [33] M. Ruano-Mayoral, C. Casado-Lumbreras, H. Garbarino-Alberti, and S. Misra, "Methodological framework for the allocation of work packages in global software development," *Journal of Software: Evolution and Process*, vol. 26, no. 5, pp. 476–487, 2014.
 - [34] L. Almeida and A. Albuquerque, "A Multi-criteria Model for Planning and Fine-Tuning Distributed Scrum Projects," *IEEE Sixth International Conference on Global Software Engineering*, pp. 75–83, 2011.
 - [35] L. H. Almeida, P. R. Pinheiro, and A. B. Albuquerque, "Applying multi-criteria decision analysis to global software development with scrum project planning," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6954, pp. 311–320, 2011.
 - [36] M. Pedras, *Uma ferramenta de apoio ao gerenciamento de desenvolvimento de software distribuído*, Federal University of Paraná, 2003.
 - [37] M. Pedras, E. Huzita, T. Tait, and G. Santiago, "Dimanager: a tool for distributed software development management," in *Proceedings of the International Conference on Enterprise Information Systems*, 2004.
 - [38] E. Huzita, T. Tait, and T. Colanzi, "Um ambiente de desenvolvimento distribuído de software-disen," in *Proceedings of the Brazilian Workshop on Distributed Software Development*, 2007.
 - [39] E. Huzita, C. Silva, and I. Wiese, "Um conjunto de soluções para apoiar o desenvolvimento distribuído de software," in *II Brazilian Workshop on Distributed Software Development*, pp. 101–110, 2008.
 - [40] S. Setamanit, W. Wakeland, and D. Raffo, "Planning and improving global software development process using simulation," in *Proceedings of the International Workshop on Global Software Development for the practitioner*, pp. 8–14, 2006.
 - [41] S.-O. Setamanit, W. Wakeland, and D. Raffo, "Using simulation to evaluate global software development task allocation strategies," *Software Process: Improvement and Practice*, vol. 12, no. 5, pp. 491–503, 2007.
 - [42] S.-O. Setamanit, W. Wakeland, and D. Raffo, "Improving global software development project performance using simulation," in *Proceedings of the Portland International Center for Management of Engineering and Technology - Management of Converging Technologies (PICMET '07)*, pp. 2458–2466, 2007.
 - [43] R. Prikladnicki and S. Marczak, "MuNDDoS: a research group on global software development," in *Proceedings of the IEEE International Conference on Global Software Engineering (ICGSE '06)*, pp. 251–252, 2006.

- [44] R. Prikladnicki, *MuNDDoS: um modelo de referência para desenvolvimento distribuído de software*, Master Thesis, PUCRS, 2003.
- [45] P. Jalote and G. Jain, "Assigning tasks in a 24-hour software development model," *Journal of Systems and Software*, 2006.
- [46] N. Mullick, M. Bass, Z. El Houda et al., "Siemens global studio project: experiences adopting an integrated GSD infrastructure," in *Proceedings of the IEEE International Conference on Global Software Engineering, ICGSE '06*, pp. 203–212, Brazil, October 2006.
- [47] D. K. M. Mak and P. B. Kruchten, "Task coordination in an agile distributed software development environment," in *Proceedings of the Canadian Conference on Electrical and Computer Engineering, CCECE '06*, pp. 606–611, Canada, 2006.
- [48] A. Barcus and G. Montibeller, "Supporting the allocation of software development work in distributed teams with multi-criteria decision analysis," *Omega*, vol. 36, no. 3, pp. 464–475, 2008.
- [49] V. S. dos Santos, T. A. B. Pereira, B. L. Ribeiro, and G. Elias, "Um framework de recomendação para alocação de equipes de desenvolvimento em projetos distribuídos de linhas de produto de software," in *Proceedings of the IV Brazilian Workshop on Distributed Software Development*, 2010.
- [50] M. Ruano-Mayoral, R. Colomo-Palacios, J. M. Fernández-González, and Á. García-Crespo, "Towards a framework for work package allocation for GSD," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 7046, pp. 200–207, 2011.
- [51] S. U. Khan, M. Niazi, and R. Ahmad, "Factors influencing clients in the selection of offshore software outsourcing vendors: an exploratory study using a systematic literature review," *The Journal of Systems and Software*, vol. 84, no. 4, pp. 686–699, 2011.
- [52] S. U. Khan, M. Niazi, and R. Ahmad, "Critical success factors for offshore software development outsourcing vendors: a systematic literature review," in *Proceedings of the Fourth IEEE International Conference on Global Software Engineering (ICGSE '09)*, pp. 207–216, Limerick, Ireland, July 2009.
- [53] S. U. Khan, M. Niazi, and R. Ahmad, "Barriers in the selection of offshore software development outsourcing vendors: an exploratory study using a systematic literature review," *Information and Software Technology*, vol. 53, no. 7, pp. 693–706, 2011.
- [54] P. J. Ågerfalk, B. Fitzgerald, and H. Holmström, "A framework for considering opportunities and threats in distributed software development," in *Proceedings of the In Proceedings of the International Workshop on Distributed Software Development*, pp. 47–61, Austrian Computer Society, Paris, France, 2005.
- [55] J. Noll, S. Beecham, and I. Richardson, "Global software development and collaboration: barriers and solutions," *ACM Inroads*, vol. 1, no. 3, pp. 66–78, 2010.
- [56] M. Jiménez, M. Piattini, and A. Vizcaíno, "Challenges and improvements in distributed software development: a systematic review," *Advances in Software Engineering*, vol. 2009, Article ID 710971, 14 pages, 2009.
- [57] R. Kommeren and P. Parviainen, "Philips experiences in global distributed software development," *Empirical Software Engineering*, vol. 12, no. 6, pp. 647–660, 2007.
- [58] A. Lamersdorf and J. Münch, "A multi-criteria distribution model for global software development projects," *Journal of the Brazilian Computer Society*, vol. 16, no. 2, pp. 97–115, 2010.
- [59] A. Lamersdorf, J. Munch, and D. Rombach, "A survey on the state of the practice in distributed software development: criteria for task allocation," in *Proceedings of the Fourth IEEE International Conference on Global Software Engineering (ICGSE '09)*, pp. 41–50, Limerick, Ireland, 2009.
- [60] I. Tamanini, P. R. Pinheiro, T. C. S. Machado, and A. B. Albuquerque, "Hybrid approaches of verbal decision analysis in the selection of project management approaches," in *Proceedings of the 3rd International Conference on Information Technology and Quantitative Management, ITQM '15*, pp. 1183–1192, Brazil, 2015.
- [61] M. Simão Filho, P. R. Pinheiro, and A. B. Albuquerque, "Task allocation in distributed software development aided by verbal decision analysis," *Advances in Intelligent Systems and Computing*, vol. 465, pp. 127–137, 2016.
- [62] M. Simão Filho, P. R. Pinheiro, and A. Bessa Albuquerque, "Applying verbal decision analysis in distributed software development: rank ordering the influencing factors in task allocation," in *Proceedings of the 11th Iberian Conference on Information Systems and Technologies, CISTI '16*, pp. 1–6, Spain, 2016.
- [63] M. Simão Filho, P. R. Pinheiro, and A. Albuquerque, "Applying verbal decision analysis to task allocation in distributed development of software," in *Proceedings of the 28th International Conference on Software Engineering & Knowledge Engineering (SEKE '16)*, pp. 402–407, 2016.
- [64] M. Simão Filho, P. R. Pinheiro, and A. B. Albuquerque, "Task assignment to distributed teams based on a qualitative multi-criteria approach," in *Proceedings of the 12th Iberian Conference on Information Systems and Technologies, CISTI '17*, pp. 1–6, 2017.
- [65] M. Simão Filho, P. R. Pinheiro, and A. B. Albuquerque, "Analysis of task allocation in distributed software development through a hybrid methodology of verbal decision analysis," *Journal of Software: Evolution and Process*, vol. 29, no. 7, p. e1867, 2017.
- [66] M. Simão Filho, P. R. Pinheiro, and A. B. Albuquerque, "Task assignment to distributed teams aided by a hybrid methodology of verbal decision analysis," *IET Software*, vol. 11, no. 5, pp. 245–255, 2017.
- [67] M. Simão Filho, P. R. Pinheiro, and A. B. Albuquerque, "Task allocation approaches in distributed agile software development: a quasi-systematic review," in *Proceedings of the Brazilian Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (WDES '16)*, vol. 349 of *Advances in Intelligent Systems and Computing*, pp. 243–252, Springer International Publishing, Maringá, Brazil.
- [68] V. R. Basili, "The Role of Controlled Experiments in Software Engineering Research," *Empirical Software Engineering*, pp. 33–37, 2007.
- [69] M. Simão Filho, *An approach structured on verbal decision analysis to support the allocation of tasks in projects of distributed software development*, Doctoral Thesis, University of Fortaleza, 2017.
- [70] S. T. March and V. C. Storey, "Design science in the information systems discipline: An introduction to the special issue on design science research," *MIS Quarterly: Management Information Systems*, vol. 32, no. 4, pp. 725–730, 2008.
- [71] L. H. P. Almeida, *McDSDS: a multi-criteria model for planning distributed software development projects with scrum*, Master Thesis, Graduate Program in Applied Informatics, University of Fortaleza, 2011.

