Usability Evaluation Framework for Domain-Specific Language: A Focus Group Study

Ildevana Poltronieri
Pontifical Catholic University of Rio Grande do Sul - PUCRS
Av. Ipiranga, 6681
Porto Alegre, Brazil
ildevana@gmail.com

Avelino Francisco Zorzo
Pontifical Catholic University of Rio Grande do Sul - PUCRS
Av. Ipiranga, 6681
Porto Alegre, Brazil
avelino.zorzo@pucrs.br

Maicon Bernardino
Federal University of Pampa - UNIPAMPA
Av. Tiaraju, 810
Alegrete, Brazil
bernardino@acm.org

Marcia de Borba Campos
Pontifical Catholic University of Rio Grande do Sul - PUCRS
Av. Ipiranga, 6681
Porto Alegre, Brazil
marciabcampos@hotmail.com

ABSTRACT

Software engineers are increasingly taking advantage of new methods to improve software quality. The use of languages developed for specific domains, which in the literature are known as Domain-Specific Languages (DSLs), has grown in the past years. Although several experimental studies that subjectively evaluate usability of these languages can be found in the literature, few of them have taken advantage of applying Human-Computer Interaction (HCI) techniques in those evaluations. Therefore, the main goals of this paper are to present a usability evaluation framework for DSLs, called Usa-DSL, and to show the evaluation of the framework through a Focus Group method. The evaluation was performed by seven specialists that discussed the framework usability and suggested some modifications of our initial proposal. The specialists recommendations were incorporated in the final framework presented in this paper.

CCS Concepts

•**Software and its engineering** → Domain-specific languages; •**Human-centered computing** → Usability testing; •**General and reference** → *Focus group; Empirical studies;*

Keywords

Usability evaluation, Usability testing,Domain-specific languages, Focus Group

1. INTRODUCTION

Usually, General-Purpose Languages (GPL), such as Java, C#, Ruby, Python, among others, are used for software development. On one hand, this variety of programming languages, allied with the complexity of several applications, may present several difficulties regarding system modeling, implementation, evaluation and maintenance. This may cause different problems and also compromise the quality of the developed systems. On the other hand, there are domain-specific applications that may benefit from languages with specific characteristics, which contribute to the increment of performance, representation, business domain abstraction, better communication between developers and business analysts, among others aspects. Hence, through the development of different languages, software engineers try to facilitate the knowledge sharing of certain domains.

Languages used to describe characteristics of certain domains are called Domain-Specific Languages (DSLs) [14]. Currently, DSLs have been applied to several different domains. For example, there are DSLs applied to software architectures anomalies [1] or performance testing [7] [8]. The difference among these DSLs are defined by theirs syntax and semantic, which are determined by the problem domain. It is important to mention that several different DSLs can be used to represent a domain in order to model its characteristics, without necessarily overlapping them.

Despite all the benefits of DSLs, there is still some effort needed to develop DSLs. Therefore, it is important that these languages meet several usability and satisfaction criteria related to the user experience [32]. Meeting these criteria will enable users to use these languages in a more independent and easier way. This is even more important if someone considers the existing diversity of domains and contexts in which DSLs can be applied to. Furthermore, users (*i.e.* software engineers) satisfaction is an important criteria that has to be taken into account when developing a DSL [24].

Therefore, considering different DSL concepts, or even different domains in which DSLs are applied to, this paper presents a framework to evaluate DSLs usability: the Usability Evaluation of Domain-Specific Languages (Usa-DSL) framework. This framework takes into consideration Human-Computer Interaction (HCI) aspects and apply them to the

evaluation of DSLs usability. This paper also presents the framework evaluation based on a Focus Group [18, 20].

This paper is organized as follows. Section 2 presents some background on the subjects related to this work. Section 3 discusses the related work. Section 4 presents the Usa-DSL framework, as well as motivations for building this framework. Section 5 introduces the Focus Group, describing the pilot instrument as well as the planning, preparation, moderation and data analysis phases of the empirical evaluation. Finally, Section 6 presents the conclusion and future work.

2. BACKGROUND

Software Engineering (SE) [33] is the area that comprises all stages of software development, adopting systematic and organized approaches to produce high quality software efficiently. The adopted approaches involve the analysis of practical issues such as cost, time, reliability and the customers needs. In SE there are different methods and techniques that can be applied depending on the type of application being developed.

One of the processes available in the literature is the Domain Engineering used in the development of reusable applications, which is intended for modeling and identification of the characteristics of an application domain [23] [36]. Its main goal is to allow same domain systems to be built from common processes and artifacts, reusing concepts and resources [26]. The Domain Engineering process allows, from the execution of its activities, the results to be obtained as design patterns, application generators, reusable components, reference architectures or Domain-Specific Languages (DSL), a.k.a. Domain Specific Modeling (DSM) [17].

The DSM is an SE methodology, which has a domain analysis phase that defines the rules, resources, concepts and properties of the domain that must be identified. Therefore, DSM as well as transformation engines and generators are technologies that combined provide support to develop Model-Driven Engineering (MDE) [29]. MDE enables systems to be developed and tested based on domain modeling. It allows to generate code or to analyze its model automatically [10].

In order to minimize the difficulties in applications development, one of the possibilities is the use of Model-Driven Development (MDD) [31]. MDD allows the automatic code generation based on systems models and is based on DSLs. A DSL is a type of programming language or specification used in software development and Domain Engineering [14]. This type of language has as purpose the construction and notation adapted to a given domain [19], *i.e.* the fundamental difference between a DSL and a General Purpose Language (GPL) is that DSL is constructed from the domain of the problem and not the domain of the solution.

DSL as other programming languages has syntax, which defines its structure, and semantics, which defines its meaning. One of the important characteristics in a DSL is the form of the adopted representation, since it must be in accordance with the concepts of the domain that is being modeled. Thus, the principle of representational fidelity [17] [35], which states that only one form represents each concept of the domain, not only simplifies the definition of the notation, but also ensures that all concepts will be represented in this language. The semantics of a DSL defines the meaning of each language construct, and each element has a meaning determined by that domain.

There has been a lot of effort to create and touse DSLs as a resource to facilitate system construction, to increase productivity and to ease its maintenance. However, since a DSL deals with the problem domain, and not with the solution domain, users (software engineers, programmers, ...) not always accept them [15]. This might be mitigated if usability criteria were used during the development of a DSL.

Usability intends to ensure that interactive systems are easy to use, easy to learn, efficacious and pleasant to use, from the users perspective [25]. Usability is not only related to user interface appearance, but refers, mainly, to how the systems interact with users. Therefore, to adopt usability criteria brings several advantages to a user in terms of productivity, better job quality, and satisfaction. To a system producer it brings reduction in terms of maintenance, meets the user expectation, reduces final training costs, and increases competitiveness. Hence, usability should be considered in the whole development cycle of a DSL, improving during DSL lifetime, in order to meet users expectations.

Usability studies started at the beginning of 1980's and is not yet systematized by the developers of interactive systems, including developers of DSLs. There is a lack of usability evaluation, or lack of reporting, during the development of DSLs [27]. Some authors mention that industry does no invest in DSL evaluation since there is no experimental evidences that show qualitative improvement in the development of DSL when a systematic DSL usability evaluation is produced [15]. Furthermore, they also consider that, since a DSL is produced interacting with domain specialists, it usually meets the usability criteria. However, the domain specialists might not be the end users, and therefore, they might be biased by their own experiences, compromising the final DSL usability.

To integrate methods, activities and artifacts from Usability Engineering in organizations that already use a process to develop systems is not simple. Nebe *et al.* [21] discuss the similarities and differences from Software Engineering and Usability Engineering, identifying patterns and processes from these two areas, and point out some fragility from Software Engineering modeling processes in relation to Usability Engineering. Nonetheless, there has been an increase in the application of usability techniques in software development, even though this is not an easy task [13]. One of the reasons might be that software engineers and usability engineers have different views on how to develop a software system, and, therefore, some conflicts might arise due to procedures or terminology differences.

Recent studies on usability evaluation show that usability has been considered during the development of DSLs [27]. Section 3 shows some of these studies. The lack of integration on Software Engineering and Usability Engineering have also been reported by those studies.

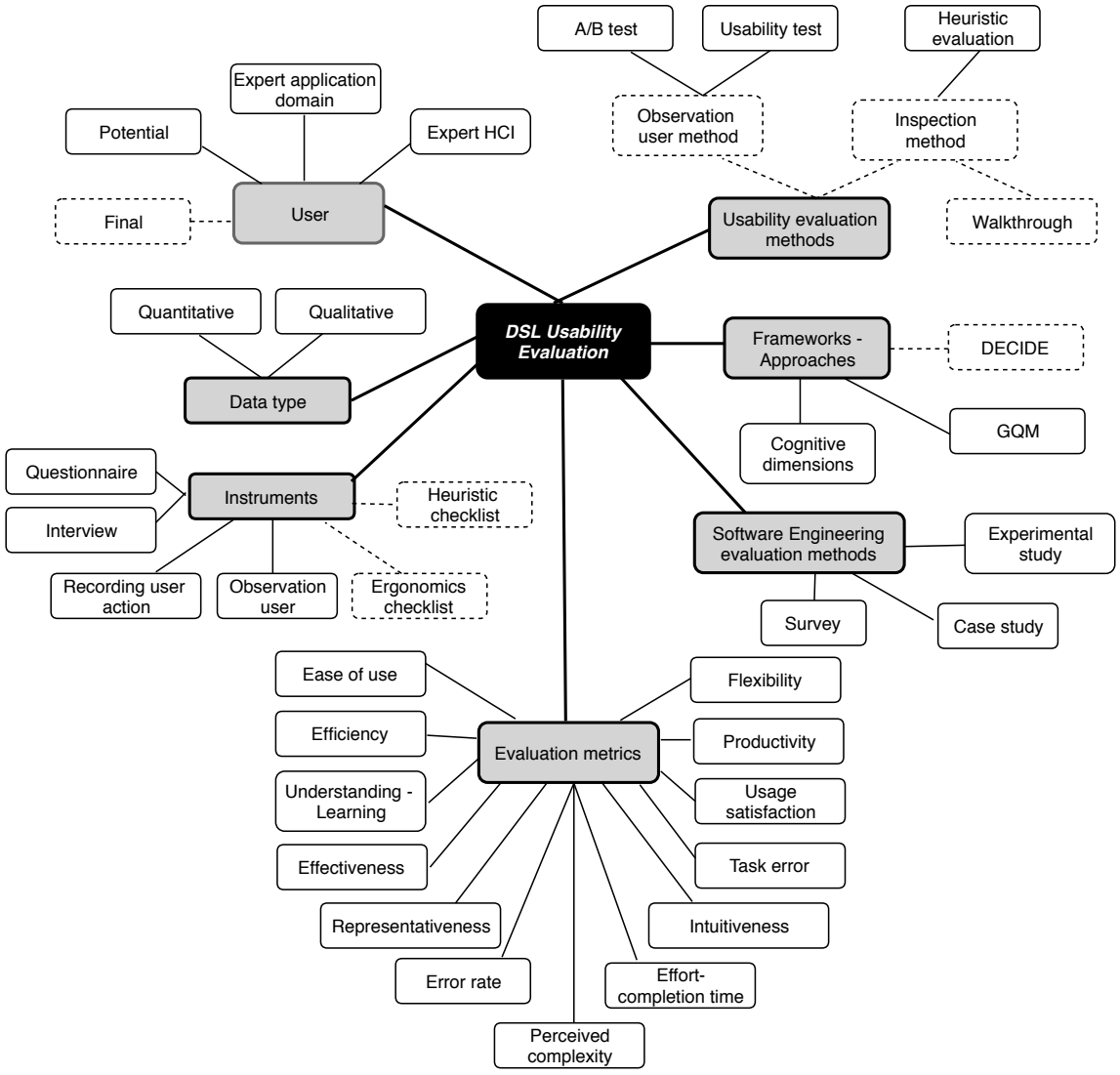

Figure 1: DSL Usability Evaluation Taxonomy [27]

3. RELATED WORK

In a previously developed study [27], a Systematic Review was performed and presents a taxonomy for DSL usability evaluation (see Figure 1). The main research described in that paper, and that are related to this work, are presented in Table 1. The next paragraphs will summarize the contributions of the papers presented in Table 1.

Albuquerque *et al.* [1] presented an evaluation method called Cognitive Dimensions Notation (CDN) that contains 14 dimensions. Such dimensions served to support the development of the characteristics of their work, *i.e.*, DSL expressiveness, which refers to in what extend the DSL represents the domain, and DSL conciseness, which refers to what terms can be deleted without compromising the domain artifact representativeness. These characteristics were also divided into metrics such as: expressiveness, which is composed of hidden dependencies, abstractions, mapping proximity; and, conciseness, which is composed of viscosity, visibility, diffusion and hard mental operations.

Barisic *et al.* [4] suggested that for usability evaluation it is important to first define the usability requirements. Each requirement is assessed by a set of quantitative metrics using Goal Question Metric paradigm (GQM). Regarding cognitive aspects, Barisic *et al.* [6] performed a controlled experiment with six participants to evaluate a cognitive model to languages based on user scenarios. The cognitive activities, in which in the language are: syntax and semantic learning, syntax composition needed to fulfill a role, syntax understanding, syntax debugging, and changing a function that was written by any developer.

Although Ewais and Troyer [12] did not explicitly describe an evaluation method, they used a strategy to evaluate the usability of a language before it would be implemented. To perform this evaluation, fourteen subjects participated in an experiment to evaluate the use of visual domain specific modelling languages for designing.

Table 1: Related Work Summary

Study	Criteria Analysis	Usability Evaluation Method	Software Engineering Evaluation Method
Albuquerque *et al.* [1]	Evaluation method called Cognitive Dimensions (CD) that contains 14 dimensions	Usability Test	Experiment Study
Barisic *et al.* [4][6]	A cognitive model to languages based on user scenarios	Usability Test	Case Study
Ewais and Troyer [12]	A strategy to evaluate the usability of a language before it would be implemented	Usability Test	Experiment Study
Barisic *et al.* [3]	A recommendation-based methodology that considers user-centered techniques	Usability Test	Not Informed
Sinha *et al.* [32]	Evaluation on four heuristics proposed by Nielsen	Heuristics Evaluation	Experiment Study
Seffah *et al.* [30]	User-centered and use case-driven requirements approaches for usability engineering	Usability Test	Not Informed
Alonso-Rios *et al.* [2]	A strategy to evaluate the usability of a language before it would be implemented	Usability Test	Experiment Study
Usa-DSL framework	An usability evaluation framework for domain-specific languages	Usability Test and Heuristic Evaluation	Focus Group Study

Different from other studies, Barisic *et al.* [5] presented the analysis of four controlled experiments. The authors mentioned that the usability evaluation performed in each experiment was based on users interviews, open questionnaires, testing using tools support and multiple-choice questionnaires. Barisic *et al.* [3] used a recommendation-based methodology that considers user-centered techniques. The main activities that their methodologies describe are: domain analysis, language design, controlled experiment as testing, deployment and maintaining.

Sinha *et al.* [32] based their evaluation on four heuristics proposed by Nielsen, and for each heuristic there was a set of metrics. On one hand, learnability was measured through the number of errors a subject made, divided by effort; while efficiency was measured by the size of the test set divided by effort. On the other hand, satisfaction was measured in four levels: frustrating, unpleasant, pleasant, and pleasurable. Therefore, it was possible to have a quantitative evaluation of a DSL when analyzing its usability.

From other point of view, Seffah *et al.* [30] mentions the obstacles that occur to the stakeholders roles on the development process, arguing that the terms "friendly user interface" and "user interface" are obstacles to interactive and usable systems. The author points out that the behaviour of both communities illustrates the separation, isolating the user interface from the rest of the system.

Another important studies is presented by Alonso-Rios *et al.* [2] that describes a usability taxonomy. This proposed taxonomy helped to support the development of the Usa-DSL framework, once many attributes shown in the authors' taxonomy, from the perspective of system usability, were dealt with in our proposal.

Although several researchers have presented some ideas on how to evaluate DSLs, all of them evaluate DSLs in an *ad hoc* manner. The first ideas related to developing a new framework to evaluate DSLs usability were presented in a previous work [27]. Regarding the techniques and methods, some studies present the adaptation or use of a set of usability metrics. Despite the efforts in previous research, there is still a lot of work to transform the conception of DSLs into an easier and more comprehensible and expressive task in relation to the domain that they intend to represent. In addition, it is also necessary to develop processes, methods and techniques that assist in the usability assessment of DSLs.

Section 4 presents the description of the Usa-DSL framework. The current framework includes suggestions from a Focus Group that was applied to evaluate the framework (see Section 5).

4. USA-DSL FRAMEWORK

In order to understand how, usually, DSL designers evaluate DSL usability, a Systematic Review [27] was performed to analyze studies that apply HCI concepts [22, 25, 34] in their evaluation (see Section 3). As mentioned before, different studies presented some discussion on how to use usability concepts to evaluate a DSL, however, to the best of our knowledge, no framework or method to perform usability evaluation of DSLs had yet been proposed. Therefore, this section presents a framework to evaluate DSL usability, called Usability Evaluation for Domain Specific Language framework (Usa-DSL).

The next subsections present the Usa-DSL structure and the details about its phases, steps and activities.

4.1 Usa-DSL Structure

The Usa-DSL framework structure is based on the project life cycle process [34], which is composed of phases, steps and activities (see Figures 2 and 3). Basically, Usa-DSL is organized in phases, in which a set of steps has to be taken. For each step in a phase, there is one or none activity that has to be executed. Notice that some steps, in certain phases, have no activities, *e.g.* step "2 - Ethical and Legal Responsibilities" in phase Analysis has no activity, while this same step in phase Execution has activity "E2 - Introduce the Form and Collect Signatures of Subjects".

There are four phases in the Usa-DSL framework: **P**lanning, **E**xecution, **A**nalysis and **R**eporting (PEAR phases).

Each phase can be split into a set of the following steps: 1 - Evaluators Profiles; 2 - Ethical and Legal Responsibilities; 3 - Data Type; 4 - Empirical Study Method (SE); 5 - Evaluation Method (HCI); 6 - Metrics; 7 - Gathering Instruments; 8 - Evaluation Instructions; 9 - Evaluation Conduction; 10 - Data Packaging and; 11 - Evaluation Reporting.

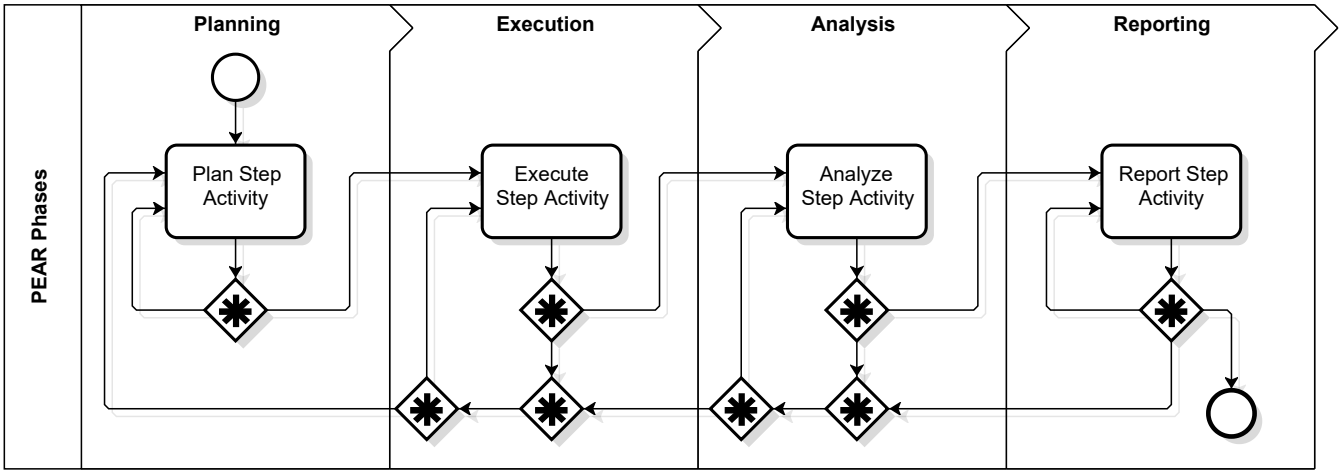

Figure 2: PEAR phases steps/activities order (BPMN notation)

Important to notice that the PEAR phases have to be executed, for each step, in that order. Finally, there are 32 activities that are distributed between phases and steps.

The Usa-DSL framework structure was planned in order to be adapted to the needs of each evaluation. It is possible to begin the "Planning" phase from any of the steps present in the Usa-DSL framework. For example, the evaluator can start the evaluation planning by the "P1 Define Evaluators Profiles" activity, or by the "P3 Define Data Type" activity. This will improve the framework flexibility, since it allows different evaluator to start the evaluation based on the activities that they feel more comfortable with, the ones that they already have some data, or even the activities that are easier to perform for a specific DSL. Besides, if the evaluator wants to perform a step in each of the PEAR phases, that also is possible, for example, it is possible to execute all activities from step "1 - Evaluators Profile" in all PEAR phases before starting activities in any other step. Furthermore, not all steps have to be performed. Some of them might not be executed, for example, the " 4 - Empirical Study Method (SE)" step is only needed if the end user will be involved.

Figure 2 shows a high-level diagram of the order in which steps/activities in the PEAR phases can be executed.

4.2 Usa-DSL Phases

As mentioned before, the Usa-DSL framework contains the PEAR phases (see Figure 3). Each phase has a set of activities that is related to a respective step.

*Phase 1 - **P**lanning*: in this phase, the evaluator organizes the planning of the aspects that will be used in order to evaluate the DSL. In this phase, documents must be defined and created, as well as decision-making about the data that has to be collected or what kind of user will be part of the evaluation, for example. To summarize, this phase is where the structure and planning of the evaluation will be constructed.

*Phase 2 - **E**xecution*: in this phase, the documents created are used, subjects are recruited, environments are created and the evaluation is performed, following the already defined protocol.

*Phase 3 - **A**nalysis*: this phase aims to accomplish the analysis of the artifacts created on the Planning and Execution phases. On the Planning phase, this analysis is executed in order for the documents to be adapted and, therefore, the decisions about the evaluation execution can be made. In this phase, the analysis is focused on the collected data and tasks created.

*Phase 4 - **R**eporting*: in this phase, the evaluator registers the used protocol, the created artifacts and analyzed data.

4.3 Usa-DSL Steps

The Usa-DSL framework is composed of eleven (11) steps. The steps of the Usa-DSL framework are described next (see Figure 3).

Step 1 - Evaluators Profiles: in this step the evaluator profile is defined, instruments to identify the evaluator are applied, the evaluator profile is analyzed and a report on that is written [1, 4, 11, 12, 16].

Step 2 - Ethical and Legal Responsibilities: similarly to the DECIDE Framework, which is an evaluation guide [25], Usa-DSL follows the best practices of ethical and legal issues to protect the user data, dignity, anonymity and well-being. Furthermore, it has to include some description to inform the users that they can stop the evaluation at any time they are not comfortable with some aspects of the evaluation process. At the end of this step, all the signed documents from the subjects are organized.

Step 3 - Data Type: in this step the type of data that will be used is defined, *i.e* the evaluator defines whether the collected data is quantitative, qualitative or both. This will depend on the method that will be used, for example, usability testing uses quantitative, while user observation can use qualitative data. Basically, this step contains only one activity that is performed during the Planning phase.

Step 4 - Empirical Study Method (SE): the Empirical Study Method suggested for Usa-DSL is based on the Wohlin *et*

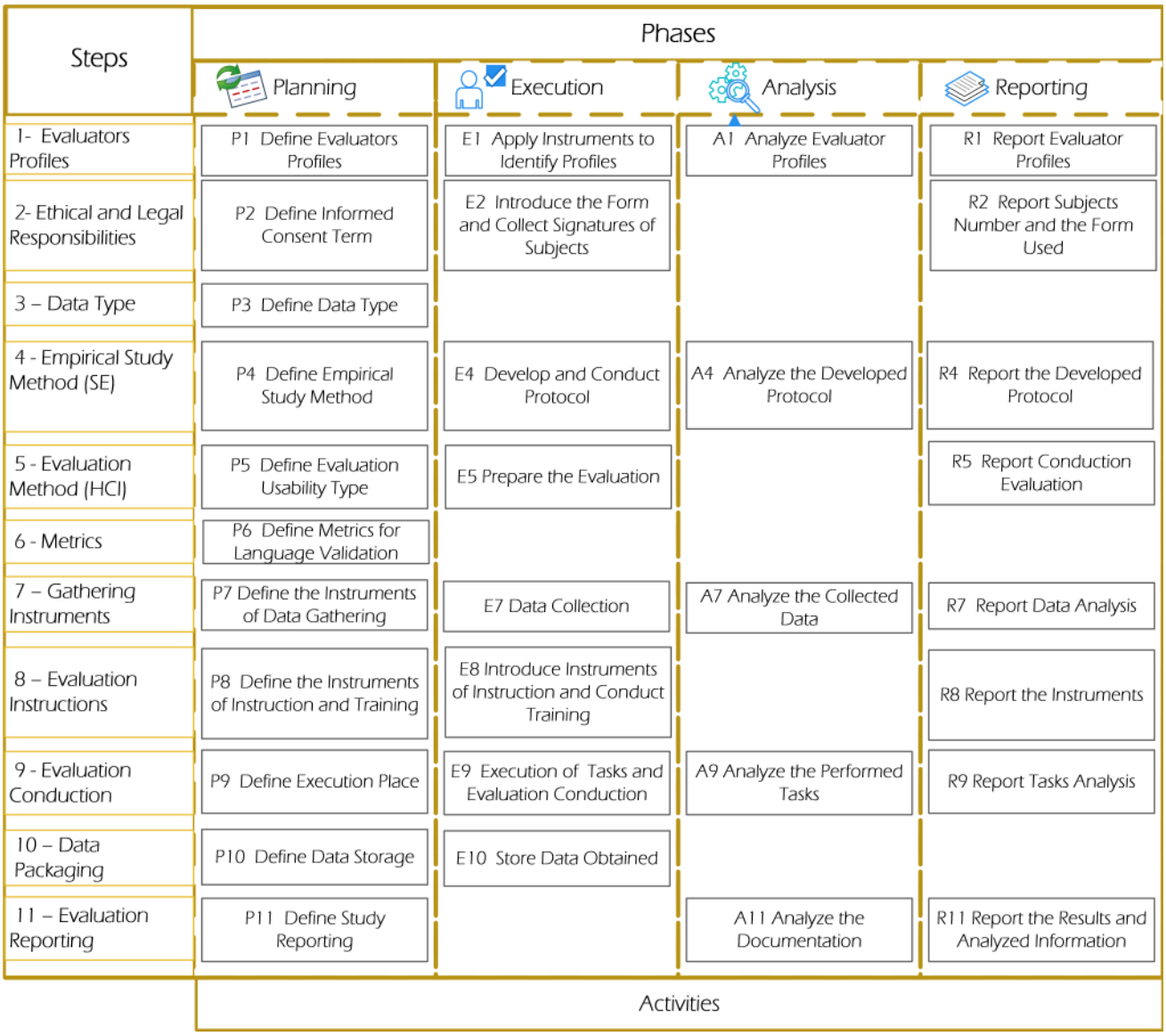

Figure 3: Usa-DSL Framework Structure

al. [37] proposal, which can be a survey, a case study or a controlled experiment. These methods can be defined based on, for example, the evaluator's profile (Step 1) or the data that will be collected (Step 3). The Empirical Study Method can be used with other evaluation methods, *e.g.* usability testing or heuristic evaluation. However, the restrictions and characteristics of every method must be always respected.

Step 5 - Evaluation Method (HCI): the evaluation methods defined on Usa-DSL can be, for example, user observation evaluation, usability testing, inspection evaluation, or heuristic evaluation. The user observation evaluation must be applied when the study intention is to obtain the end users opinion about the DSL usability aspects. The inspection evaluation aims to verify the relevance of the language on the usability specialist level.

Step 6 - Metrics: the metrics used on Usa-DSL were defined from an SLR mapping [27]. They are comprehension/learning, ease of use, effort/conclusion time, observed complexity and efficiency. These metrics will guide the definition of the evaluation instruments questions to be applied during the evaluation. Similarly to Step 3, this step has only one activity performed during the Planning phase.

Step 7 - Gathering Instruments: the instruments were based on the studies of [25] and [28], *e.g.* heuristic checklist, ergonomic checklist, questionnaires, interview, use observation or user action recording.

Step 8 - Evaluation Instructions: according to Wohlin *et al.* [37], the evaluation instructions can be composed of use manual, instruments or task to be performed. These instruments must be distributed and used when executing an

empirical method. They are used, for example, to clarify the participants of the evaluation on what will be evaluated and when the evaluation will take place.

Step 9 - Evaluation Conduction: this is the step in which the aspects defined on the previous steps are applied. Therefore, it is necessary that the previous steps were executed and tested thoroughly, before involving the evaluation participants. Hence, a pilot test must be executed prior to the application of the evaluation to the actual participants. This will guarantee that the evaluation is viable. Furthermore, it is also important to guarantee that the needed number of participants will be achieved, otherwise, the results may not be statistically relevant for a quantitative evaluation.

Step 10 - Data Packaging: when the evaluation is finalized, the material used for training and the collected data should be stored in order to allow the study replication when necessary. This will allow future language evaluation and its comparison with the new collected data.

Step 11 - Evaluation Reporting: this report must follow the evaluation method that was chosen in step "5 - Evaluation Method (HCI)". Each evaluation method provides a specific report with different fields that must be filled.

4.4 Usa-DSL Activities

The Usa-DSL framework activities are composed by a set of actions used to plan, execute, analyze and report the evaluation. The full set of activities can be seen in Figure 3. The description of the activities from all phases can also be found at `https://github.com/Ildevana/Usa-DSL/wiki/Usa-DSL-Structure`.

It is worth mentioning that the identification of each of the 32 activities is composed of an ID and its name. ID is composed of a letter and a number. The letter represents a phase and the number a step, *e.g.* "E5 Prepare the Evaluation" is an activity that belongs to phase **E**xecution and is associated with the "**5** - Evaluation Method (HCI)" step.

4.4.1 Planning Phase Activities

The **P**lanning phase contains eleven (11) activities. These activities define the whole evaluation protocol. They are:

P1 - Define Evaluators Profiles: the goal of this activity is to define the evaluators profiles, which will be related to the evaluation method that will be used. The evaluation can be performed by, for example, an HCI expert, a domain analyst, a domain developer or a domain tester.

P2 - Define Informed Consent Term: it is a formal document that describes the evaluation goal, how the evaluation will take place, how the data will be collected, how the data will be protected, and so on. Usually, it is recommended the use of ethical codes from organizations like, for example, The Association for Computing Machinery (ACM)[1].

P3 - Define Data Type: the collected data type from the evaluation can be quantitative and/or qualitative. The quantitative data are numeric results that predict the quantity of answers attributed to determined item of a question. The qualitative data is composed of subjective information related to the participant's opinion about the studied object. These data aim to predict what kind of information the evaluator intends to obtain. Albuquerque *et al.* [1] suggest the use of two data types, in order to obtain a wider and more complete view about the participant opinions. Barisic *et al.* [4], on the other hand, use quantitative data and consider that to be sufficient for the goal of their research.

P4 - Define Empirical Study Method: there are different empirical evaluation methods that can be used to evaluate usability. These methods have to involve users during data collection. This activity is closely related to activity P2. Examples of empirical methods are: Controlled Experiment, Survey or Case Study.

P5 - Define Evaluation Usability Type: as mentioned in the description of step "5 - Evaluation Method (HCI)", evaluation can be through end users, HCI or DSL experts. This activity is related to activities P1, P3 and P4.

P6 - Define Metrics for Language Validation: the metrics depend on the evaluation goal and usability criteria that someone wants to evaluate. Examples of criteria that may be evaluated are: easy to learn, easy to remember, easy to use, effort/conclusion time, perceived complexity, utility, satisfaction, conclusion rate, task error rate, efficiency or effectiveness.

P7 - Define the Instruments of Data Gathering: some of the instruments that can be used to collect data can be heuristic checklist, log capture, use observation, interview or questionnaire.

P8 - Define the Instruments of Instruction and Training: the Usa-DSL framework use the following instruments: DSL guide, user scenario and language presentation. This activity also defines the tasks that will be executed by the user, when an empirical method is chosen. In that case, this activity has a close relation to P3, P4 and P5.

P9 - Define Execution Place: the place where the evaluation will take place depends on the data type that will be collected, the empirical study method that was chosen or even the usability type. For example, places could include a laboratory, via e-mail or through web, or even the users work place.

P10 - Define Data Storage: data packaging is a important activity, since this data might be used later in to replicate the evaluation.

P11 - Define Study Reporting: this activity is responsible for describing the way the results of the evaluation will be registered.

4.4.2 Execution Phase Activities

The **E**xecution phase is composed by eight (8) activities. Each of these activities is used after the Planning phase step. They are:

E1 - Apply Instruments to Identify Profiles: questionnaire that characterizes the profile of the evaluation of participants is applied. This document is used to obtain information such as: DSL/Domain experience time, training, and area of activity.

[1] https://www.acm.org/code-of-ethics

E2 - Introduce the Form and Collect Signatures of Subjects: in this activity, the consent form must be presented to the participants and after their reading and consent, it must be signed and a copy is given to the researcher that is conducting the evaluation. The consent form provides the subjects with sufficient written information to decide whether to participate in a research study or not.

E4 - Develop and Conduct Protocol: this activity consists of developing the evaluation protocol, describing all the steps and documentation that will be used, such as the type of evaluation, experimental study, context, hypotheses, variables of the study, profile of the participants, the instruments, type of data, data storage and how the study will be reported. This protocol must be performed by the researcher carefully following the planned steps and activities.

E5 - Prepare the Evaluation: the evaluation instruments should be organized, the equipment arranged in the rooms, the participants must be available at the scheduled date and time, and the questionnaires answered.

E7 Data Collection: by applying the characterization questionnaires, collecting instruments and obtaining the data recorded in audio or video, they must be compiled and stored for later tabulation, transcription and analysis.

E8 - Introduce Instruments of Instruction and Conduct Training: the presentation and training are intended to guide the functioning of the language, regarding syntax and semantics, as well as to instruct on the usage scenario, that is, explain the task to be performed in the evaluation process. The delivery of the language manual and usage scenario refers to the delivery of the printed or online documents. These documents describe the functioning of the language and its syntax and semantics. They contain the description of the usage scenario that must be expressed as a requirement or task.

E9 - Execution of Tasks and Evaluation Conduction: the task must be modeled according to the usage scenario delivered to the participants and must be performed from the tool that supports the execution of the language. Upon completion of the task modeling, the researcher may conduct an interview with the participants and thereby obtain their opinion about the language being evaluated. In addition to the interview, the researcher can choose only the use of the questionnaire, filled by participants after completing their tasks.

E10 - Store Data Obtained: after performing the evaluation, the collected data should be stored in a database or other location in order to compile the data later. If data are quantitative, it is important to tabulate them so that their behavior can be observed later and thus to obtain conclusions. If the data is qualitative, it is important to process the interviews, annotations, answers to open questions, recordings and access logs, trying to obtain patterns and a set of relevant information for the study.

4.4.3 Analysis Phase Activities

As mentioned before, the **A**nalysis phase contains five (5) activities:

A1 - Analyze Evaluators Profiles: the analysis of the profiles is used to gather the number of participants and the type of knowledge they have. These profiles can be classified as: `Beginner` - one who does not have solid knowledge on the domain or on DSL. `Intermediate` - one who has some knowledge on the domain and/or on DSL `Advanced` - one who has solid knowledge on the domain and on DSL.

A4 - Analyze the Developed Protocol: in the analysis activity of the study protocol, all the described steps should be reviewed in detail and how they will be performed in order to ensure the validity of the study.

A7 - Analyze the Collected Data: when analyzing the data collected during the evaluation, standardization, hypothesis testing, analysis of images and logs, transcription of interviews and videos are performed.

A9 - Analyze the Performed Tasks: the developed models should be checked by more than one researcher to verify and to obtain the task execution rate and the error rate performed by the participants. After, the evaluation of those that did not reach the objectives of the task, or did not complete the intended task, will be discarded.

A11 - Analyze the Documentation: the documentation used in the evaluation must all be analyzed by the researcher and checked by a second researcher to ensure the consistency of the produced information and documentation.

4.4.4 Reporting Phase Activities

The final phase is **R**eporting, which is composed of eight (8) activities that aim to register what was performed during the previous evaluation phases. These activities are:

R1 - Report Evaluator Profiles: when reporting the participants' profile, the classification and the total number of participants who performed the evaluation should be taken into account. Furthermore, other information should be described if it appears in the characterization questionnaire.

R2 - Report Subjects Number and the Form Used: all documents used in the evaluation should be described in detail and attached to the final report.

R4 - Report the Developed Protocol: the study protocol should be described for each planned, executed and analyzed step.

R5 - Report Conduction Evaluation: HCI evaluation methods must be described: `Usability Testing` - this evaluation aims to test whether potential users would use the language developed to model the domain to which it was proposed; `Heuristic evaluation` - this is a usability inspection method, which is applied by HCI specialists who are guided by a set of heuristics developed by Nielsen. This method aims to identify usability problems in the evaluated language.

R7 - Report Data Analysis: `Quantitative data` - should be reported through charts, spreadsheets or hypothesis testing. `Qualitative data` - can be represented by an image, interview transcript, annotation excerpts, categorization and standards, high-level video narratives, and fragments of open-ended questions.

R8 - Report the Instruments: the instruments used, charac-

terization questionnaire, language manual, usage scenario, interview script, opinion questionnaire, among others, must be detailed at high level in the protocol and arranged in an appendix in the document used to present the study.

R9 - Report Tasks Analysis: the evaluation must be reported according to the chosen method. The usability test will be reported through the protocol of an experiment, case study or survey. When a heuristic evaluation is performed, the analysis performed by the specialists, as well as the activities carried out and the generated models, should all be reported.

R11 - Report the Results and Analyzed Information: at the end of the evaluation the data should be fully described in a report format, containing all the documents attached to the report.

5. EVALUATION: FOCUS GROUP

In order to evaluate the Usa-DSL framework several strategies could have been used, for example, a focus group [20] or an empirical controlled experiment [37]. This paper presents the evaluation performed using a Focus Group method, which gathers qualitative data during group discussion sessions. This method was chosen because it is a useful method that can be used to measure the reaction of specialists and, therefore, some straightforward conclusions can be drawn by the group. A focus group is organized in phases [9, 18, 20]: planning, preparation, moderation, and data analysis and reporting (see Figure 4). Usually, a focus group is composed by a moderation team (normally an interviewer/facilitator and moderation assistants) and a set of subjects. For the evaluation subjects, HCI, Software Engineering and Performance Testing experts were invited.

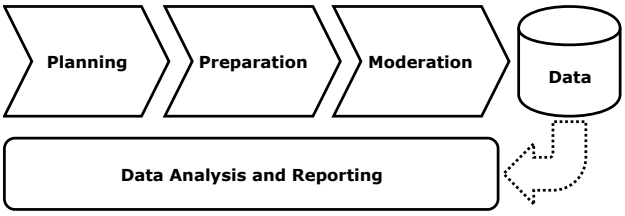

Figure 4: Focus Group Process

Furthermore, to verify whether the focus group phases were ready to be applied to the subjects, a test pilot with two subjects was used. These subjects belonged to the DSL Canopus project [7] [8]: a project analyst and a developer. After the test pilot, some modifications were applied to the framework structure before submitting it to the focus group, for example, timings were altered, the questionnaires glossary was improved, and the number of activities that would be discussed was reduced. These two subjects were later involved in the focus group as assistants, one as recorder and other as timekeeper.

The next sections detail the results from each phase of the focus group.

5.1 Planning and Preparation

The main goal of the focus group was to validate the Usa-DSL framework in order to understand whether the framework phases, steps and activities would effectively prepare a usability protocol to evaluate the usability of a DSL. Hence, the planning and preparation phases had to allow the subjects, during the moderation phase, to understand the framework structure and objectives. Furthermore, the planning and preparation had to be able to produce good discussions among the subjects during the moderation phase. Hence, the discussion session guide, documents to be presented to the subjects, questions that needed answers, and all the environment for the focus group, were planned/ prepared during these phases. All these were previously verified in the test pilot, as mentioned before.

During the planning phase, the goals of the focus group, the profiles of the subjects, the way the discussion would be conducted, the role of the interviewer and the assistants, date and place for the focus group, and which documents would be used, were defined. The goal was already mentioned in the previous paragraph. Date and place were set as 2017, May and a technological park from a federal university[2], respectively. Also, as mentioned before, the subjects had to have experience on using or designing DSLs, or understanding of HCI evaluation. Some of the subjects had knowledge on both HCI and DSLs. In the end, seven subjects were selected to be part of the focus group.

In order to prepare the environment, and to avoid any kind of interruption during the discussion session, the following preparation was executed prior to the evaluation of the Usa-DSL framework: 1) the meeting room was prepared with some audio and video recording equipment; 2) all the printed documents were reviewed and accounted for; 3) audio and recording equipment were tested.

In order to assist the subjects to visualize the framework structure a board with the Usa-DSL framework (see Figure 3) was always available for the subjects. Subjects also used post-it, pens, and had access to the guide, informed consent term and questionnaire.

5.2 Moderation

In the moderation phase, it was ensured that the subjects felt comfortable, respected and free to expose their opinions [18]. To achieve this goal, a script was followed as a guide. This script presents a welcome message to the subjects, the instructions on the "Informed Consent Term", the completion of the profile questionnaire and the printed documents that would be used during the session[3].

In order to achieve this goal, a guide as a script was suggested. This script shows a welcome message to the subjects, the instructions about the Informed Consent Term, the completion of the characterization questionnaire and a document describing all activities that would be performed during the session.

To mitigate understanding problems during the discussion

[2]Technological Park of Pampa (PampaTec) from UNIPAMPA. `http://porteiras.s.unipampa.edu.br/pampatec/`

[3]Documents are available at `http://tiny.cc/SAC-UE-2018`.

session, the context of each research topic was presented and, also, how the session would be organized. The first part of the session would be used to discuss the steps of the framework (see Figure 3). After that, each of the phases of the framework would be discussed. The third part of the session would be used to discuss the activities of the framework. Finally, during the closing of the session, some general discussion on the initial framework proposal and the considerations from the group would be presented. In the last part of the session, the subjects would also present some final considerations.

During the session parts, the mediator would allow free discussions among the subjects. The mediator would only intervene when the discussion would get out of the scope of the goal of the focus group, or when some of the subjects was not participating in the discussion. The subjects were free to discuss any topic (in each part of the session, as explained before), and they would decide what had to be performed regarding each topic, *i.e.* to maintain, to join, to modify, to include, to change syntax, to change semantics, or to remove something. In each part of the session, the group would elect a rapporteur that was responsible to fill the questionnaire at the end of each part of the discussion session.

The duration of the discussion session was two hours and twenty minutes (2h20min), including the presentation time of each topic, its objectives and intentions. The opinions expressed by the subjects were recorded in audio and video and later transcribed. Namely, the audio recording was used to support the video recording, so that it could help in understanding the discussion.

5.3 Data Analysis

The phase of analysis and interpretation of the generated data constitutes an important part of the qualitative research, considering the context, the behavior and the perception of the subjects [18] [20]. For the data analysis phase, the audio from the video was transcribed and the recorded audio was used when the sound of the video was not clear. The transcript followed the order in which the study script was planned, separating the discussion by session and comparing with what was reported in the questionnaire delivered by the rapporteur. The analysis presented in this section were firstly performed by one researcher, and later the conclusions were discussed with a second researcher to validate the results. The subjects experience, expressed in the profile questionnaire and presented in the Table 2, was also taken into account.

The next sections present a summary of the subjects discussions. This summary was based on the transcription of the recordings performed during the discussion session.

5.3.1 First Session Part: Usa-DSL Steps

At the start of this session part, the subject identified by S1 mentioned that he had already read the material that he had received by e-mail. Furthermore, he also said that the framework seems to be for generic enough to be used not only for DSLs. He questioned the reason for the existence of steps "4 - Empirical study method" and "5 - Evaluation method", but as the discussion progressed he understood that one step is related to SE and the other one to HCI. S2 mentioned the "3 - Data Type" step and asks the other subjects why the "3 - Data Type" step should be defined before the " 4 - Empirical Study Method" step. He mentions that if he knew how to use a particular method, it would be easier to define the "3 - Data Type". However, S5 said that the order of the steps "3 - Data Type" and "4 - Empirical Study Method" makes sense and argued with S2 that it would not be enough to be an expert in a method to perform an evaluation. In the first interaction of subject S4, he expressed his idea that steps "3 - Data Type" and "4 - Empirical study Method" should be merged, but was convinced by the explanation of S6 that the data type must be defined before the empirical study method and stated that the data to be collected can change the method to be applied. S3 agreed that "3 - Data Type" must be defined before the "4 - Empirical Study Method"; at this point the interviewer instigated S7 to participate in the discussion, but he did not have anything further to add.

> *"[...] the order of the Data Type and Empirical Study Method makes sense, if I am, for example, an expert in an empirical study method it would not be enough to carry out an evaluation, because*

Table 2: Subjects Profile - Focus Group

S	Profile
S1	Used DSL UML and SQL, but did not participate in a development project. Participated in a heuristic evaluation, but did not conduct usability assessments. Never participated or conducted empirical experimental evaluation.
S2	Used DSL SQL, UML and Relax and participated in the development project of DSL Relax. Participated and conducted a heuristic evaluation. He participated in a survey, but never conducted any empirical experimental evaluation.
S3	Used DSL SQL, UML and HTML, but did not participate in a development project. Participated in a heuristic evaluation, but did not conduct usability assessments. Participated in a controlled experiment and conducted a survey.
S4	Used DSL SQL and HTML, but did not participate in a development project. Participated in a heuristic evaluation, but did not conduct usability assessments. He participated in a case study, a controlled experiment and a survey, but never conducted an empirical experimental evaluation.
S5	Used DSL SQL and HTML, but did not participate in a development project. Participated in a heuristic evaluation, but did not conduct usability assessments. Participated in a controlled experiment, but never conducted an experimental empirical evaluation.
S6	Used DSL Method 2ed, VDM-SL, Vienna, but did not participate in the development project. Participated in the Usability Test and Heuristic Evaluation and also conducted a Usability Test. It did not answer the question of participation in an empirical experimental evaluation but claims to have conducted a case study and experiment.
S7	Made use of DSL SQL, UML and HTML, but did not participate in the development project. Participated and conducted a Heuristic Evaluation. Did not conduct or participate in an empirical experimental evaluation.

if the data we want to obtain is quantitative and he only knows how to do Case Study this would not solve [...]" (S5)

"[...] the data to be collected can change the empirical study method to be applied." (S6)

The second issue raised was regarding step "6 - Metrics". Subjects S2 and S4 repeatedly questioned whether metrics should be selected from the data type or whether they would depend on the empirical study method and whether metrics could be changed or new metrics could be included at the time of the evaluation. After a lot of questioning, S4 concluded that the choice of metrics is based on the data type and the group was persuaded to maintain the order.

The last issue to be discussed was on step "8 - Evaluation Instructions". S7 suggested that it should be placed before step "6 - Metrics", but quickly S5 replied that it was not possible to instruct someone about the evaluation before beginning the evaluation. After that, this issue was considered as resolved by the subjects.

At the end of the topic, when the rapporteur began to respond the questionnaire, the group suggested reading item by item so that, in common agreement, the alternatives, justifications and changes would be described. Basically, the subjects strongly agreed with most questions that were asked in the questionnaire. Although they believed step "7 - Gathering Instruments" could be changed to "7 - Evaluation Tools", in the end, they did not really suggested that change since it was not mandatory. After 15 minutes of discussion, the group decided not to modify the steps of the framework.

"[...] cannot instruct on the system operation before preparing the evaluation." (S5)

This section described how the subjects behaved during the discussion session. The next sections do not present the way they discussed, but describe a summary of the discussion in each session part.

5.3.2 Second Session Part: Usa-DSL Phases

Initially, Usa-DSL was composed of the following phases: "Definition", "Execution", "Analysis" and "Results". However, during the discussion session it was clear that the "Definition" term should be wider, and, therefore, it was changed to "Planning". There were some questioning related to the "Execution" and "Results" terms. The subjects were not convinced when the evaluation data collection, recording and results dissemination activities should be performed. In the end, there was a general understanding that the "Execution" phase should include activities related to data collection. Besides, the "Results" phase name was changed to "Reporting", since it includes the activities "Record of Results" and "Data Collection".

"[...] the term "Results" does not seem to be a good name for a phase [...] results gives a discontinuity perception, and it seems that the evaluation finishes there and that there is nothing else to be done [...]" (S4)

"[...] the term "Results" is not clear for a phase name [...]" (S5)

5.3.3 Third Session Part: Usa-DSL Activities

At the beginning of the third session part, a board with the complete view of the framework was presented to all the subjects. This board contained the framework phases, steps and activities, even though during this part of the session the goal was to discuss only the framework activities (see Figure 3). Furthermore, a document containing each activity description was available. First, the description was read and discussed by the subjects.

First, there was a discussion on the activities names and whether they were included in the right phase. There was not questioning regarding the step in which the activity was included to. After that, the subjects chose an activity randomly to start the discussion. Some subjects questioned the importance to include the place in which the evaluation would take place, and also if this could be included in an activity called "Define and Conduct the Evaluation". In the end, the subjects considered that it would be important to keep the activity as "Define Execution Place".

In order to organize the discussions, the subjects decided to discuss the activities by phase. There were some suggestions to split activities ("Prepare and Conduct the Evaluation"), to join activities ("Execution of Tasks" and "Evaluation Conduction") and to create new activities ("Compile and Protocol Review"). Regarding this suggestions, S3 led the discussion and pointed out that their goal was to evaluate the Usa-DSL framework structure and not to describe activities following some usability evaluation method or software testing technique.

At the end of this part of the session, the subjects read the description of the activities again and decided to make some corrections on duplicated information in the activities. The subjects also considered that the examples mentioned in the "Define Experimental Study Method" activity could induce to someone to choose certain evaluation method. Hence, if that was not the intention, then that should be avoided.

During the discussions about the framework activities, there was a better understanding on the Usa-DSL framework. See some statements from some of the subjects:

"The execution of the framework works as a matrix that crosses steps and phases that results in activities." (S4)

"The framework is a set of good practices." (S6)

5.3.4 Fourth Session Part: Usa-DSL Structure

The last part of the discussion session was used to close the discussion and also to confirm the framework structure that was suggested by the focus group. Even though the duration of the discussion session was long, *i.e.* 2h20min, the subjects remained interested and engaged in the discussions. They confirmed that the structure of the framework was good and only some minor changes should be made, for example, change some terms names. Some minor com-

ments were added at this moment. For example, S4 suggested that a document describing recommendations or usage rules should be added. S1 mentioned that it seemed that an evaluator with no experience in usability evaluation would be able to carry out an evaluation using this framework. Two subjects, S1 and S6 said that it would also be important to have a workflow for an evaluator to make it easier for someone to follow the framework. The subjects also reported that they had plenty of time to discuss all the topics that were supposed to be discussed.

5.3.5 *Usa-DSL Changes after the Focus Group*

Some changes were proposed by the Focus Group and were incorporated in the final Usa-DSL framework (see Section 4). Those proposals were related to the name of the phases or inclusion of new activities. For example, subjects from the Focus Group suggested to use the term "Planning" for the first phase, rather than the term "Definition", which was used before the Focus Group. Another name that the subjects suggested changing was the last phase name, from "Result" to "Results Registering", which in the end was changed to "Reporting". Regarding the inclusion of new activities, the subjects suggested the inclusion of two activities in the Planning phase, *i.e.* "Define the Instruments of Instruction and Training" and "Compile and Review Protocol". While the former was included in the framework, the latter was not since it should be a sub-activity in all other phase activities. The last change suggested by the subjects was to gather together activities "Execution of Tasks" and "Evaluation Conduction", which were two separate activities in the Execution phase. These two activities became activity "E9 - Execution of Tasks and Evaluation Conduction".

It is worth mentioning that another suggestion that was incorporated to the final version of Usa-DSL was to include an identification to each activity, since this would facilitate future discussions and framework organization. Therefore, each activity contains an ID and its name, where the ID is composed by a letter and a number. This was also applied to the steps of the framework.

6. CONCLUSION

Domain engineers aim to, through the development of different languages, facilitate the creation of new concepts and theories in order to minimize the difficulties inherited from applications development. One way of minimizing this difficulties it to use Domain Specific Languages, DSL. Although these languages help the developers, their usability has to be analyzed in a thorough way.

This paper presented a framework that will help DSL developers to evaluate the usability of the languages that they are proposing. This framework was evaluated using a focus group method, which confirmed that the framework will help DSL developers. The subjects that participated in the Focus Group had previous experience developing or using DSLs and they believe their job would have been easier if they had the framework to help to improve the DSLs.

For future research, it would be important: 1) To accomplish the evaluation of several DSLs, preferably using different usability methods and also the Usa-DSL framework; 2) To evaluate DSLs using different evaluator profiles or different evaluation frameworks; 3) To propose a process in order to assist the activities presented in the Usa-DSL framework; 4) To improve artifacts, such as: checklist, manuals, questionnaires and protocols, which will support the evaluation process and the Usa-DSL framework; 5) To evaluate empirically the artifacts developed using the Usa-DSL framework.

7. ACKNOWLEDGEMENT

The authors would like to acknowledge DELL, CNPq and CAPES for their financial support in the development of this work.

8. REFERENCES

[1] D. Albuquerque, B. Cafeo, A. Garcia, S. Barbosa, S. Abrahão, and A. Ribeiro. Quantifying Usability of Domain-Specific Languages: An Empirical Study on Software Maintenance. *Journal of Systems and Software*, 101:245–259, 2015.

[2] D. Alonso-Ríos, A. Vázquez-García, E. Mosqueira-Rey, and V. Moret-Bonillo. Usability: A Critical Analysis and a Taxonomy. *International Journal of Human–Computer Interaction*, 26(1):53–74, 2009.

[3] A. Barisic, V. Amaral, and M. Goulão. Usability Evaluation of Domain-Specific Languages. In *Quality of Information and Communications Technology*, pages 342–347, 2012.

[4] A. Barisic, V. Amaral, M. Goulão, and A. Aguiar. Introducing Usability Concerns Early in the DSL Development Cycle: FlowSL Experience Report. In *International Workshop in Model-Driven Development Processes and Practices*, pages 8–17, 2014.

[5] A. Barisic, V. Amaral, M. Goulão, and B. Barroca. Evaluating the Usability of Domain-Specific Languages. *Formal and practical aspects of domain-specific languages: Recent developments*, pages 2120–2141, 2012.

[6] A. Barivsic, V. Amaral, M. Goulão, and B. Barroca. Quality in Use of Domain-specific Languages: A Case Study. In *3rd ACM SIGPLAN Workshop on Evaluation and Usability of Programming Languages and Tools*, pages 65–72, 2011.

[7] M. Bernardino, E. Rodrigues, and A. F. Zorzo. Performance Testing Modeling: an empirical evaluation of DSL and UML-based approaches. In *31st ACM Symposium on Applied Computing*, pages 1660–1665, 2016.

[8] M. Bernardino, A. F. Zorzo, and E. Rodrigues. Canopus: A Domain-Specific Language for Modeling Performance Testing. In *9th International Conference on Software Testing, Verification and Validation*, pages 157–167, 2016.

[9] M. Bloor, J. Frankland, M. Thomas, and K. Robson. *Focus Groups in Social Research*. Sage Publications Ltd, 2001.

[10] J. W. M. Brambilla, M.; Cabot. *Model-Driven Software Engineering in Practice*. Morgan & Claypool Publishers, 2012.

[11] F. Cuenca, J. V. Bergh, K. Luyten, and K. Coninx. A User Study for Comparing the Programming Efficiency of Modifying Executable Multimodal Interaction Descriptions: A Domain-specific Language Versus Equivalent Event-callback Code. In *6th Workshop on Evaluation and Usability of Programming Languages and Tools*, pages 31–38, 2015.

[12] A. B. Ewais and O. De Troyer. A Usability Evaluation of Graphical Modelling Languages for Authoring Adaptive 3D Virtual Learning Environments. In *6th International Conference on Computer Supported Education*, pages 459–466, 2014.

[13] X. Ferré, N. J. Juzgado, H. Windl, and L. L. Constantine. Usability basics for software developers. *IEEE Software*, 18:22–29, 2001.

[14] M. Fowler. *Domain Specific Languages*. Addison-Wesley, 2010.

[15] P. Gabriel, M. Goulão, and V. Amaral. Do software languages engineers evaluate their languages? In *13th Conferencia Iberoamericana in Software Engineering*, pages 149–162, 2010.

[16] I. Gibbs, S. Dascalu, F. C. Harris, and Jr. A Separation-based UI Architecture with a DSL for Role Specialization. *Journal of Systems and Software*, 101:69 – 85, 2015.

[17] J. Kelly, S.; Tolvanen. *Domain-specific modeling : enabling full code generation*. IEEE Computer Society, Hoboken, N.J. Wiley-Interscience, 2008.

[18] R. A. Krueger and M. A. Casey. Designing and Conducting Focus Group Interviews. *Social analysis, selected tools and techniques*, 4(23):4–24, 2002.

[19] M. Mernik, J. Heering, and A. M. Sloane. When and how to develop domain-specific languages. *ACM Computing Surveys*, 37(4):316–344, Dec. 2005.

[20] D. L. Morgan. Focus Groups. *Annual Review of Sociology*, 22(1):129–152, 1996.

[21] K. Nebe, D. Zimmermann, and V. Paelke. *Integrating Software Engineering and Usability Engineering*. IntechOpen, 2008.

[22] J. Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[23] G. v. d. L. F. J. Pohl, K.; Böckle. *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer-Verlag, 2005.

[24] I. Poltronieri, A. F. Zorzo, M. Bernardino, and M. de Borba Campos. Usa-dsl: Usability evaluation framework for domain-specific languages. In *33rd ACM Symposium on Applied Computing*, pages 2013–2021, 2018.

[25] J. Preece, H. Sharp, and Y. Rogers. *Interaction Design: Beyond Human-Computer Interaction*. Wiley, 2015.

[26] G. Prieto-Diaz, R.; Arango. *Domain Analysis and Software Systems Modeling*. IEEE Computer Society Press, 1991.

[27] I. Rodrigues, M. B. Campos, and A. F. Zorzo. Usability Evaluation of Domain-Specific Languages: a Systematic Literature Review. In *19th International Conference on Human-Computer Interaction*, pages 522–534. Springer, 2017.

[28] J. M. Rouly, J. D. Orbeck, and E. Syriani. Usability and Suitability Survey of Features in Visual Ides for Non-Programmers. In *5th Workshop on Evaluation and Usability of Programming Languages and Tools*, pages 31–42, 2014.

[29] D. C. Schmidt. Guest editor's introduction: Model-driven engineering. *Computer*, 39(2):25–31, Feb 2006.

[30] A. Seffah, R. Djouab, and H. Antunes. Comparing and Reconciling Usability-Centered and Use Case-Driven Requirements Engineering Processes. In *2nd Australasian User Interface Conference*, pages 132–139, 2001.

[31] B. Selic. Model-driven development: its essence and opportunities. In *9th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing*, pages 7 pp.–, April 2006.

[32] A. Sinha and C. Smidts. An experimental evaluation of a higher-ordered-typed-functional specification-based test-generation technique. *Empirical Software Engineering*, 11(2):173–202, 2006.

[33] I. Sommerville. *Software Engineering*. Addison-Wesley, 2001.

[34] D. Stone, C. Jarrett, M. Woodroffe, and S. Minocha. *User Interface Design and Evaluation*. Interactive Technologies. Elsevier Science, 2005.

[35] Y. Weber, R. ; Zhang. An analytical evaluation of niam's grammar for conceptual schema diagrams. *Information Systems Journal*, 6(2):147–170, 1996.

[36] C. T. R. Weiss, D. M.; Lai. *Software Product-Line Engineering: a family based software development process*. Addison-Wesley, 1999.

[37] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering: An Introduction*. Springer - Verlag, 2012.

ABOUT THE AUTHORS:

Ildevana Poltronieri is a PhD student in Computer Science. She received her Master in Computer Science from Pontifical Catholic University of Rio Grande do Sul – PUCRS in 2018, her Bachelor in Information Systems from PUCRS in 2016, and a degree in Mathematics from URCAMP in 2003. She also holds a specialization in Applied Mathematics in 2006. She has experience in the area of Computer Science, with emphasis on Information Systems. She is currently a researcher collaborator in the Laboratory of Empirical Studies in Software Engineering - LESSE UNIPAMPA. She researches in the fields of Human-Computer Interaction and Software Engineering. Her research interests include Software Product Lines, Domain-Specific Language, and Usability Evaluation.

Avelino Francisco Zorzo is a full time professor at Pontifical Catholic University of Rio Grande do Sul (PUCRS), coordinator for professional post-graduate accreditation for Ministry of Education in Brazil, and Member of Brazilian Computer Society. He received his Computing Science PhD from Newcastle University in England in 1999, and a post doctorate at the Cybercrime and Computer Security Centre at Newcastle University in 2012. He was Industry Liaison and Education Director at Brazilian Computing Society, head of Computing Science School at PUCRS, member of Ethical in Research Committee at PUCRS, director at IT Users Society in Rio Grande do Sul, and member of Board for the Association of IT Companies in Rio Grande do Sul. His main research interests are security, dependability, and operating systems. He has published more than 100 papers in refereed conferences and journals.

Maicon Bernardino received his PhD in Computer Science at PUCRS in 2016. He worked for the CST in Analysis and Systems Development at FTEC, the CST in Digital Games at FEEVALE, and Federal University of Technology - Paraná (UTFPR). Currently, he works as an adjunct professor at Federal University of Pampa (UNIPAMPA) and the coordinator of Laboratory of Empirical Studies in Software Engineering (LESSE). His work experience involves Project Management, Performance Testing, Model-Based Testing, DSL, SPL, Gaming, Databases, and Software Testing.

Marcia de Borba Campos received her PhD in Informatics in Education from Federal University of Rio Grande do Sul in 2001. She is a CNPq researcher of the Brazilian Computer Society, and a member of the UX group from Association of IT Companies in Rio Grande do Sul. She worked for Pontifical Catholic University of Rio Grande do Sul as an adjunct professor (1996-2017). She was a member of the Ethical in Research Committee at PUCRS, an Internship coordinator at PUCRS, an Educational Technologies coordinator at PUCRS. Also, she worked as an adjunct coordinator of Virtual and a coordinator of assistive technologies in the Faculty of Computer Science at PUCRS. Her main research interests are Human-Computer Interaction, Usability, UX Design, Assistive Technologies, and Informatics in Education and Games. She has published more than 80 papers in refereed conferences and journals.