

Heuristic Evaluation Checklist for Domain-specific Languages

Ildevana Poltronieri¹^a, Avelino Francisco Zorzo¹^b, Maicon Bernardino²^c, Bruno Medeiros² and Marcia de Borba Campos³^d

¹PUCRS, Av. Ipiranga, 6681, Partenon, Porto Alegre, RS, Brazil

²UNIPAMPA, Av. Tiaraju, 810 - Ibirapuitã, Alegrete, RS, Brazil

³CESUCA, Rua Silvério Manoel da Silva, 160, Colinas, Cachoeirinha, RS, Brazil

Keywords: Checklist Heuristic, Empirical Method, Heuristic Evaluation, Domain-specific Language.

Abstract: Usability evaluation of a Domain-Specific Language (DSL) is not a simple task, since DSL designers effort might not be viable in a project context. Hence, we ease DSL designers work by providing a fast and simple way to evaluate their languages and, therefore, reduce effort (spend time) when a new DSL is developed. In order to do that, this paper presents a structured way to build a Heuristic Evaluation Checklist (HEC) for DSLs. This checklist is different from traditional checklists since it is focused on DSL. Once a checklist is provided, the evaluators only follow a set of heuristics and freely point out the found errors when using the DSL. Basically, the produced checklist provides a set of questions, based on the heuristics that direct an evaluation for a specific domain. In order to show how our proposal can be applied to a DSL and to provide an initial evaluation of our HEC, this paper shows also an instance to evaluate graphical and textual DSLs. Furthermore, this paper also discusses the qualitative analysis of an initial evaluation for the proposed HEC through seven interviews with Human-Computer Interaction (HCI) experts. Finally, a brief example of use applying the developed checklist is presented.

1 INTRODUCTION


In recent years we have seen a significant increase in the development of DSLs. Along with this development came the need to evaluate the usability of DSLs under development, so that later they are implemented and distributed to end users. Nonetheless, this type of evaluation is still a problem for groups that develop DSLs (Barišić et al., 2018) (Mosqueira-Rey and Alonso-Ríos, 2020) (Poltronieri et al., 2018).


To help solving that problem, the Human-Computer Interaction (HCI) field (Sharp et al., 2019) presents several approaches to evaluate the usability of any artifact. Among them, the most common used approaches for usability evaluation process are: Usability Testing (UT) (Nielsen, 1993) and Heuristic Evaluation (HE) (Nielsen and Molich, 1990). The former is a black-box approach technique whose goal is to observe real users using the product to uncover


potential issues and points for improvement. The latter is a knowledge-based inspection method. Nielsen's heuristic evaluation method (Nielsen, 1994) is the best known one, which is a heuristic-guided inspection that provides general heuristic principles of good interface design, *i.e.* aimed at maximizing usability of the evaluated software artifact.


Furthermore, the evolution of software systems increasingly relies on the search for specialized solutions to an application domain. Thus, Model-Driven Engineering (MDE) (Schmidt, 2006) highlights the important role played by Domain-Specific Languages (DSL) (Van Deursen et al., 2000), which provide constructions based on notations and abstractions that are specific to the problem domain. However, a research gap is the process of evaluating such DSLs regarding their usability.

In this context, some researchers have proposed ways to evaluate or organize the evaluation of DSLs (Mosqueira-Rey and Alonso-Ríos, 2020) (Poltronieri et al., 2018). For example, the study proposed by Poltronieri *et al.* (Poltronieri et al., 2018) describes an Usa-DSL Framework composed of steps, phases and

^a <https://orcid.org/0000-0002-8830-4230>

^b <https://orcid.org/0000-0002-0790-6759>

^c <https://orcid.org/0000-0003-2776-8020>

^d <https://orcid.org/0000-0002-5417-2524>

activities to help in the usability evaluation of DSLs. The framework structure is derived from the project life-cycle process (Stone et al., 2005). In their framework, steps are defined in eleven (11) focus areas; phases are composed of four (4) cycles of execution; and, activities are formed by a set of thirty two (32) concepts that are distributed among phases. The framework was designed to be used based on the needs of each assessment.

Usa-DSL Framework Evaluation was performed through interviews and a focus group involving subjects with experience in Human-Computer Interaction (HCI) and Software Engineering (SE). The evaluation aimed to present the framework and to obtain the opinion of subjects as to its clarity, ease of use and understanding. Based on those evaluations, the authors concluded that such framework needed a systematic support for its execution. Thus, to systematize its use, the Usa-DSL Process is needed. Such process may provide guidelines for the phases, steps and activities of the framework. In the process, definitions were mapped to the profiles, the details of the tasks performed by each activity, as well as the creation of work products that support the inputs and outputs of the process activities. Those work products are classified in the process according to their type: templates, checklists, tools, metrics, etc.

Therefore, the objective of this paper is to present a checklist for usability evaluation called **Heuristic Evaluation Checklist (HEC) for DSL**. Furthermore, we also instantiated this HEC for one DSL to demonstrate the applicability in one context. We also discuss the results of the evaluation through interviews with HCI experts to verify the completeness of the proposed checklist for the heuristic evaluation process in DSLs.

This paper is organized as follows. Section 2 presents concepts on DSLs and HE, and highlights related work. Section 3 introduces our HEC for graphical and textual DSLs. Section 4 outlines a systematic method for interviews, illustrates its testing via a pilot test, presents profile and discusses qualitative analysis. Section 5 presents a toy example to which our HEC was applied to. Section 6 finalizes this paper.

2 BACKGROUND

This section presents the main concepts on Domain-Specific Languages and Heuristic Evaluation. Also, related work associated with our research is presented in this section.

2.1 Domain-specific Languages

According to Van Deursen *et al.* (Van Deursen et al., 2000) a Domain-Specific Language (DSL) is “a programming language or executable specification language that offers, through appropriate abstractions, focused expressive power and usually it is restricted to a specific problem domain”. Like other languages, DSLs must have a set of sentences well known by their own syntax and semantics. Fowler (Fowler, 2010) asseverates that a DSL is defined as “a computer programming language with limited expressiveness and focused on a particular problem domain”.

The application of DSLs allows software to be developed faster and more effectively. The major advantage observed in using a DSL is that the knowledge required for its applicability is abstracted to another level. In this way, domain experts can understand, validate, and modify code, tailoring the model to their needs, making the impact of change easier to understand. There is still a significant increase in productivity, reliability, ease of use and flexibility (Van Deursen et al., 2000). According to Mernik (Mernik et al., 2005), DSLs can be classified under three different dimensions: **origin**, **appearance** and **implementation**.

Regarding the **origin** of a DSL, it can be internal and external. An **internal** DSL is designed from the syntactic and semantic rules of an existing language, which may be a general purpose language, or another DSL. An **external** DSL is language that relies on its own infrastructure for lexical, syntactic, semantic analysis, interpretation, compilation, optimization, and code generation.

With regard to the **appearance** dimension, a DSL can be classified as **textual**, **graphical**, **tabular** and **symbolic**. When in textual format, a DSL allows the domain to be expressed with characters, which are then combined to generate words, expressions, sentences and instructions that follow the grammar rules previously established in the language. Non-textual DSLs follow the same logic, but using graphical models to allow the user to express the domain knowledge with a higher level of understanding and using symbols, tables, figures and connectors.

As far as the **implementation** dimension is concerned, DSLs can be classified from the perspective of their implementation. These classifications form four groups: (i) well-known execution DSL; (ii) DSL that serve as input to application generators; (iii) non-executable DSLs but useful as input to application generators; (iv) DSL not designed to be executed.

In general the main consideration for building a DSL should be its origin as each approach has specific advantages and disadvantages that are inherent in each

type (Fowler, 2010). Although external DSL may have an effort associated with building it many times higher than that of an internal DSL, there are currently tools that support DSL development. These tools are known as **Language Workbenches (LWB)** and apply language-oriented programming concepts to provide a higher level of abstraction for complex infrastructure issues (Fowler, 2005).

2.2 Heuristic Evaluation

Heuristics are based on practical knowledge that comes from continued daily experience. Heuristic knowledge builds over years of practice as a compilation of ‘what works’ and ‘what does not’.

Heuristic Evaluation (HE) is an **inspection method** that does not involve end users. In HE, analysis is performed by experts who advocate for the user, that is, knowing what the users’ wants and needs, and knowing the possible HCI techniques they evaluate whether a particular computational artifact provides a good experience for the user (Sharp et al., 2019).

The purpose of a HE is to **identify problems** (for a particular user profile and task set), consisting of a group of three (3) to five (5) HCI experts who inspect the interface or, in this case, the DSL without involving users using a heuristic list (guidelines), empirical basis with the intention of generating as a result a report of potential problems and recommendations for the evaluated solution (Nielsen, 1993).

Heuristic evaluation is a method designed to find usability issues during an iterative design process. It is a simple, fast and inexpensive method to evaluate HCI when compared to empirical methods. The method is based on a set of usability heuristics, which describe desirable interaction and interface features.

The **heuristic evaluation method** was proposed by Jakob Nielsen in 1994 (Nielsen, 1994). This is a heuristic-driven inspection in which general principles of good interface design aims at maximizing artifact usability. Traditionally, ten (10) heuristics (Nielsen, 1994) have been used to cover new technologies and computing environments. These heuristics have already been altered and expanded since their original proposal. In the literature, several papers (Sadowski and Kurniawan, 2011) (Sinha and Smidts, 2006) have proposed heuristic checklists based on Nielsen’s heuristics to inspect the usability of computational solutions.

2.3 Related Work

Several works have applied usability evaluation methods to assess Domain-Specific Languages (DSLs). In

this section, we mention the ones that are directly related to our proposal.

Barišić *et al.* (Barišić et al., 2018) introduced a conceptual framework, called USE-ME, which helps the usability evaluation through an iterative incremental development process of DSLs. The framework is a systematic approach based on user interface experimental evaluation techniques. In their study, the authors presented the feasibility of the conceptual framework by means of an industrial case study.

Hermawati and Lawson (Hermawati and Lawson, 2016) presented a systematic review of seventy (70) studies related to usability heuristics for specific domains. The study objective was to map the heuristic evaluation processes applied to specific domains. Their work identified points for improvements and further research. The most important aspect pointed out by the authors was the lack of validation effort to apply heuristic evaluation as well as the robustness and accuracy of the applied validation method.

Sadowski and Kurniawan (Sadowski and Kurniawan, 2011) derived eleven (11) Language Feature Heuristics through potential heuristics based on Nielsen’s ten (10) heuristics and thirteen (13) cognitive dimensions framework.

Sinha and Smidts (Sinha and Smidts, 2006) presented an approach that uses a measurement model to evaluate usability indicators of a DSL. The authors pointed out that usability measures were defined for their applications or, more specifically, for the graphical interfaces of these applications.

Poltronieri *et al.* (Poltronieri et al., 2018) presented the Usa-DSL Framework to guide a systematic usability evaluation for Textual and Graphic DSL. This is the framework we will follow in our work.

Recently, Mosqueira-Rey and Alonso-Ríos (Mosqueira-Rey and Alonso-Ríos, 2020) worked in a set of usability heuristics for DSLs. In their study, they introduced a case study for evaluating their approach. The proposed approach is based on the usability taxonomy published in their previous work (Alonso-Ríos et al., 2009), which is composed of the following attributes: *Knowability*, *Operability*, *Efficiency*, *Robustness*, *Safety* and *Subjective Satisfaction*.

Summarizing (see Table 1), Barišić *et al.* focuses on a conceptual framework process to support usability evaluation in DSLs. Hermawati and Lawson provide evidence of poor usability assessments of DSLs developed on the basis of a secondary study. Sadowski and Kurniawan discuss language feature heuristics. Sinha and Smidts present the evaluation on four heuristics proposed by Nielsen. Poltronieri *et al.* present a systematic framework for usability

Table 1: Related Work Summary.

Study	Criteria Analysis	Usability Evaluation Method
(Barišić et al., 2018)	Introduce a conceptual framework	Usability Evaluation
(Hermawati and Lawson, 2016)	Present a systematic review of seventy (70) studies related to usability heuristics for specific domains	Heuristic Evaluation
(Sadowski and Kurniawan, 2011)	Present evidences from the evaluation of two parallel programming case studies in order to evaluate the usability problems in programming languages.	Heuristic Evaluation and Cognitive Dimensions
(Sinha and Smidts, 2006)	Evaluation on four heuristics proposed by Nielsen	Heuristic Evaluation
(Mosqueira-Rey and Alonso-Ríos, 2020)	Present a set of usability heuristics for DSLs and introduce a case study to evaluate their approach	Not Informed
(Poltronieri et al., 2018)	A usability evaluation framework for domain-specific languages	Usability Test and Heuristic Evaluation

evaluation. Mosqueira-Rey and Alonso-Ríos present a heuristic evaluation based on a proposed usability taxonomy. The authors also present usability heuristics based on a taxonomy they built, but the methodology is very different from our Checklist Heuristic. The authors, also argue that their heuristics can help identifying real problems of usability, including even simple DSLs.

Complementary to these studies, the focus of our approach is to present a Heuristic Evaluation Checklist for DSL based on the concepts of the Nielsen’s ten heuristics. The **difference of our checklist** is that we direct the evaluation so that its results can be more intuitive and ease to use for DSL designers who want to know about the usability of their DSLs. As for the related work, the development of the Heuristic Checklist using Nielsen’s Heuristics is the key point of our proposal, since those heuristics are consolidated in the area and with that it facilitates the recognition of its concepts when the evaluators apply our checklist.

3 HEURISTIC EVALUATION CHECKLIST

The Heuristic Evaluation (HE) method is a widely used approach for usability inspection, in which a group of evaluators inspect an interface design based on a set composed of ten (10) usability heuristics and a severity score rating from 0 to 4 for each encountered problem (Nielsen, 1994) (Nielsen and Molich, 1990).

Although heuristic evaluation is frequently used for usability assessment, these heuristics are used to evaluate user interfaces for many different domains. In some studies, heuristics adjustment are needed to ensure that specific usability issues of certain domains are not overlooked (Hermawati and Lawson, 2016). Several authors use an informal process to develop or

adapt usability heuristics and do not follow an established and systematic protocol for heuristic evaluation. In our approach we **use a set of heuristics to evaluate the usability of applications with specific features, and specific aspects not covered by generic sets of usability heuristics.**

This adaptation was based on questions related to our research domain and the 10 heuristics proposed by Nielsen. One of the main goals of this checklist is to enable teams that are part of different phases of the development process to understand and evaluate their application. In our evaluation we apply that to DSLs.

This approach not only brings the DSL design team closer to the HE method, but also assists the HCI experts who will evaluate the DSL in understanding the problem domain that will be evaluated. For a better understanding of the HE methodology extension, we describe it in the next section.

The methodology to develop our Heuristic Evaluation Checklist follows the methodology proposed by (Quiñones et al., 2018) (see Figure 1). The first steps were: to understand Heuristic Evaluation and Domain-Specific Language concepts, to adapt existing heuristics for the DSL domain, to produce a set of questions based on a systematic literature review (Rodrigues et al., 2017), and to create our initial Heuristic Evaluation Checklist. After that we submitted this preliminary checklist to be evaluated by a set of HCI experts through interviews. Finally, we got a modified, and final, Heuristic Evaluation Checklist that we applied to an example of use.

3.1 A Heuristic Evaluation Checklist for Graphical and Textual DSLs

Heuristic Evaluation Checklist (HEC) for graphical and textual DSLs is based on extension of artifacts from the HE method. This checklist is an artifact de-

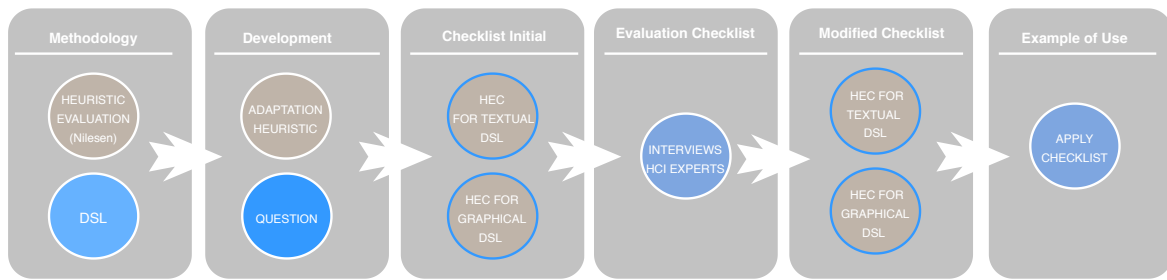


Figure 1: Methodology.

signed to guide the heuristic evaluation of DSLs and it must be used in the context of a usability evaluation process for DSLs. This evaluation should be planned by an analyst, developer or DSL tester, and be conducted by heuristic evaluation experts.

The checklist structure consists of five (5) columns (see Table 3): the first column contains the identification of the heuristic, the second column is related to the description of the ten (10) Nielsen’s heuristics, but adapted to the context of DSLs, the third column refers to the questions that guide the evaluation and related to each one of the heuristics, the fourth column is the severity degree of the usability problems found and the fifth column is designed for the description of issues found in the DSLs. A snippet of the Heuristic Evaluation Checklist for graphical and textual DSLs is shown in Table 3¹.

This checklist is intended to guide the evaluation of several kinds of DSL. Thus, three distinct versions were created: Heuristic Evaluation Checklist for textual DSLs; Heuristic Evaluation Checklist for graphical DSLs; and, Heuristic Evaluation Checklist for graphical and textual DSLs.

The Heuristic Evaluation Checklist for graphical and textual DSLs cover questions related to both types of DSLs. In this checklist, the first heuristic, “H1: Visibility of system status”, for instance, has as description: “The DSL should always keep users informed about what is going on, through appropriate feedback within reasonable time.” and it is guided by three distinct questions.

The first question is “Does the graphical DSL provide immediate and adequate feedback on their status for each user action? (For example, after an include or exclude task, the language displays a commit message)?”. The second question: “Do the elements available for the user specifically execute only one command? (For example, the ‘undo’ button only performs

undo actions)”. The last question (not shown in Table 3): “Does the textual DSL provide immediate and adequate feedback on the status of each user action? (For example, after an include or exclude task, the language displays a commit message)?”.

The Heuristic Evaluation Checklist for textual DSLs is composed of the same heuristics and descriptions. However, two questions guide the first heuristic on the checklist for textual DSLs: “Does the Textual DSL provide immediate and adequate feedback on their status for each user action? (For example, after an include or exclude task, does the language display a commit message?)” and “Do the elements available for the user specifically execute only one command? (For example, do the keywords on the Textual DSL are used for specific purposes?)”.

The same heuristic on the Heuristic Evaluation Checklist for graphical DSLs has also two guiding questions, which have the same content of the guiding questions on the Heuristic Evaluation Checklist for textual DSLs but focused on the visual aspects of the DSL. To measure the severity degree of the found usability problems Table 2 was used.

Table 2: Severity Degree.

Sev.	Type	Description
0	Not applicable	I don’t agree that this is a usability problem at all
1	Cosmetic problem only	Doesn’t need to be fixed unless extra time is available on the project
2	Minor usability problem	Fixing this should be given low priority
3	Major usability problem	Important to fix, so should be given high priority
4	Usability catastrophe	Imperative to fix this before the product can be released

The use of our checklist is different from previous checklists since each of the heuristics is guided by questions that direct the evaluation to what one seeks to evaluate in a DSL. In the original heuristic evaluation,

¹Due to space restrictions, only some information related to our HEC is presented in this paper. Our complete HEC can be found at <https://drive.google.com/drive/folders/1obuVQ-67P49fnqMqB-SNJr7JHUsFVHQ?usp=sharing>.

the evaluators only follow the heuristics and freely point out the errors found when using the system.

This checklist was directed to a specific need, *i.e.* to evaluate DSLs. Our methodology intends to make the DSL evaluation clearer and more direct, easing the evaluators task (HCI Experts), even for evaluators that might not be familiar with the DSL domain. Therefore, this study presents, as an example of use, a Heuristic Evaluation Checklist for graphical and textual DSL.

4 METHOD: CHECKLIST EVALUATION

To evaluate the Heuristic Evaluation Checklist for graphical and textual DSL we used the qualitative analysis approach that was performed through online interviews. The Interview Method (Clark et al., 2019) was chosen because the main goal of this study was to obtain the respondents perception about the checklist content and presentation. The interviewees were invited from participants' references, thus ensuring that they were experts on HCI or at least had seen usability evaluation before.

Before the interviews, the documents necessary to the understanding of the analyzed domain, *i.e.* DSLs, were sent to the interviewees, and also the tasks that would be performed were clarified. Thereafter, the participants needed to perform the analysis of the checklist and to provide their contribution during the interview.

The list of documents that were provided to the experts before the interviews were *Informed Consent Term (ICT) and Profile Questionnaire, Survey Guide-line, and Information on the Interview.*

4.1 Pilot Test

The pilot test was a small trial to assure that the study was viable. This test checked whether the procedure and questionnaire questions were set properly, and to identify if the process and documents had any potential problem. Furthermore, during the pilot test, small adjustments were made to the main study documents and procedures. To validate the protocol and interview documents, we performed a pilot test with an HCI expert.

During the pilot test, the pilot subject had access to the documents in the same way that actual participants would have, in order to obtain the faithful perception of what would be carried out during the interviews. We verified the duration of the test, the understanding of the messages sent by e-mail, the level of knowledge

on the topic to be evaluated and, finally, the analysis of the checklist.

The pilot subject suggested some improvements on the checklist, for example, to add some examples and to include a text box at the end of the instrument, so that the HCI expert could mention problems that would not fit the issues present in the questionnaire. In general, the pilot subject believed that the instructions on the invitation e-mail were clear and met the purpose of the study, as well as the tasks and DSL examples used to accomplish the tasks.

4.2 Profile

For the interviews, seven (7) participants were recruited by e-mail. This sample was selected looking for experienced professionals in HCI (researchers that published relevant papers on HCI conferences). Furthermore, participants were also invited when recommended by other participants who were considered experts. After acceptance, the documents for the interview were sent to them. The Profile Questionnaire was used to identify the experience of the participant and other relevant information. In order to obtain the information regarding the participants' experience, the following questions were asked: **Q1** - What is your name?; **Q2** - What is your work position?; **Q3** - Which usability evaluation method(s) have you already participated in (Heuristic Evaluation, Usability Testing or None)?; **Q4** - Which usability evaluation method(s) have you already conducted (Heuristic Evaluation, Usability Testing or None)?; **Q5** - What is your level of expertise related to HCI (Very Poor, Poor, Neutral, Strong or Very Strong)?; **Q6** - What is your level of expertise related to Usability (Very Poor, Poor, Neutral, Strong or Very Strong)?; and, **Q7** - What is your level of expertise related to Heuristic Evaluation (Very Poor, Poor, Neutral, Strong or Very Strong)?

In this paper the answers of the profile questionnaire, as well as the other answers of the study were identified by the label attributed to each participant, *i.e.* from P1 to P7 (see Table 4).

In this study, as mentioned before, the participants were experts on HCI and most of them had already performed a Heuristic Evaluation. One participant had no experience on Heuristic Evaluation, but had a strong level of expertise on HCI, and his consideration on the checklist was relevant on the view of a DSL designer. The other participant who had no knowledge on Heuristic Evaluation had experience on Usability Testing and a very deep knowledge on HCI. The reported experience could be perceived from the responses captured during the interviews.

Table 3: Snippet for our HEC.

Heuristic	Description	Question	Severity (Check each of the problems found)					Description of each error occurrence
			0	1	2	3	4	
H1: Visibility of system status	The DSL should always keep users informed about what is going on, through appropriate feedback within reasonable time.	Does the DSL provide immediate an adequate feedback on its status for each user action? For example, after an include or exclude task the language displays a commit message?						
		Do the elements available for the user specifically execute only one command? For example, the “undo” button only performs undo actions.						

Table 4: Subjects Profile.

Q1	Q2	Q3			Q4			Q5		
		HE	UT	None	HE	UT	None	5.1	5.2	5.3
P1	Professor HCI field	X	X		X	X		V	V	V
P2	Quality Assurance Engineer	X	X		X			S	N	N
P3	Professor/Developer			X			X	N	N	N
P4	Professor HCI field		X			X		V	S	S
P5	Professor HCI field	X	X		X	X		S	N	N
P6	Professor HCI field	X	X		X	X		V	S	S
P7	PhD.Candidate in Computer Science	X	X		X	X		S	S	V

HE - Heuristic Evaluation, UT - Usability Testing V - Very Strong, S - Strong, N - Neutral

4.3 Interviews

The interviews started after the pilot test and after each participant had submitted the ICT. The interviews were conducted over a period of six (6) months between December 2018 and June 2019. The execution of the interviews were predominantly online (5 online and 2 in-person). All interviews were audio recorded in order to perform further analysis. Each interview lasted an average of 60 minutes. The interviews were semi-structured, providing a certain flexibility to adjust questioning based on participant responses. Each interview covered five central topics: (1) Definition of heuristics; (2) Checklist’s structure and organization; (3) Checklist’s content; (4) Amount of information displayed; (5) Checklist’s template.

Each topic is guided by questions that direct the interview purpose:

- **Topic 1** - the first topic was assessed by two questions: **T1Q1** - Are the heuristic definitions appropriate for assessing a DSL? **T1Q2** - Is it possible to understand the objective of each heuristic after reading its respective definition?
- **Topic 2** - the second topic was assessed by three

questions: **T2Q1** - Does the order in which heuristics are organized adequate? **T2Q2** - Do you think heuristics should be grouped, for example, in relation to graphical aspects or documentation? **T2Q3** - If so, which heuristics should appear at the beginning or end of the checklist?

- **Topic 3** - the third topic was assessed by five questions: **T3Q1** - Do the questions that correspond to each of the heuristics reflect their purpose? **T3Q2** - In your opinion, are the above questions linked to heuristics clearly and unambiguously? **T3Q3** - Is there any question you do not understand? **T3Q4** - Would you add any question? Which ones? **T3Q5** - Would you remove any question? Which ones?
- **Topic 4** - the fourth topic was assessed by one question: **T4Q1** - The checklist is guided by 10 heuristics that are composed of around 32 questions in the most extensive Checklist. Regarding the extension of the checklist, what is your opinion?
- **Topic 5** - the last topic was assessed by one question: **T5Q1** - What do you think about the way the checklist is presented? (Heuristics, Definitions, Questions, Degree of Severity and description of

found errors).

After the interviews were completed, the opinions' transcription were performed and organized according to the Inductive Thematic Analysis method (Braun and Clarke, 2006). These analyzes are presented in the next section.

4.4 Qualitative Analysis

The interview analyses was performed using the Inductive Thematic Analysis method (Braun and Clarke, 2006), which categorizes the main themes gathered from the experts' responses. This approach is common on HCI qualitative research (Clark et al., 2019) (Luger and Sellen, 2016).

For the Inductive Thematic Analysis execution, the audios from the interviews were transcribed. Then, the content was coded by similarity, forming group themes. As the last step, two (2) researchers reviewed the created themes, making some adjustments to best represent the obtained information. The analysis is presented next.

Each theme has a summary description and quotes that supports the theme's objective.

4.4.1 Checklist's Description

This theme presents the opinions related to the descriptions used in the heuristic checklist. As the proposed checklist is an instrument customized to contemplate all the Nielsen's Heuristics focusing on the DSL domain, one of our main concerns was if the description of the heuristics was appropriated. All participants affirmed that the descriptions created would be useful for the checklist's execution. Participants P1 and P7, who have experience in HE and UT, highlighted the following:

"I think the definitions are appropriated. They are embracing and generic enough, following the pattern created by Nielsen but with a focus on DSLs." [P7]

"It is going to be natural for the evaluators to execute the evaluation as they can evaluate in a broad perspective and make annotations." [P1]

4.4.2 Detailing

Five participants (P1, P2, P3, P4 and P5) reported the need for more details related to the checklist presentation and also to its content. The main issues reported are related to the severity rating usage and the lack of questions in specific heuristics.

"It would be helpful to have a small text explaining the severity rating." [P3]

"The evaluator needs to describe the error in a clear way in order to assimilate the severity rating." [P4]

"There should be more questions about consistency. I notice that there are some questions related to patterns, but there is a lack of questions about consistency." [P2]

The above quotes highlight some improvements to be made to the checklist. While developing the customized checklist, we tried to avoid extensive texts in order to not cause fatigue when the evaluation process occurs. However, the feedback received from HCI experts emphasized the need for clarification on certain aspects of the checklist, such as the purpose of the severity classification and the extent of the questions.

4.4.3 Incomprehension

Four (4) participants (P1, P2, P4 and P6) made suggestions for improvements to the checklist or its structure. The suggestions were related to Heuristic 7 (Flexibility and Efficiency of Use) and on the execution of the checklist according to the severity of the classification provided rating.

"I see the severity rating here, but imagine that I did not find any error... What should I mark?" [P6]

The feedback analysis provided by the HCI experts led us to ask how the evaluators would perform the DSL evaluation using this checklist.

Participant P1 reported that the classification of the provided severity scale was not intuitive to use, arguing that it was difficult to understand its use. We mitigated this issue by applying the suggestions captured in the previous theme (Detailing) and adding a small text to guide the severity scale classification.

Another interesting feedback collected was related to Q3 and Q4 of Heuristics 7, as some evaluators disagreed on their answers. The statements of the HCI experts underline the need for minor modifications to make their purpose clearer.

4.4.4 Evaluator's Profile

Although Participant P3 was not an expert in usability evaluation, he had considerable experience in software development. Hence, P3 stated that the evaluation using the proposed checklist may be performed by any professional with background in system evaluation, even if this professional is not an expert in heuristic evaluation.

"I think it could be used by evaluators who are not experts in heuristic evaluations but have already evaluated systems using other methods." [P3]

The statement above highlights the adaptability of this instrument for a wide-range of professionals

related to usability.

4.4.5 Content Changes

Two participants (P1 and P4) suggested changes on the content of the heuristic checklist. In general, they stated that some questions needed to be reformulated and/or classified in another heuristic.

“Questions 3, 4 and 5 of Heuristic 3 must be classified in Heuristic 5” [P1]

“Well, I think that the question presented in Heuristic 10 needs to be reviewed.” [P4]

“When you ask about help and documentation... I don't think that documentation is important for this kind of evaluation, when I read documentation I think about a broad documentation about the system” [P4]

4.4.6 Template Changes

All participants reported the need of changing the order of the visual components that compose the checklist form. The main suggestions were related to the position of the elements and the lack of space for adding relevant information.

“I think that you could use colors to enhance the reading of the heuristics” [P5]

“Maybe the severity description could be at the beginning of the checklist, not at the end as it is. In this way the access for this information will be easier” [P7]

“It would be interesting to have a blank space for the evaluator to add some relevant issues” [P3]

In order to mitigate these issues, we reviewed the checklist's template and followed the suggestions of some participants, such as changing the order of problem description and severity rating, so that the checklist would be more intuitive to use; and adding a blank box for the evaluator to describe errors that were not contemplated by the used heuristics.

4.4.7 Instrument's Amplitude

Participant P7 pointed out one weakness of the checklist by reporting that if evaluators follow narrowly the questions presented in the checklist, maybe some errors presented in the DSL would not be found. We are aware of this weakness and we mitigated it by emphasizing that the role of the questions are to guide the evaluators on common usability problems and the evaluators must report other perceived problems in the extra blank box.

“Perhaps, I don't know if the evaluators would find all the errors if they followed only your questions.” [P7]

4.5 Discussion

This study presents the development of a HEC in which the checklist is guided by questions that conduct the evaluation for a specific domain.

The analyses performed on the interviews showed that this HEC can assist evaluators to conduct DSL evaluations. Furthermore, it was also noticed that a checklist guided by questions directed to the context of use provides a better understanding about the evaluation.

Some changes proposed by the HCI experts were applied to the final HEC (presented in Section 3). These changes were related to the content of the checklists, as well as, their structure and template. Such changes were pertinent so that it was possible to carry out the first study through an example of use (see Section 5), in which a textual DSL was analyzed. The purpose of performing the evaluation of this example of use was to get insights into the proposed Heuristic Evaluation Checklists for DSLs.

5 EXAMPLE OF USE

In order to verify the applicability of the proposed HEC, we asked five subjects to experiment it on an example of use. The example of use is a well-known DSL in the academic environment, *i.e.*, LaTeX. This language was chosen since all the participants had previous experience on using it. The following artifacts were chosen for this example of use: informed consent term, participants profile questionnaire, DSL guide, list of task to be performed, and a copy of the Heuristic Evaluation Checklist for textual DSLs².

It is worth mentioning that since the main objective of this study was to obtain a first view of the feasibility of the Heuristic Evaluation Checklist for textual DSL. Thus, we only sent the documentation via e-mail and collected the participants' perceptions regarding the heuristic evaluation of LaTeX through our checklist proposal. Hence, from the responses from the participants, we performed a qualitative analysis regarding each of the found problems and their severity.

5.1 Analysis

Before discussing the results for each of the ten heuristics in our HEC, we describe the participants profile:

²Due to space restrictions, only the main information related to the example of use is presented in this paper - all example of use can be found at <https://drive.google.com/file/d/1KKhse6OgnKbqRtreW-zZOGCehmFnILWS/view?usp=sharing>.

one is undergraduate student and four are master students in Software Engineering; the average time of experience using or designing DSL is 2.4 years; all participants have experience in performing or participating in usability evaluation; three participants have just one year experience in usability evaluation; two of them have already participated on usability evaluation using usability testing; two of them have already participated of usability evaluation using heuristic evaluation; just one of the participants has already conducted a usability evaluation using heuristic evaluation.

The analysis of the results points for each heuristic as follows:

H1 Q2: Participants E2, E3 and E5 agreed that there is no undo button and mentioned that the action is only possible using the Ctrl + Z keys. Regarding the degree of severity, two of the participants believed that fixing this should be given low priority.

H2 Q3: Participant E5 considered that LaTeX has abbreviated keywords, and that this makes it difficult for other users to adopt this language. He also reported that it is important to fix this problem, *i.e.* it should be given high priority.

H3 Q6: Participant E3 reported that some errors are shown in real time and others only after compilation. The participant also indicated that LaTeX does not provide information about commit, and this may be part of the tool that instantiates the language. Regarding the degree of severity for this problem, the participant considered it a cosmetic problem.

Q7: Regarding H3, four participants (E1, E2, E3 and E5) reported that there are synchronization problems, the environment warns that changes are made in a certain period of time and that they may not have been saved. Furthermore, they also stated that the changes are saved automatically, but if there are internet connection problem, the re-connection message may not be accurate and changes in the document might not have been saved. E1 mentions: "The problems are only showed after the .tex compilation". This question had disagreement of the degree of severity among participants, *i.e.* 1, 2 and 4. Hence, there was no consensus among them related to this question.

H4 No problem found.

H5 Q11: Participants E1, E4 and E5 mentioned that the environment does not have confirmation boxes or buttons for actions. E1 assigned severity degree 2, while E4 and E5 assigned severity degree 1.

H6 No problem found.

H7 Q16: The participants mentioned the following for this question: E2 mentions that if someone considers the generated pdf as an output, in this case the changes do occur; E4 states that there is no graphic DSL, only the preview of the text written in the generated pdf; and, E5 considers that only when the changes to the textual DSL are compiled, they are observed in the graphical DSL. The severity degree assigned is zero, so they do not consider any usability problem here.

Q19: Participants E2 and E3 mentioned that to have a color change, LaTeX commands must be entered. These participants considered that this is not a usability problem and attributed zero to the degree of severity.

H8 Q20: Participants E2, E3 and E4 stated that the error messages are not intuitive or easy to understand, on the contrary they are confusing and hinder rather than help. Such messages are difficult to quickly identify the problem, so it is often necessary to have technical knowledge to deal with errors. Therefore, E2 assigned 3 to the degree of severity, while E3 and E4 assigned 2 to the degree of severity.

H9 Q21: Participants E1, E2 and E5 mentioned that there is no tutorial for LaTeX, however the templates have it. Moreover, related to the degree of severity attributed by those who say there is no documentation ranges from cosmetics to usability catastrophe.

5.2 Discussion on the Example of Use

Our Heuristic Evaluation Checklist helped the participants to respond more adequately the questions regarding the evaluation of the DSL. The ones that had previous experience on heuristic evaluation performed better than the ones that did not have previous experience. Hence, it seems that, despite our HEC helping the participants to evaluate a DSL, it needs some further instructions on how to fill the questionnaires. Maybe this can help not so experienced participants to better understand and answer the questions.

6 CONCLUSION

The Heuristic Evaluation (HE) method is a widely used approach for usability inspection. This method is easy to perform and it allows the discovering of various usability issues. The HE method is freely applied by the evaluators. These evaluators go through

the application interface pointing out the errors, and consequently classifying them to the degree of severity.

Although Heuristic Evaluation is frequently used for usability evaluation, heuristics are used to evaluate user interfaces for many different domains. For this reason, many researchers adapt the heuristics to their application domain. Several authors use an informal process to develop or adapt usability heuristics and do not follow an established and systematic protocol for heuristic assessment.

Our approach presents a different strategy to apply HE. The proposal provides a checklist that is different from existing solutions. In our proposal, each heuristic is guided by questions that direct the evaluator to effectively evaluate the DSL.

Regarding the evaluation of the created HEC, we performed 7 (seven) interviews with researchers and professionals on HCI. These interviews were analyzed using the Inductive Thematic Analysis method, which resulted in a group of common themes of opinions, suggestions and ideas discussed in this study. These results led us to make improvements in the HEC.

Some of the most important questions provided by the HCI experts were related to the depth and amplitude of the questions and how the evaluators would use the checklist to perform the heuristic evaluation on a DSL. Suggestions regarding the checklist's template, such as changing the order the columns and providing the evaluator with a blank box to freely report problems not yet classified, were discussed and accepted.

Moreover, we also engaged in a deep discussion around covering all the aspects of Textual and Graphical DSLs using the Nielsen's heuristics. We decided to create more questions for specific heuristics, such as Flexibility and Efficiency of Use (H7). However, it must be stated that the checklists created must be seen by the evaluators as a guidance to perform the inspection and if they find usability problems not covered by the content of the checklist, they must describe them in the provided blank box.

Based on the opinions obtained through the interviews, we made some changes to the HEC, which were suggested by the interviewees. After completing these changes, we applied the proposed checklist using an example of use, in order to understand its behavior when used in a context close to a real one. Thus, this usage example included (5) participants, who used the Latex textual DSL. Preliminary results showed that our HEC helps experienced and non experienced usability evaluators, but some improvements are needed for non experienced usability evaluators.

As future work, we aim to apply some Norman design principles (Norman, 2013) to our proposal, *i.e.* affordance, cognitive overload, and visibility. We

still intend to analyze the applicability of the created checklists in several real scenarios in order to understand how the evaluation procedure will be performed with those artifacts.

ACKNOWLEDGEMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. Avelino F. Zorzo is supported by CNPq (315192/2018-6).

REFERENCES

- Alonso-Ríos, D., Vázquez-García, A., Mosqueira-Rey, E., and Moret-Bonillo, V. (2009). Usability: A critical analysis and a taxonomy. *International Journal of Human-Computer Interaction*, 26(1):53–74.
- Barišić, A., Amaral, V., and Goulao, M. (2018). Usability driven dsl development with use-me. *Computer Languages, Systems & Structures*, 51:118–157.
- Braun, V. and Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101.
- Clark, L., Pantidi, N., Cooney, O., Doyle, P., Garaialde, D., Edwards, J., Spillane, B., Gilmartin, E., Murad, C., Munteanu, C., Wade, V., and Cowan, B. R. (2019). What makes a good conversation?: Challenges in designing truly conversational agents. In *Conference on Human Factors in Computing Systems (CHI)*, pages 475:1–475:12. ACM.
- Fowler, M. (2005). Language Workbenches: The Killer-App for Domain Specific Languages?
- Fowler, M. (2010). *Domain Specific Languages*. Addison-Wesley Professional, 1st edition.
- Hermawati, S. and Lawson, G. (2016). Establishing usability heuristics for heuristics evaluation in a specific domain: Is there a consensus? *Applied Ergonomics*, 56:34 – 51.
- Luger, E. and Sellen, A. (2016). “like having a really bad pa”: The gulf between user expectation and experience of conversational agents. In *Conference on Human Factors in Computing Systems (CHI)*, pages 5286–5297. ACM.
- Mernik, M., Heering, J., and Sloane, A. M. (2005). When and how to develop domain-specific languages. *ACM Computing Surveys*, 37(4):316–344.
- Mosqueira-Rey, E. and Alonso-Ríos, D. (2020). Usability heuristics for domain-specific languages (dsls). In *SAC*, pages 1340—1343.
- Nielsen, J. (1993). *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Nielsen, J. (1994). 10 Usability Heuristics for User Interface Design. Available in: <https://www.nngroup.com/articles/ten-usability-heuristics/>.

- Nielsen, J. and Molich, R. (1990). Heuristic evaluation of user interfaces. In *Conference on Human Factors in Computing Systems*, pages 249–256. ACM.
- Norman, D. (2013). *The Design of Everyday Things: Revised and Expanded Edition*. Basic Books.
- Poltronieri, I., Zorzo, A. F., Bernardino, M., and de Borja Campos, M. (2018). Usa-DSL: Usability Evaluation Framework for Domain-specific Languages. In *SAC*, pages 2013–2021. ACM.
- Quiñones, D., Rusu, C., and Rusu, V. (2018). A methodology to develop usability/user experience heuristics. *Computer Standards & Interfaces*, 59:109 – 129.
- Rodrigues, I., Campos, M. B., and Zorzo, A. (2017). Usability Evaluation of Domain-Specific Languages: a Systematic Literature Review. In *International Conference on Human-Computer Interaction*, pages 522–534. Springer.
- Sadowski, C. and Kurniawan, S. (2011). Heuristic evaluation of programming language features: Two parallel programming case studies. In *Workshop on Evaluation and Usability of Programming Languages and Tools*, pages 9–14. ACM.
- Schmidt, D. C. (2006). Guest editor’s introduction: Model-driven engineering. *Computer*, 39:25–31.
- Sharp, H., Preece, J., and Rogers, Y. (2019). *Interaction Design: Beyond Human - Computer Interaction*. Wiley Publishing, 5th edition.
- Sinha, A. and Smidts, C. (2006). An experimental evaluation of a higher-ordered-typed-functional specification-based test-generation technique. *Empirical Software Engineering*, 11(2):173–202.
- Stone, D., Jarrett, C., Woodroffe, M., and Minocha, S. (2005). *User Interface Design and Evaluation*. Interactive Technologies. Elsevier Science.
- Van Deursen, A., Klint, P., and Visser, J. (2000). Domain-specific languages: An annotated bibliography. *SIGPLAN Notes*, 35(6):26–36.