Adoption of the HTA Technique in the Open Source Software Development Process

Rosa Llerena[1], Nancy Rodríguez[2](✉), Lucrecia Llerena[2], John W. Castro[3],
and Silvia T. Acuña[2]

[1] Ministerio de Educación, Quevedo, Ecuador
rosa.llerena@educacion.gob.ec
[2] Departamento de Ingeniería Informática, Universidad Autónoma de Madrid, Madrid, Spain
{nrodriguez,llerena}@uteq.edu.ec, silvia.acunna@uam.es
[3] Departamento de Ingeniería Informática y Ciencias de la Computación,
Universidad de Atacama, Copiapó, Chile
john.castro@uda.cl

Abstract. The growth in the number of non-developer open source software (OSS) application users and the escalating use of these applications have led to the need and interest in developing usable OSS. OSS communities do not generally know how to apply usability techniques and are unclear about which techniques to use in each activity of the development process. The aim of our research is to adopt the HTA usability technique in the OpenOffice Writer OSS project and determine the feasibility of adapting the technique for application. To do this, we participated as volunteers in the project. We used the case study research method during technique application and participation in the community. As a result, we identified adverse conditions that were an obstacle to technique application and modified the technique to make it applicable to OSS projects. We can conclude from our experience that these changes were helpful for applying the technique, using web artifacts like forums and collaborative tools like Cacoo, although it was not easy to recruit OSS users to participate in usability technique application.

Keywords: Open source software · Usability techniques · Design · Hierarchical Task Analysis (HTA)

1 Introduction

OSS has spread so swiftly that it now rivals commercial software systems [1]. OSS communities do not as yet enact standard processes capable of ensuring that the software that they develop has the attributes of good software [2]. The inadequate definition of processes, activities, tasks and techniques within OSS development has led researchers from several areas to gravitate towards this field of research with the aim of correcting this situation. Usability is one of the key quality attributes in software development. In recent years, OSS has come to be an important part of computing.

However, several authors have acknowledged that the usability of OSS is poor [3–5].

In this respect, the empirical study conducted by Raza et al. [6] reports that 60% of respondents (non-developer users) stated that poor usability is the main obstacle that OSS applications have to overcome if users are to migrate away from commercial software. On this ground, OSS projects must tackle the usability level and usability-related problems at length [5].

On one hand, the HCI field offers usability techniques whose key aim is to build usable software. However, they are applied as part of HCI methods and not within the OSS development process. On the other hand, the OSS development process focuses on source code and thus on the development of functionalities. The OSS development process has a number of features (like functionality-focused development) which prevent many of the HCI usability techniques from being adopted directly [7]. This community has now started to adopt some usability techniques. Most of the techniques taken on board by the community are for evaluating usability [7], whereas it has not adopted many techniques related to requirements analysis and design. Some techniques have been adapted ad hoc for adoption in OSS development projects [7]. Only a few research papers have reported the use of the HTA technique in OSS developments [8]. The HTA technique was applied in the Multiplex medical tool as described in Doesburg [8]. HTAs were used to decompose and compare various tasks related to the intravenous medication process. These HTAs were useful in comparing how various tasks are performed with both the current infusion system and the proposed control system. Participants were presented with five patient cases, designed with the help of an experienced ICU nurse (Intensive Care Unit). The cases were carefully created so that the decisions made by the participants do not influence the execution of the tasks. The tasks would be carried out both in the simulated environment and in Multiplex, to then contrast the results. Clicks, task time, errors and application response time were measured. The tests showed that users had difficulty understanding the buttons, which increased the probability of errors in the doses of intravenous therapies.

This paper addresses the research problem of how to adopt the HTA usability technique within the OSS development process, and particularly within a real OSS project called OpenOffice Writer. To do this, we previously identified which problems had to be solved in order to be able to apply the technique. Some authors claim that the main reasons for the generally poor usability of OSS developments are that OSS developers have tended to develop software for themselves [9] and that the development community is uninformed about who its users are [10]. HTA involves three linked stages: information gathering, diagramming, and analysis [11]. For the collection of information, existing information is reviewed (for example, operating manuals, procedures, etc.) to establish how things are done, what information is needed and whether or not the task is performed satisfactorily. To carry out the diagramming, hierarchical trees are used to facilitate the analysis of the tasks. The HTA technique was developed by Annette and Duncan [12]. HTA is the oldest and best-known task analysis technique, which is still valid, although there are new ways to apply the HTA technique [13, 14]. The main purpose of HTA is to understand how a system works and whether or not it achieves its objectives. This provides a functional analysis instead of a behavioral description. The difference between HTA and other task analysis techniques (e.g., GOMS and Object-Action Interface Model) is that in the other techniques a list of activities with cognitive aspects is made whereas in HTA the goals of the task are identified. On this ground, we have selected the HTA usability technique for adoption in the OpenOffice project.

This paper is organized as follows. Section 2 describes the research method followed to apply the usability technique. Section 3 describes the proposed solution. Section 4 discusses the results. Finally, Sect. 5 outlines the conclusions and future research.

2 Research Method

In order to validate our proposal for adopting the HTA technique in OSS development projects (particularly in OpenOffice Writer), we had to volunteer for this project. This is equivalent to being members of the OSS community of volunteers. We used a case study as the qualitative research method to validate our research [15]. We use a non-experimental design, since we do not randomly assign subjects or control the groups. From a case study, we learn about the experiences of applying usability techniques adapted to OSS projects. The case study is the best research method for carrying out this validation. On this ground, we followed the guidelines set out by Runeson et al. [15]. This research method is used when the phenomenon under study (in this case, the adoption of techniques with adaptations) is studied within its real-world context (in this case, OSS projects).

3 Proposed Solution

In this section, we briefly describe the HTA usability technique applied in an OSS Project. First, we specify the characteristics of the selected OSS project (OpenOffice Writer). Second, we describe the HTA technique as prescribed by HCI, followed by the details of the changes made to this technique for application to the OSS project. Finally, we report the results of applying the HTA technique.

3.1 Case Study Design

The case study is one of the most popular forms of qualitative empirical research. A case study investigates the phenomenon of interest in its real-world context. To be exact, the phenomenon of interest for this research is the adoption of the HTA technique with adaptations, whereas the real-world context is an OSS project. We will give a general description of the procedure enacted to perform the case study. Our case study is based on the research question: Is it possible to determine whether some adaptations of the HTA usability technique would enable its adoption in a real OSS project?

OpenOffice Writer is the selected OSS project in which the HTA technique is to be adopted. OpenOffice is currently one of the most popular OSS projects and a model of a successful OSS project. It is a large-scale, well-organized and structured OSS project, which also has a large user community.

3.2 Changes to the HTA Usability Technique

On the one hand, Annett [16] acknowledges that the HTA is based on copying how tasks are routinely performed, making it appropriate for task computerization projects (e.g., accounting records, product industrialization, etc.). On the other hand, Shepherd considers HTA as a strategy to examine tasks, focused on refining the performance criteria, concentrating on the skills of the members, understanding the contexts of the tasks and generating useful hypotheses to overcome performance problems [17]. Therefore, HTA is a very detailed study of a group of users that allows us to understand the current system and the information flows in it [18]. The HTA technique, belonging to the HCI Interaction Design activity, is an iterative process of identification and decomposition of tasks into sub-tasks, together with the precision of such decomposition [19]. This technique is very useful to structure the observed information about how the user organizes the tasks that usually carries out in his work. Therefore, the use of this technique can complement the Eduction and Requirements Analysis efforts when it comes to a system that aims to support the user in carrying out their common tasks. Also, this technique serves to model how users organize their activities and what means they use to carry them out [18].

Annett and Duncan [16] and other publications [13, 14, 19, 20] propose procedures to apply the HTA technique. Although these procedures are very similar to each other, for processes focused on user-centered development, a suitable approach is that of the work of Preece et al. [19]. For this reason, we have used the version of Preece et al. [19] as a reference, since it is the simplest among the authors studied to make the respective adaptations.

To adapt the HTA technique, it must first be formalized. Subsequently, modifications must be made to the formalized technique to be able to incorporate it to OSS development. It should be clarified that the formalization of the HTA technique is carried out in this work, since the proposed procedure is not explicitly defined by Preece et al. [19]. Below, the first five steps of the HTA technique by Preece et al. [19] are described and the unfavorable or inconvenient conditions that hinder their incorporation into OSS development are detailed.

The first step is the specification of the main work or task area, and it aims to recognize the main functions of the software tool to determine whether a failure may be occurring or not. Preece et al. [19] do not establish how to obtain the information to establish which are the main tasks. To overcome this unfavorable condition, we propose that this information be obtained by other means than those established by the HCI (for example, forums and surveys). The second step of this technique is to break down the main task into subtasks, each subtask corresponding to an action that needs to be performed in order to complete the task. Preece et al. [19] suggest breaking down each main task into four to eight subtasks specified in terms of objectives that cover the entire area of interest of the selected software project. Tasks should be written in such a way that they are easily recognized by the user. The output product obtained in this step is the document "list of tasks and subtasks". However, Preece et al. [19] do not specify the format of the document associated with this step. The third step corresponds to including the plans in the list of tasks and sub-tasks.

The plans describe under what conditions the users will perform the sub-tasks. Also, the plans control the order for the execution of the tasks. Preece et al. [19] do not specify the format of the document associated with this step.

The fourth step is known as drawing the subtasks, which consists of drawing the sub-tasks in an outline. Preece et al. [19] do not concisely suggest the type of schema to use to draw these sub-tasks. The output product obtained in this step is the hierarchical tree of tasks. This step does not explicitly identify the associated tasks for drawing these schematics, nor does it specify the format of this product associated with this step. The fifth step is to decide the level of detail that the decomposition of tasks requires and at what point to stop since in this step a consistent treatment of the situation is ensured. In this fifth step, Preece et al. [19] propose that the decomposition of tasks continues until the information flows were more easily represented from a much lower level of description to a higher level. In this step, it has been identified as a disadvantage that developers must come together to validate the level of detail defined in the decomposition of tasks obtained in the previous step. In other words, the developers must be physically together, a condition that cannot occur due to the characteristics of the OSS projects. Therefore, a modification to the technique must be made. To resolve this unfavorable condition, it is proposed to request feedback from the project developers via email.

Table 1 summarizes for each step of the technique the unfavorable conditions analyzed and the main adaptations proposed. There are mainly three adaptations: (i) Users participate online, (ii) it is necessary to obtain certain information to apply the technique by means other than those prescribed by the HCI and (iii) that the expert can be replaced by a developer or expert user of the OSS project. Particularly, in our case the expert was replaced by an HCI student under the supervision of two expert usability researchers. For this reason, there is no risk that the quality of the software will be negatively affected when applying the adaptation proposed for the HTA technique.

3.3 OpenOffice Writer Case Study Results

In this study, steps from the original HTA technique have been added to facilitate its application in OSS projects. Next, for each step in Table 2, the tasks carried out in the adaptation of the proposed HTA technique are detailed. By defining each task, new unfavorable conditions arise, and new adaptations are proposed to incorporate this technique into OSS projects.

To test the feasibility of our adaptation of the HTA technique, it was necessary to apply it in OpenOffice Writer. This tool is a cross-platform word processor that is part of the suite of applications of the Apache OpenOffice office suite. For the first task (Specifying the main work area or tasks), an analysis was made of the information obtained through means other than those established by the HCI (such as forums, emails and surveys). This obtained information refers to: (i) the problems reported in the OpenOffice Writer online forum, (ii) the problems reported by email and that are reported in the OpenOffice subscription lists and (iii) the results obtained in the survey applied for the Personas technique [21]. The Personas technique was applied in the first phase of the research to obtain the user profile of the OpenOffice Writer tool. In this step, we have specifically proceeded to extract and process the information from the following data sources: the “Personas” online survey,

Table 1. Summary of the identified adverse conditions and the proposed adaptations for the HTA technique.

Technique steps [19]	Adverse conditions	Proposed adaptations
1. Specify the area of work or main task	• It is necessary to obtain information (e.g., commonly recurring problems) to apply the technique in the way prescribed by HCI	• The information necessary to apply the technique is obtained through other means different to those established by HCI (e.g., through forums)
2. Break down tasks into subtasks	• The tasks associated to these steps are not explicitly identified	• The tasks associated to each step are detailed • The format for the output product is specified
3. Include the plans to carry out the subtasks		
4. Draw the subtasks		
5. Deciding the level of detail that the decomposition of the tasks requires	• The face-to-face participation of the developers is necessary • The tasks associated to these steps are not explicitly identified	• Ask developers for feedback through email • The tasks associated to each step are detailed
6. Deciding the level of depth and amplitude of the decomposition of tasks		
7. Assigning numbering to tasks		
8. Check the task analysis	• The tasks associated to these steps are not explicitly identified • It is indispensable to employ a usability expert • The format of the document associated to these steps is not specified	• The tasks associated to each step are detailed • The expert can be a developer, an expert user of the OSS Project or an HCI student (under the supervision of a mentor) • The format for the output product is specified
9. Present the task analysis in written format		

the OpenOffice online forum, the Bugzilla repository, the emails received in Outlook from the subscription lists (users@openoffice.apache.org, dev@openoffice.apache.org, issues@openoffice.apache. org). Messages received from subscription lists were filtered by words that may be related to the OpenOffice Writer tool.

Regarding the online forum, its members' task is to support other users in their learning or solving specific problems with the OpenOffice Writer application. In addition, information was obtained on the specific page where errors are reported (bugs) of the OpenOffice tool called Bugzilla. Within Bugzilla, the Writer product and the messages with the User Interface (UI) component were examined, because this component is related to User Interface problems and most likely with usability issues related to user interaction with the application.

Table 2. Steps and tasks of the HTA technique adapted to apply it in an OSS project

Steps of the adapted HTA technique	Tasks
1. Specifying the main work area or tasks	• Define the main tasks to be analyzed • Specify names for the main tasks
2. Redrawing the hierarchical task tree	• Break down tasks into subtasks • List each task and subtask in decomposition
3. Drawing the subtasks	• Diagram the hierarchical tasks tree • Include plans that guide task development
4. Defining the level of detail for the decomposition of tasks	• Define the limit of the system description
5. Deciding the level of depth and amplitude	• Decide the stop condition for task decomposition • Decide depth and breadth level for decomposing tasks
6. Checking the task analysis	• Review task analysis
7. Requesting feedback for the task analysis	• Request feedback to improve task analysis
8. Selecting the users that are to participate in the application of the technique	• Act with the users who confirmed their participation in the research through their email contribution
9. Designing a format for data collection	• Design format for data recording
10. Executing the HTA of the defined tasks	• Executing the HTA of defined tasks with real users
11. Analyzing and interpreting the data obtained in the execution of tasks	• Analysis of the information provided by users in the development of tasks from the previous step
12. Redrawing the hierarchical task tree	• Perform the hierarchical tree diagram with the results obtained in the previous step by collaborative work • Include plans for the new hierarchical tree
13. Submit task analysis report	• Make the report with the pre-analysis and post-analysis of tasks • Present the report with pre-analysis and post-analysis of tasks

To extract the information from these two data sources (online forum and Bugzilla), we opted to use the Python tool. This tool is very easy to use and has a fast learning curve, which along with its versatility make it a high-quality language for data analysis. This programming language was selected to easily implement a source code using libraries that allow data mining. Once the information was extracted from these two data sources, the data analysis was carried out with the R tool to determine the patterns of problems or difficulties that users have with OpenOffice Writer. Table 3 shows some of the categories of problems identified in the online forum, emails and the Bugzilla website, as well as the difficulties registered in the online survey "Personas".

Table 3. Main problems reported in the forums, emails, Bugzilla and difficulties expressed in the "Personas" survey.

Category reported in	Forum	Emails	"Personas" survey	Bugzilla
Insert heading/footer	X			
Insertion of images	X		X	
Use of capital letters	X			
Use of tables	X	X	X	X
Use of the contextual menu to apply fonts	X			
Design formulas	X			

Based on this list of problems, we have defined five main tasks related to the purpose of our analysis and that are performed by users of the OpenOffice Writer tool. These tasks are to: (i) Write a document with capital letters, (ii) write a story that includes images, (iii) design a table with formulas, (iv) improve the design of a document, and (v) write a help manual.

For step two (breaking down the main task into subtasks), once the main tasks to be executed by OpenOffice Writer users have been defined, we break them down into subtasks considering them to have a minimum of six and a maximum of eight subtasks. As a result, we have the document "List of Tasks and Subtasks of task 1 Write a document with capital letter". For this step, it is proposed to use a conventional numerical system that emphasizes the hierarchy of tasks (for example: 1, 2, 3 for the first level, 1.1, 1.2, 1.3 for the second level, etc.).

Table 4 shows a fragment of the document that lists the tasks and subtasks in a numerical scheme specifying the type of Action (sequence, decision or iteration). Column 2 shows the type of action to be taken ((A)ction, (D)ecision, (I)teration).

For step three (drawing the subtasks), we draw the tasks and subtasks in the form of a tree diagram to have an overview of the tasks. The diagrams have been produced with the online tool CACOO 2.0, which provides a manipulable means for exploring information and increases cognitive resources, since it becomes a visual resource to expand human working memory. Each task is drawn with a box and the tasks that cannot be divided in turn are underlined (one line below the box). Figure 1 presents a fragment of the hierarchical tree designed with the collaborative tool CACOO 2.0 and is related to the main Task 1 (Writing a document with capital letters). The same was done for the rest of the tasks.

In the fourth step (defining the level of detail for the decomposition of tasks), the plans to control the sequence in which these subtasks are performed must be considered, and they show the conditions that indicate when the subtasks are applicable. The purpose of these plans is to exclude impossible scenarios and establish conditions for what can

Table 4. Fragment of the “Writing a document with capital letters” task

GOAL: To write a document in 2 columns, which includes capital letters and a page header	Type of action
1. Open the application OpenOffice Writer	A
2. Write the title of the document: “The core of the processor”	A
3. Apply two formats to the title: centered and bold	A
4. Type the following text into the document: Processor manufacturers are very clear; they want to create increasingly powerful PCs. To this end, taking advantage of improvements in the manufacturing process, they add the greater number of cores possible to the processor	A
5. Divide the document into two journalistic columns	A

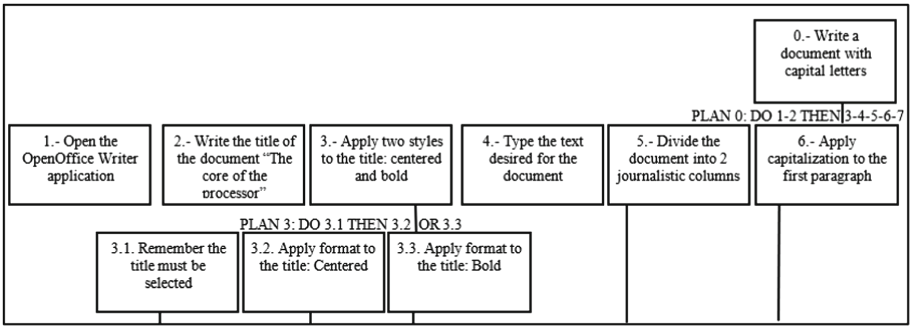
0.- Write a document with capital letters
PLAN 0: DO 1-2 THEN 3-4-5-6-7
1.- Open the OpenOffice Writer application
2.- Write the title of the document “The core of the processor”
3.- Apply two styles to the title: centered and bold
4.- Type the text desired for the document
5.- Divide the document into 2 journalistic columns
6.- Apply capitalization to the first paragraph
PLAN 3: DO 3.1 THEN 3.2 OR 3.3
3.1. Remember the title must be selected
3.2. Apply format to the title: Centered
3.3. Apply format to the title: Bold

Fig. 1. Hierarchical tree of the “Writing a document with capital letters” task 1.

be prevented and what can happen. For example, in Task 1 (Writing a document with capital letters), PLAN 3: DO 3.1 THEN DECIDE BETWEEN 3.2 or 3.3, indicates that we must first run subtask 3.1 and then decide between running subtask 3.2 or subtask 3.3. Creating this detailed analysis of a main task in other simpler tasks has taken us a long time, in order to make each step explicit and making it less likely that the user will ignore any knowledge they require. At the same time, other opportunities have been identified to improve the user experience. For example, knowing that applying a capital letter style is a difficult option for a user to locate can influence the design and implementation of new functionality in the OpenOffice Writer tool. Developers should consider increasing the visibility of this option and making it easy for users to locate.

For the fifth step (deciding level of depth and amplitude), the detailed decomposition is performed according to the PxC rule that calculates the efficiency of continuing the analysis based on the probability of failure (P) x the cost of failure (C). In other words, the decomposition of the tasks that requires additional effort to analyze is only carried out when there are reasons to consider that the current performance is not acceptable. For example, in objective 7 (Apply capital letter) of Task 1 (Write a document with capital letters) (Table 3), we have considered this activity to be very important and difficult to execute (P = high and C = high), so all the additional effort of the analysis is concentrated here. In this case, developers have not adequately understood what tasks are essential and how the system should support users in performing them. Once we have defined the level of detail and the two dimensions (depth and breadth) we consult the developers via email if the decomposition of the tasks carried out in the previous step matches the conceptual and functional model that they expected to have on the application. Our hierarchical task analysis will allow any developer to quickly understand what their application does and how its capabilities translate into the system user experience.

In the sixth step (checking the task analysis), an HCI student under the tutelage of a mentor acts as an expert to perform the hierarchical task analysis check. Once the task analysis has been checked, it can serve as a system documentation, so developers can quickly understand how users interact with their software system. It is important to mention that the developers of the OpenOffice project are very aware that there is a need to look for new methods and practices of the HCI to improve the usability of this project. Therefore, hierarchical task analysis is an effective means of raising awareness among developers about reconsidering user engagement for software enhancement and being willing to listen to end users for that purpose.

Regarding the seventh step (requesting feedback for the task analysis), to reinforce the check of an initial version of the task analysis, feedback is requested from the project developers via email. An improved version of task analysis allows developers to explore various approaches to complete the same task or help optimize particular interactions between a user and a software system.

In the eighth step (selecting the users that are to participate in the application of the technique), the participation of representative users is necessary to discuss current or future tasks, but it is impossible to obtain a group of users that we can consider representative because they do not have enough time. Consequently, we dialogued with a group of volunteer users to collaborate with us in the application of the HTA technique. To do this, we made contact via email with some of the OpenOffice Writer users whom we knew were willing to participate in our research. Finally, we decided to invite an OpenOffice Writer user who was not involved in task decomposition to check the consistency of the task analysis.

In step nine (designing a format for data collection), once the diagrams of the tasks to be executed by the OpenOffice Writer user were produced, they were analyzed to determine possible human errors, especially those related to procedures or usability problems derived from the current design of the software product. This analysis is recorded in the document "Task Analysis Table". Table 5 contains the format of the document "Task Analysis Table" and is made up of four columns: the first column indicates the *action* performed by the user, followed by the *cause* and *effect* columns observed during the execution of the action and finally the *redesign* column where a possible task improvement is provided.

For the execution of step ten (executing the HTA of the defined tasks) the physical presence of users meeting with the evaluator or evaluators is necessary. This is impossible because the users of the OSS communities are spread out across the world. Therefore, a remote observing session is established to have an appropriate perspective with a guest user. It is important to mention that evaluators may need to contact users as many times

Table 5. Format of the document “Task Analysis Table”

TASK 1			
USER NUM: START TIME: END TIME: TASK DURATION:			
ACTION	CAUSE	EFFECT	REDESIGN

as necessary to clarify all the questions that are necessary. However, because the work in the OSS community is done by volunteers in their spare time, it is difficult to apply several iterations to achieve greater precision in task analysis. In the execution of the HTA technique, it was agreed with the participant to carry out a remote observation on a specific day and time, all communication had to be done electronically, including explanations, supply of materials and data collection. Specifically, two tools were used: Skype, to be able to talk to the user and see their reactions, and TeamViewer, to visualize their interaction with the application by remotely accessing their screen view. Once the HTA was completed, an HCI student under the tutelage of a mentor acts as an expert to analyze and interpret the data obtained, which will finally allow the tasks to be redesigned and organized appropriately within the software system. Also, developers can be suggested to include new functions within the system and the user interface.

In step eleven (analyzing and interpreting the data obtained in the execution of tasks), the task analysis process is not only based on the registration of existing subtasks, but also provides possible improvements resulting from the introduction of new facilities for the development of a task. The implications of potential changes to these tasks should be re-recorded in the hierarchical task trees. Here the advantage of working with the CACOO platform is that it allows simultaneous collaborative work, that is, the same diagram can be edited at the same time by more than one person. This is related to step twelve (redrawing the hierarchical task tree). Therefore, the redesign of the hierarchical task trees was done jointly between the researchers and the developers of the OpenOffice Writer application. Finally, in step thirteen, to present the task analysis report, we designed a table that will document the details of the specific tasks, the details of the interactions between the user and the current system, as well as any problems related to these tasks. Table 6 presents a fragment of the document “HTA Report”, which corresponds to the final product of the application of the HTA technique.

4 Discussion of Results

Communication with the OpenOffice community was troublesome because not all the users were willing to participate in the application of the usability technique. However, our experience of participating in large projects (e.g., LibreOffice Writer and OpenOffice Writer) has revealed that it is very difficult to recruit real end users to participate in the application of usability techniques in OSS projects generally [21, 22]. During the application of the HTA technique to the OpenOffice Writer project, the key problem

Table 6. Fragment of the document "HTA Report"

Higher-order task	Plan	Subtasks	Information flow through the interface	Information the user already knows	Pre-analysis notes	Post-analysis notes	Improvement in the user interface
Writing a document with capital letters	PLAN 0: DO 1-2, then 3-4-5-6-7-8	1. Open the OpenOffice Writer app	Start the OpenOffice Writer processor				
		2. Write the title "The core of the processor"	Type the title of the document				
		3. Apply formats to the title: centered and bold	Apply formats: centered and bold	Use of shortcuts			
		4. Type the text that will be contained in the document	Type text	The user removes centered and bold and applies bold without specifying it as an action			
		5. Dividing the document into 2 journalistic columns	Format>Columns>2	Use of shortcuts			
		6. Apply capital lettering to the first paragraph of the first column	Format>Paragraph>Initials>Show Initials		The user does not easily locate the "Capital Letters" option	The user, confused, reviews all the tabs of the "Paragraph" option to find the "Capital Letters" option	Directly show the Capital Letters option in the Format menu

was user availability, as many are volunteers and had very little spare time. This is preliminary research. In the HTA technique, the main adaptations made were three First, users participate online through web artifacts (forums). In the case of the HTA technique, the use of the forum allowed a reliable analysis of the tasks for developers to make decisions regarding improving the interface design. Second, the usability expert is replaced by a developer, expert user, or HCI student under the tutelage of a mentor. Particularly, in our case the expert was replaced by a team of junior experts supervised by a main expert. Third, it is necessary to obtain certain information to apply the technique in the manner prescribed by HCI. This information can be models or graphic representations of the tasks performed by users, becoming low-cost alternatives to achieve the same objectives that the HCI prescribes.

The HTA technique must be adapted and applied with the participation of users, for a better understanding of the tasks to be carried out and that is more understandable for the components of the team that are not accustomed to the notations typically of Software Engineering. Therefore, more cases studies are required to validate the proposed adaptations. Note that there are other usability techniques (for example, user profiles, heuristic evaluation) that might benefit from the proposed adaptations (for example, HCI students supervised by a mentor standing in for experts) to enhance technique adoption in the OSS development process.

5 Conclusions

The aim of this research was to evaluate the feasibility of adopting adapted HCI usability techniques in OSS projects. To be precise, we adapted the HTA technique for application in the OpenOffice Writer project. It was by no means easy to find users to volunteer to apply the HTA technique. As mentioned, users generally have very little time, and it was hard to get them to participate without an incentive. We identified three main adverse conditions that are barriers to the application of HTA in OSS development projects: (i) the need to have a usability expert to apply the technique, (ii) the unavailability of on-site users and (iii) the need to obtain certain information to apply the technique in the manner prescribed by HCI. In order to surmount these barriers, (i) a HCI student or group of HCI students supervised by a mentor substitutes the usability expert, (ii) OSS users participate remotely and (iii) the necessary information is obtained by means other than those established by the HCI (for example, in forums). We believe that it is necessary to educate users and OSS community members generally to raise awareness of the importance of application usability and publicize existing usability techniques in order to encourage participation. As future research, we intend to conduct further case studies to adapt and apply more usability techniques in OSS projects in order to validate the proposed adaptations and study new web artefacts that can be adapted to OSS communities to improve communication. This pilot study will be expanded by adapting and applying the HTA technique in the Libre OpenOffice project to confirm our findings.

Acknowledgements. Work funded by the Secretariat of Higher Education, Science, Technology and Innovation (SENESCYT) of the Government of Ecuador as part of an academic scholarship granted for postgraduate training, and Quevedo State Technical University through doctoral scholarships for university professors. Also, this research was funded by the FEDER/Spanish Ministry of Science and Innovation – Research State Agency: project MASSIVE, RTI2018-095255-B-I00, the R&D programme of Madrid (project FORTE, P2018/TCS-4314), and project PGC2018-097265-B-I00, also funded by: FEDER/Spanish Ministry of Science and Innovation – Research State Agency. Finally, this research received funding from the University of Atacama "DIUDA 22316" project.

References

1. Schryen, G., Kadura, R.: Open source vs. closed source software. In: 2009 ACM Symposium on Applied Computing, SAC 2009, pp. 2016–2023 (2009). https://doi.org/10.1145/1529282.1529731
2. Noll, J., Liu, W.-M.: Requirements elicitation in open source software development: a case study. In: 3rd International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, FLOSS 2010, pp. 35–40 (2010). https://doi.org/10.1145/1833272.1833279
3. Smith, S., Engen, D., Mankoski, A., Frishberg, N., Pedersen, N., Benson, C.: GNOME Usability Study Report. Technical Report, Sun Microsystems (2001)
4. Çetin, G., Gokturk, M.: A measurement based framework for assessment of usability-centricness of open source software projects. In: 4th International Conference on Signal Image Technology and Internet Based Systems, SITIS 2008, November 2008, pp. 585–592 (2008). https://doi.org/10.1109/sitis.2008.106
5. Raza, A., Capretz, L.F., Ahmed, F.: Users' perception of open source usability: an empirical study. Eng. Comput. **28**(2), 109–121 (2012). https://doi.org/10.1007/s00366-011-0222-1
6. Raza, A., Capretz, L.F., Ahmed, F.: An empirical study of open source software usability: the industrial perspective. Int. J. Open Source Softw. Process. **3**(1), 1–16 (2011). https://doi.org/10.4018/jossp.2011010101
7. Castro, J.W.: Incorporación de la Usabilidad en el Proceso de Desarrollo Open Source Software. Tesis Doctoral. Departamento de Ingeniería Informática. Escuela Politécnica Superior. Universidad Autónoma de Madrid (2014)
8. Doesburg, F.: Developing a System for Automated Control of Multiple Infusion Pumps, July 2012
9. Raza, A., Capretz, L.F., Ahmed, F.: An open source usability maturity model (OS-UMM). J. Comput. Hum. Behav. **28**(4), 1109–1121 (2012)
10. Benson, C., Müller-Prove, M., Mzourek, J.: Professional usability in open source projects: GNOME, OpenOffice.org, NetBeans. In: CHI 2004, Extended Abstract on Human factors in Computing System, CHI EA 2004, April 2004, pp. 1083–1084 (2004). https://doi.org/10.1145/985921.985991
11. Granollers, T.: MPIu+a Una metodología que integra la Ingeniería del Software, la Interacción Persona Ordenador y la accesibilidad en el contexto de equipos de desarrollo mutidisciplinares. Tesis Doctoral. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Lleida (2004)
12. Gray, M.J., Annett, J., Duncan, K.D., Stammers, R.B.: Task analysis. In: Department of Employment Training Information Paper 6. HMSO (1971)
13. Paternó, F.: Model-Based Design and Evaluation of Interactive Applications, Pisa, Italy (2000)
14. Stary, C., van der Veer, G.C.: Task analysis meets prototyping: seeking seamless UI-development. In: Extended Abstracts on Human Factors in Computing Systems, CHI 1999, pp. 104–105 (1999)

15. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. J. Empir. Softw. Eng. **14**(2), 131–164 (2009). https://doi.org/10.1007/s10664-008-9102-8
16. Annett, J., Duncan, K.D.: Task analysis and training design. Occup. Psychol. **41**, 211–221 (1967)
17. Shepherd, A.: HTA as a framework for task analysis. Ergonomics **41**(11), 1537–1552 (1998)
18. Ferré, X.: Marco de Integración de la Usabilidad en el Proceso de Desarrollo Software. Tesis Doctoral. Facultad de Informática. Universidad Politécnica de Madrid (2005)
19. Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., Carey, T.: Human-Computer Interaction, 1st edn. Addison-Wesley Pub. Co. (1994)
20. Stanton, N.A.: Hierarchical task analysis: developments, applications, and extensions. Appl. Ergon. **37**(1), 55–79 (2006). https://doi.org/10.1016/j.apergo.2005.06.003
21. Llerena, L., Rodriguez, N., Castro, J.W., Acuña, S.T.: Adapting usability techniques for application in open source Software: a multiple case study. Inf. Softw. Technol. **107**, 48–64 (2019). https://doi.org/10.1016/j.infsof.2018.10.011
22. Llerena, L., Castro, J.W., Acuña, S.T.: A pilot empirical study of applying a usability technique in an open source software project. Inf. Softw. Technol. **106**, 122–125 (2019). https://doi.org/10.1016/j.infsof.2018.09.007