



Universidad Autónoma  
de Madrid

**Biblos-e Archivo**  
Repositorio Institucional UAM

**Repositorio Institucional de la Universidad Autónoma de Madrid**

<https://repositorio.uam.es>

Esta es la **versión de autor** del artículo publicado en:  
This is an **author produced version** of a paper published in:

Information and Software Technology 107 (2019): 48 – 64

**DOI:** <https://doi.org/10.1016/j.infsof.2018.10.011>

**Copyright:** © 2018 Elsevier B.V.

El acceso a la versión del editor puede requerir la suscripción del recurso

Access to the published version may require subscription

# Adapting Usability Techniques for Application in Open Source Software: A Multiple Case Study

Lucrecia Llerena<sup>1</sup>, Nancy Rodriguez<sup>1</sup>, John W. Castro<sup>2</sup>, Silvia T. Acuña<sup>1</sup>

<sup>1</sup>Departamento de Ingeniería Informática, Universidad Autónoma de Madrid,  
Calle Francisco Tomás y Valiente 11, 28049 Madrid, Spain

<sup>2</sup>Departamento de Ingeniería Informática y Ciencias de la Computación, Universidad de Atacama,  
Avenida Copayapu 485, 1530000 Copiapó, Chile

<sup>1</sup> lucrecia.llerena@estudiante.uam.es, <sup>1</sup> nancy.rodriguez@estudiante.uam.es, <sup>2</sup> john.castro@uda.cl,  
<sup>1</sup> silvia.acunna@uam.es

## ABSTRACT

**Context:** As a result of the growth of non-developer users of OSS applications, usability has over the last ten years begun to attract the interest of the open source software (OSS) community. The OSS community has some special characteristics (such as worldwide geographical distribution of both users and developers and missing resources) which are an obstacle to the direct adoption of many usability techniques as specified in the human-computer interaction (HCI) field.

**Objective:** The aim of this research is to adapt and evaluate the feasibility of applying four usability techniques: user profiles, personas, direct observation and post-test information to four OSS projects from the viewpoint of the development team.

**Method:** The applied research method was a multiple case study of the following OSS projects: Quite Universal Circuit Simulator, PSeInt, FreeMind and OpenOffice Writer.

**Results:** We formalized the application procedure of each of the adapted usability techniques. We found that either there were no procedures for adopting usability techniques in OSS or they were not fully systematized. Additionally, we identified the adverse conditions that are an obstacle to their adoption in OSS and propose the special adaptations required to overcome the obstacles. To avoid some of the adverse conditions, we created web artefacts (online survey, wiki and forum) that are very popular in the OSS field.

**Conclusion:** It is necessary to adapt usability techniques for application in OSS projects considering their idiosyncrasy. Additionally, we found that there are obstacles (for example, number of participant users, biased information provided by developers) to the application of the techniques. Despite these obstacles, it is feasible to apply the adapted techniques in OSS projects.

**Keywords:** Open Source Software, Usability Techniques, User Profiles, Personas, Direct Observation, Post-Test Information.

## 1. Introduction

Open source software (OSS) has spread so swiftly that it now rivals commercial software systems in terms of deployment [1]. Some OSS communities nowadays do not have processes in place to guarantee that, taking into account the features of this community as a whole, the developed software is good [2]. Shortcomings with respect to process, activity, task and technique definition in the field of OSS development has led researchers from different fields to take an interest in this field of research and try to remedy the failings [3–5].

Usability is one of the key software development quality attributes [6]. In recent years, OSS has come to be an important part of computing [7–18]. However, several authors have acknowledged that the usability of OSS is poor [19–21]. In the empirical study conducted by Raza et al. [22], 60% of respondents (non-developer users) claimed that poor usability is the main obstacle to be overcome by OSS applications in order to encourage users to migrate from commercial software [21]. The main reasons for the generally poor usability of OSS developments are: OSS developers have tended to develop software for themselves [4,23], the developer community is very much in the dark about who its users are [19,24], OSS communities operate according to a meritocracy based on software code input, and, due to the unavailability of resources, testing and bug reporting are almost exclusively conducted by volunteers and end users [7].

On one hand, the human-computer interaction (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 feature development. The OSS development process has special characteristics (e.g., the members of the community are geographically distributed, resources are in short supply and the culture may be quite

alien to interaction designers) [7] [20] [22]. This prevents many of the HCI usability techniques from being adopted directly [25].

The OSS community has now started to adopt some usability techniques, mostly usability evaluation techniques [25]. The OSS community has not adopted many techniques related to requirements analysis. Some usability techniques have been adapted ad hoc for adoption in OSS development projects [25]. The adoption of usability appears to be less straightforward in the OSS development process than in commercial development due to some of the characteristics of the OSS community, like: (i) its feature-centred development, (ii) worldwide geographical distribution, (iii) shortage of resources, (iv) a culture that may be alien to interaction designers [7] [20] [22]. Consequently, usability technique adoption is a demanding task because most HCI techniques are not designed for the type of environment in which OSS is developed [25]. Our research aims to determine how to adopt a set of usability techniques in the OSS development process. To do this, we analyse and identify which obstacles have to be solved to be able to apply these techniques in OSS projects. Our research work addressed two areas: SE and HCI. With the aim of bringing these two areas together, we use usability techniques as a bridge in order to adopt knowledge of the HCI area in the SE area, specifically in the OSS development process [25].

This research has two aims. First, we intend to adapt four usability techniques (two requirements engineering techniques and two evaluation techniques) for adoption in the OSS development process. The selected techniques related to requirements engineering were: personas and user profiles to conduct user analysis. The evaluation-related techniques were: post-test information and direct observation to evaluate the usability of installed systems. For our research, we selected the first two techniques (user profiles, personas) because they should improve and enrich the requirements analysis activity and the last two (post-test information and direct observation) because they are useful for improving usability in response to errors identified in software evaluation. Second, we set out to determine the feasibility of adopting these four usability techniques adapted to four real OSS projects. In the following, we describe the criteria used to select the OSS projects. On one hand, Quite Universal Circuit Simulator (QUCS)<sup>1</sup> and PSeInt<sup>2</sup> were chosen considering that they were not very ambitious software development projects with a low level of coding. Additionally, they were at very early stages of the development when user segments have not yet been defined. On the other hand, FreeMind<sup>3</sup> and OpenOffice Writer<sup>4</sup> were picked because they are projects that have organized and structured user communities, which is beneficial for software usability evaluation. The research method that we used to validate the feasibility of our proposal for adopting usability techniques adapted to OSS projects is a multiple case study [26]. This method is useful for extending the information on the phenomenon under study (adoption of adapted usability techniques) and determining whether or not the results are consistent, achieving sounder conclusions with a real setting (OSS projects). Consequently, we had to volunteer for OSS projects and join the respective OSS communities.

This paper is organized as follows. Section 2 introduces related work. Section 3 reports the research method followed to apply the usability techniques in OSS projects. Section 4 describes the identified adverse conditions and the adaptations made to usability techniques. Section 5 reports the multiple case study design and planning. Section 6 reports the results of multiple case study. Section 7 discusses the results. Section 8 describes the study validity threats. Finally, Section 9 outlines the conclusions and future work.

## 2. Related Work

In recent years, the worldwide OSS community has adopted just over 50% of evaluation-related HCI techniques. However, only about 20% of usability techniques related to requirements engineering and design activities have been adopted [25]. Therefore, more research is required to support the adoption of techniques related to requirements engineering in OSS developments. In view of the importance of HCI and SE, it is only logical to study the user-centred software development activities in OSS projects. This is especially true of the requirements engineering stage, because the discovery of user requirements during the early development activities is useful for putting right any software defects detected later on [27].

---

<sup>1</sup> <https://sourceforge.net/projects/qucs/?source=directory>

<sup>2</sup> <http://pseint.sourceforge.net>

<sup>3</sup> <https://sourceforge.net/projects/freemind/?source=directory>

<sup>4</sup> <https://sourceforge.net/projects/openofficeorg.mirror/>

There are papers in the literature reporting the usability evaluation of some OSS applications [14,28,29]. Assa et al. [14] studied the usability issues facing software developers using code analysers by evaluating one of the popular open-source static-code analysis tools. Al-Odan and Al-Daraiseh [28] conducted a thorough study, placing five of the most popular free and open source software tools side by side for comparison with respect to both user acceptance and technical specifications. Ternauciuc and Vasiiu [29] tried to inventory existing methods for testing and improving usability, with a particular focus on e-learning platforms. However, usability technique definition and integration into OSS projects is a complicated process, which has not been researched at length [20,27,30–32]. Existing papers suggest that usability techniques should be reconceptualized. However, they do not explain how the OSS community should go about adaptation. Nichols and Twidale [4] and Ternauciuc and Vasiiu [29] are the only authors to put forward some general ideas for improving usability. However, the issues to be taken into account in order to adopt such techniques in OSS developments are unclear.

On the other hand, Castro [25] proposes a framework for integrating usability techniques into OSS developments. This framework is composed of a number of adaptations in response to the adverse conditions for adopting usability techniques in OSS development projects. In order to adopt usability techniques in OSS development projects, it is necessary to: (i) study the adverse conditions preventing the use of HCI techniques, and (ii) analyse what types of and which adaptations are necessary if these techniques are to be used in OSS projects [25]. The adverse conditions are classified into three major groups (families of adaptations). First, some usability techniques require a usability expert (most OSS projects do not have experts on their team). Second, some techniques require the participation of one or more users meeting face to face (OSS users are geographically distributed all over the world). Third, some techniques have to be applied step by step, prepared in advance or require preliminary information (OSS community work is wholly voluntary and done by members in their spare time) [25].

Although research examining usability in OSS has been published, there is no standardized procedure for determining how to adopt usability in OSS development [33] [34] [35]. Likewise Scacchi states that there is no internationally accepted OSS development model defining OSS development in practice [36]. The first step in our research is to study how the OSS community uses usability techniques in its development projects. In order to discover the activities conducted in the OSS development process, we conducted a systematic mapping study (SMS) [33].

Castro's is the only published research to study usability problems and techniques occasionally adopted in OSS projects in an integrated manner and to report the current state of usability in the OSS community [25]. As a result of the literature review, we can say that only one of the research papers reports a general and systematic proposal for integrating usability techniques into the OSS development process, considering the particular characteristics, philosophy and idiosyncrasy of the OSS development process, without forfeiting the essence of usability techniques [37]. Two SMS related to usability in OSS were conducted in advance of our research. A SMS reviews the literature on a particular field of interest [37]. The first SMS was conducted by Castro [25], reviewing papers published up until 30 July 2013. The second SMS was conducted with a search range from 1 August 2013 to 30 April 2015 [38] and later updated considering 30 July 2017 as the search end date.

The literature review discovered some papers reporting and describing studies that report the adoption in OSS projects of the user profile [27,39], personas [20,27,40,41], direct observation [27,42–44] and post-test information [45] techniques. Just a few papers reported the use of the user profile techniques: [27,39]. The user profiles technique has been adopted in some OSS projects (e.g., GIMP, a 3D animation package, a bitmap graphics application) to define representative user types. The adoption of this technique provided information about non-user developers [27,39]. These projects in particular had the resources required to apply the user profiles technique as specified by HCI.

Very few research papers have reported the use of the personas technique in OSS development projects [20,27,40,41]. According to research by Çetýn and Gokturk [20,40] and Terry et al. [27], the necessary information for applying the personas technique was gathered from descriptions provided by the OSS community and not through face-to-face interviews of user groups. Faily and Lyle [41] presented four guidelines that software engineering tools should incorporate to support the design and evolution of personas. This technique was not applied as specified by HCI because these guidelines are based on their experiences of modifying the open-source CAIRIS requirements management tool to support design and development activities for the EU FP7 webinos project.

Very few papers have reported the use of the direct observation technique [27,42–44]. According to Terry [27], direct observation was performed informally when family and friends of the developers used the application or when advanced users performed demonstrations at conferences. There is no predefined

object of study for such observations. This technique was adopted in several OSS projects, including a bitmap graphics application and a desktop tool. The direct observation technique was used to compare effectiveness, efficiency and satisfaction with task performance on two OSS library information management systems (Koha and Evergreen) [42]. In the Pika OSS project (tool for library catalogue information search), the direct observation technique was used to evaluate usability problems. This evaluation was carried out across two sessions. Four users participated in the first session. They were observed while performing tasks designed by the author. The second session took place after improving the interface design and introducing new system functions, during which four users performed new tasks set by the researchers [43]. Neither of the studies [42,43] identified adaptations to the direct observation technique for adoption in OSS projects. Jing et al. [44] evaluated the application of LITE, which is used to manage healthcare resources. Observation in this study was not performed in situ as specified by HCI. Instead, observation took place online via teleconference, where one of the researchers observed the participants during task performance, offered assistance and answered questions raised by participants about how the LITE tool worked.

In regard to the post-test information technique, we found only one paper reporting the application of this technique in an OSS project [45]. We discovered that this technique was adopted with adaptations in the Roguelike roleplaying game: a group of students led by a usability expert applied the technique standing in for usability experts as specified by the HCI community.

### **3. Research Method**

The research method used to validate our research is the multiple case study [46] according to which we gather the results and experiences of applying usability techniques adapted to OSS projects. This research method is used when the phenomenon under investigation (in this case, the adoption of an adapted usability technique) is studied within its real setting (in this case, an OSS project). OSS projects are the perfect setting for the case study reported here because OSS communities are, according to several authors [7,24,29,47], unfamiliar with usability techniques, do not have resources to conduct usability testing and no usability experts are involved in their projects [4,7,19,24,29,48,49]. Small project teams in particular have little information about what techniques are at their disposal for improving usability [20,49–52].

According to Runeson et al. [26] the key feature of a multiple case study is that it offers more information because: (i) more data are gathered than in an ordinary case study, and (ii) the different case study characteristics round out the information. However, it is important to note that the cases studies should never be confused with statistical replications and statistical sampling. Multiple case studies have to be replicated literally or heuristically based on the theoretical framework of the research. Therefore, the choice of the second and subsequent cases should be derived from the first case study because, otherwise, there would be no logical line of continuity of the research [26]. In other words, the cases cannot be selected at random, as they should be clearly related. All four of our case studies are related to each other by their characteristics (e.g., they are OSS development projects without usability experts).

The main criterion for the selection of this method (multiple case study) is that there is little room for experimentally manipulating the phenomenon under study: the adoption of usability techniques in a real OSS project [53]. Additionally, it is a qualitative study requiring the collection of data using techniques that do not aim to measure or associate measurements with numbers such as group discussion, open interviews, group or community interaction and introspection [53]. Therefore, we decided to use web artefacts (e.g., online surveys, forums, wikis) to collect data. The research questions are stated in Section 5 and are related mainly to how to adapt usability techniques for adoption in real OSS projects. We adapted four usability techniques. On one hand, user profiles, personas, were applied to the QUCS, PSeInt projects, respectively. On the other hand, the post-test information and direct observation techniques were applied together to the FreeMind and OpenOffice Writer projects. Figure 1 illustrates how we conducted our multiple case studies [26][53]. The bullet points listed below the activities (Conduct case study) within the methodological process of a multiple case study show how the information is output for the respective case study by applying techniques like virtual meetings, online surveys, forums, online interviews, observation and questionnaires.

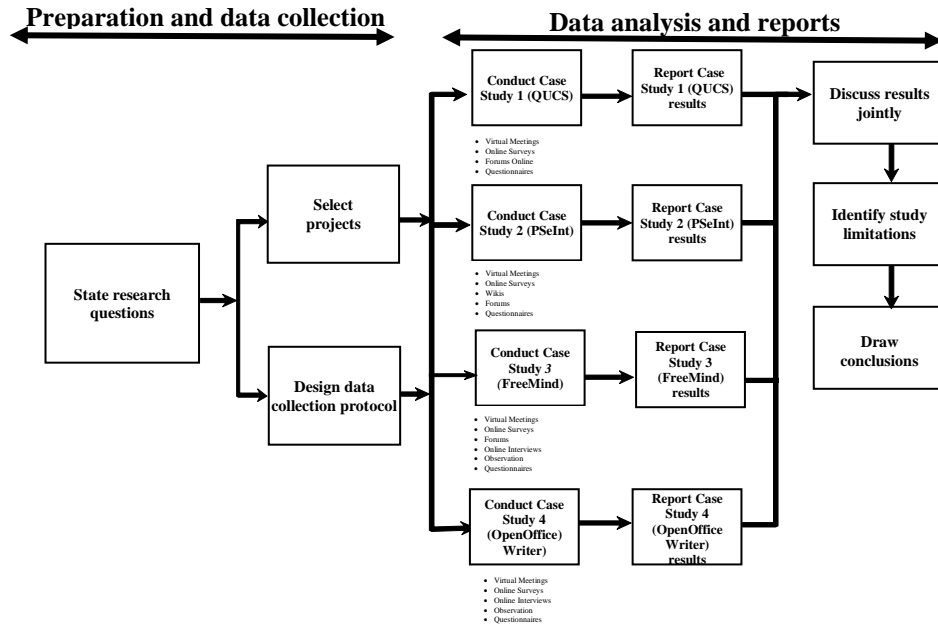


Figure 1: Activities carried out during the development of the multiple case study (adapted from [26][53])

#### 4. Identified Adverse Conditions and Usability Technique Adaptations

The techniques analysed in this research (user profiles, personas, post-test information and direct observation) need to be adapted for adoption in the OSS development process because these OSS communities have characteristics that the HCI world does not account for like, for example, worldwide geographical distribution of their members, a code-centred world view, unavailability of resources and a culture that may be somewhat alien to interaction designers [7] [20] [22]. Although usability techniques require conditions that cannot be generally met in the OSS world, they can be adapted to the idiosyncrasy of OSS projects.

The usability technique adaptation protocol entails specifying the techniques for adoption in the OSS development process. HCI authors do not specifically define the procedures for applying usability techniques in OSS projects. Besides, the procedures are, generally speaking, not defined in enough detail, that is, this problem applies not only to OSS but also to usability techniques generally. Therefore, the adaptation of the usability techniques that we propose is a four-stage process: (i) formalize all the steps to be taken to adapt the usability technique, that is, clearly specify the details of the steps of which the technique adaptation procedure is composed, (ii) analyse each of the steps to identify the adverse conditions that are an obstacle to technique application in OSS, (iii) propose the adaptations required to overcome the above obstacles to their adoption in the OSS development process, (iv) apply the adapted technique in the selected OSS project. In the remainder of this section, we describe the adaptations that have been taken into account for the adoption of these techniques in OSS projects.

##### 4.1. User profiles technique and its adaptations

User profiles are a way of gathering information about the planned system users [54]. Different procedures for applying this technique have been reported in the analysed literature [54,55]. The approach proposed by Mayhew [55] is a good option because it offers a comprehensive description of the technique as regards what to do and how to do it. According to Mayhew, the user profiles technique is divided into 14 steps. In the following, we describe some of these steps, detail the adverse conditions that are an obstacle to its adoption in OSS projects and specify the adaptations proposed to overcome these adverse conditions.

According to HCI prescriptions, project developers have to meet in person in Steps 1 and 2 (Determine user categories and key user characteristics) to discuss user categories and characteristics. This is a condition that cannot be met due to the special characteristics of OSS projects. Therefore, the technique needs to be adapted in order to remotely and asynchronously request feedback from the project administrator (via email) regarding the information required to create user profiles.

In Steps 3, 4 and 5 (Prepare, gather management feedback on and review the draft questionnaire), the researchers designed the draft questionnaire to create the user profiles. As the members of the OSS

community are distributed all over the world, it is not practicable to hold face-to-face meetings to discuss the structure and content of this draft questionnaire. Instead, we propose that the questionnaire design should be sent by email to the OSS project administrator for review. Table 1 summarizes the steps, identified adverse conditions and proposed adaptations for the user profile technique.

**Table 1.** Summary of the identified adverse conditions and proposed adaptations for the user profiles technique

User profile technique steps [55]	Adverse conditions	Proposed adaptations
1. Determine user categories.	<ul style="list-style-type: none"> <li>OSS project developers are required to meet in person.</li> </ul>	<ul style="list-style-type: none"> <li>Email the project administrator for a description of the categories and characteristics of the possible user profiles.</li> </ul>
2. Determine key user characteristics.		
3. Prepare draft questionnaire.	<ul style="list-style-type: none"> <li>Developers are required to participate face to face.</li> </ul>	<ul style="list-style-type: none"> <li>Email the project administrator for a review of the draft questionnaire by email.</li> </ul>
4. Gather management feedback on draft questionnaire.		
5. Review questionnaire.		
6. Pilot questionnaire.	<ul style="list-style-type: none"> <li>Users are required to participate face to face.</li> </ul>	<ul style="list-style-type: none"> <li>Meetings are online; they are replaced by a wiki where users can give their opinion on the questionnaire design.</li> </ul>
7. Fine tune questionnaire according to pilot test.	<ul style="list-style-type: none"> <li>There are no adverse conditions.</li> </ul>	<ul style="list-style-type: none"> <li>N/A</li> </ul>
8. Select user sample.	<ul style="list-style-type: none"> <li>Developers are required to participate face to face.</li> </ul>	<ul style="list-style-type: none"> <li>Ask the project administrator by email for the list of user emails.</li> </ul>
9. Distribute questionnaires.	<ul style="list-style-type: none"> <li>There are no adverse conditions.</li> </ul>	<ul style="list-style-type: none"> <li>N/A</li> </ul>
10. Design data input format.	<ul style="list-style-type: none"> <li>Document structure associated with this step is not specified.</li> </ul>	<ul style="list-style-type: none"> <li>Specify the structure of the output product.</li> </ul>
11. Enter data.	<ul style="list-style-type: none"> <li>A usability expert is required to apply this technique step.</li> </ul>	<ul style="list-style-type: none"> <li>Replace the usability expert by a team of junior experts supervised by a senior expert.</li> </ul>
12. Summarize data.		
13. Interpret data.		
14. Report data.	<ul style="list-style-type: none"> <li>(There are no adverse conditions. The proposed adaptation is based on the work method applied by the OSS community).</li> </ul>	<ul style="list-style-type: none"> <li>Publish study conclusions on forums and also distribute by electronic mailing list to OSS community.</li> </ul>

Table 2 gives, for each of the proposed user profiles steps, a brief description of the tasks to be performed for their application in an OSS project. The numbering of the original version (column 1, Table 1) and the adapted technique (column 1, Table 2) differs due to the changes made to the technique for the purposes of adaptation.

**Table 2.** Steps and tasks of the user profiles technique adapted for application in an OSS project

User profiles technique steps	Tasks
1. Determine user categories and key user characteristics.	<ul style="list-style-type: none"> <li>Email the project administrator to ask for possible user categories and characteristics.</li> </ul>
2. Prepare draft questionnaire.	<ul style="list-style-type: none"> <li>Create a questionnaire template adapted to OSS project needs.</li> </ul>
3. Gather feedback on the draft questionnaire.	<ul style="list-style-type: none"> <li>Ask the project administrator to review the draft questionnaire.</li> </ul>
4. Review the questionnaire.	<ul style="list-style-type: none"> <li>Review the feedback provided by the project administrator for inclusion in the questionnaire.</li> </ul>
5. Pilot the questionnaire.	<ul style="list-style-type: none"> <li>Invite two application users to take a pilot questionnaire via a wiki.</li> </ul>
6. Review the questionnaire.	<ul style="list-style-type: none"> <li>Review the feedback from the two users given via the wiki for inclusion in the questionnaire.</li> </ul>
7. Select user sample.	<ul style="list-style-type: none"> <li>Select a user sample for application in the final survey.</li> </ul>
8. Distribute questionnaires.	<ul style="list-style-type: none"> <li>Distribute the questionnaires as an online survey.</li> </ul>
9. Design data input template.	<ul style="list-style-type: none"> <li>Design a data input template as a spreadsheet to facilitate data summary.</li> </ul>
10. Data statistics.	<ul style="list-style-type: none"> <li>Enter, analyse and summarize the data using a format similar to the one suggested by [54].</li> </ul>
11. Report data.	<ul style="list-style-type: none"> <li>Report conclusions and design implications for the OSS community.</li> </ul>

#### 4.2. Personas technique and its adaptations

The aim of the personas technique is to output a representation of end users as guidance for application design [56]. This technique is capable of gathering, analysing and synthesizing information related to users that are to interact with the software system and, therefore, help to focus software analysis and design on the features and objectives of the product end user [54]. The personas technique cannot be applied directly in the OSS development process because the OSS community has characteristics to which the HCI world is unaccustomed (e.g., a code-centred world view, unavailability of resources and a culture that may be somewhat alien to interaction developers). The personas technique [56] is composed of seven steps. In the following, we describe the several steps of this technique and report the adverse conditions that are an obstacle to its adoption in OSS developments.

The aim of Step 1 of the technique (Identify behavioural variables) is to identify the behavioural variables of product end users (e.g., attitude towards information technologies). Cooper et al. [56] suggest that users be interviewed to gather the necessary information from which to put together the behavioural variables. However, OSS project users are geographically distributed all around the world. Thus, this characteristic of OSS projects is an adverse condition for technique application. To deal with this adverse condition, we propose that users participate in an online survey. The researchers designed the questionnaire (questions and measurement scales) used in the online survey. Step 2 (Map interviewed subjects to behavioural variables) establishes ranges on the behavioural variables identified from scalar responses to the questions designed for the survey proposed in Step 1. Cooper et al. [56] do not specify how to calculate the behavioural variable ranges for mapping. This step has to be performed by a usability expert, which is an obstacle because experts very seldom participate in OSS development. As no usability expert is available, we propose substituting the usability expert by a team of junior experts supervised by a senior expert. In the Step 3 (Identify significant behavioural patterns), Cooper et al. [56] do not specify how to identify behavioural patterns. This step requires technique and usability expertise. As no usability expert is available, we propose substituting the expert by a team of junior experts supervised by a senior expert. Table 3 summarizes the adverse conditions identified and the main adaptations proposed for Cooper et al.'s personas technique [56].

**Table 3.** Summary of the identified adverse conditions and proposed adaptations for the Personas technique

Personas technique steps [56]	Adverse conditions	Proposed adaptations
1. Identify behavioural variables.	<ul style="list-style-type: none"> <li>No users or physical spaces are available for face-to-face meetings.</li> </ul>	<ul style="list-style-type: none"> <li>Users participate online via an online survey.</li> </ul>
2. Map interviewed subjects to behavioural variables.	<ul style="list-style-type: none"> <li>Expertise of people familiar with the technique and usability is required.</li> </ul>	<ul style="list-style-type: none"> <li>The usability expert is either a developer, an expert OSS project user, or a team of junior experts supervised by a senior expert.</li> </ul>
3. Identify significant behavioural patterns.		
4. Synthesize key characteristics and goals of personas.	<ul style="list-style-type: none"> <li>Expertise of people familiar with the technique and usability is required.</li> <li>The format of the document associated with this step is not specified.</li> </ul>	<ul style="list-style-type: none"> <li>The usability expert is either a developer, an expert OSS project user, or a team of junior experts supervised by a senior expert.</li> <li>The format of the output product is specified.</li> </ul>
5. Check for redundancy and completeness.	<ul style="list-style-type: none"> <li>Expertise of people familiar with the technique and usability is required.</li> </ul>	<ul style="list-style-type: none"> <li>The usability expert is either a developer, an expert OSS project user, or a team of junior experts supervised by a senior expert.</li> <li>The format of the output product is specified.</li> </ul>
6. Expand the description of personas attributes.	<ul style="list-style-type: none"> <li>Expertise of people familiar with the technique and usability is required.</li> <li>The format of the document associated with this step is not specified.</li> </ul>	
7. Define and specify types of personas.	<ul style="list-style-type: none"> <li>Expertise of people familiar with the technique and usability is required.</li> </ul>	

We adopted some of the steps proposed by Cooper in order to facilitate the application of the Personas technique [56]. Table 4 shows, for each of the adapted personas technique steps, the tasks to be performed for their application in an OSS project.

**Table 4.** Steps and tasks of the adapted personas technique for application in an OSS project

Steps of the adapted personas technique	Tasks
1. Identify behavioural variables and map behavioural variables.	<ul style="list-style-type: none"> <li>Formulate preliminary virtual survey to identify personas.</li> </ul>
2. Identify significant behaviour patterns in the behavioural variables and synthesize key characteristics and goals of personas.	<ul style="list-style-type: none"> <li>Cluster the survey data of the virtual survey administered to the user segments.</li> <li>Analyse the data from the virtual survey of the user segments administered in the previous step.</li> </ul>
	<ul style="list-style-type: none"> <li>Define personas.</li> </ul>
3. Check for redundancy and completeness.	<ul style="list-style-type: none"> <li>Formulate the second and third virtual survey for the developer and user segments (selected at random).</li> <li>Analyse the virtual survey data administered in the previous step.</li> <li>Refine the created personas.</li> </ul>
	<ul style="list-style-type: none"> <li>Specify the primary and secondary personas.</li> </ul>
	<ul style="list-style-type: none"> <li>Create the stories describing the personas specified in the previous step.</li> </ul>
4. Expand, describe and define types of personas.	

### 4.3. Direct observation technique and its adaptations

The direct observation technique consists of directly observing individual users performing specially prepared tasks or performing their routine work. Therefore, it requires an observer to take down the behaviour or record the performance of users, for example, by measuring the time taken to complete specific sequences of actions. Three or more users are required to apply the direct observation technique



[57]. Nielsen [57] and Preece [58] study the direct observation technique. The approach recommended by Preece is a good option because it is the simplest description of how to apply the technique. In the following, we describe the only step of the direct observation technique according to Preece, the identified adverse conditions and the proposed adaptations for adoption in OSS development projects.

The only step for the execution of the direct observation technique is (data collection). It involves visiting users while they are doing their job [58]. The goal is for the observer to take notes on what he or she sees. The observers should be non-intrusive so as to ensure that the users do their job as usual. At times, the user may be interrupted to ask questions to clarify the activities that he or she is performing, although this should be done as little as possible. This step is an obstacle insofar as the users of OSS applications are geographically distributed all over the world, and observation at the site where they use the applications is out of the question. Additionally, a usability expert acting as an observer is required to apply the technique. On this ground, the technique needs to be adapted for application.

Three adaptations are required. First, we propose the use of a biased sample including family and friends of the observer as users. Second, we suggest replacing the usability expert by a team of junior experts supervised by a senior expert. Third, we recommend observing geographically distributed OSS users remotely. For this purpose, we used different tools to transfer text, audio and video files over the Internet.

Table 5 shows, for each of the adapted direct observation technique steps, the tasks to be performed for their application in an OSS project.

**Table 5.** Steps and tasks of the adapted direct observation technique for application in an OSS project

Steps of the adapted direct observation technique	Tasks of the adapted direct observation technique
1. Prepare the observation session.	• A usability expert establishes goals, requirements and task to be performed by users.
	• Users are invited via the project forum or email to participate in technique application.
	• The session is scheduled according to the dates and times when participants are available.
2. Conduct the observation session.	• Users are asked whether they are interested in participating in technique application.
	• Participants are informed about the reasons for the usability study.
	• The observation session should be a non-intrusive as possible.
3. Report the results of the observation session.	• A usability expert must analyse, summarize and report the results of executing the observation session to the OSS community.

#### 4.4. Post-test information technique and its adaptations

The post-test information technique involves interviewing each user at the end of the usability test. The aim of the interview is to ask the user for feedback and suggestions with regard to both the test conducted and the tool that is being evaluated [59]. In the analysed literature, the post-test information technique was studied by Constantine [59]. In the following, we describe what is, according to Constantine, the only step of the post-test information technique, the adverse conditions that are an obstacle to the adoption of this technique in OSS development projects and the proposed adaptations to overcome the above conditions. The only step of this technique (Hold interview) requires the participation of a usability expert who is responsible for designing the interview and interviewing the user [59].

As OSS application users are geographically distributed all over the world, and, as a general rule, OSS projects do not have usability experts on their team, it is necessary to make three adaptations to be able to apply the post-test information technique. First, we propose recruiting a biased sample including family and friends of the interviewer as users. Second, geographically distributed OSS users should be interviewed remotely. Third, the usability expert should be replaced by a team of junior experts supervised by a senior expert. Table 6 shows, for each of the adapted direct observation technique steps, the tasks to be performed for their application in an OSS project.

**Table 6.** Steps and tasks of the adapted direct observation technique steps for application in an OSS project

Steps of the adapted post-test information technique	Tasks of the adapted post-test information technique
1. Decide the interview approach.	• Define the main aim of the interview.
2. Create the interview.	• Design the interview questions.
3. Pilot test the interview.	• Pilot test the interview design to assure that it is user friendly.
4. Hold the interview.	• Perform the interview.
5. Report the results.	• Analyse and report the results to the OSS community.

Note that, before conducting each case study, users were first asked to give their consent to participate in our research. Therefore, the usability techniques were applied in the OSS projects by the users that gave their consent.

## 5. Design and plan the multiple case study

We use a multiple case study as the research method to validate our research. Depending on the context of the case, a case study can be designed as a single or multiple case study. A multiple case study is composed of two or more cases from different contexts [26]. Case studies are one of the most popular forms of qualitative empirical research [46]. A case study investigates the phenomenon of interest in its real-world context. The phenomenon of interest for this research is the adoption of adapted usability techniques, whereas the real-world context is OSS projects. The study design is not experimental, because we neither assign subjects at random nor control the study groups. Additionally, subjects are observed in a real-world setting [60]. It is not easy to run controlled experiments in the field of OSS because the characteristics of OSS communities (e.g., availability, expertise, experience, etc.) are unmanageable. Since not all OSS project team members have the same characteristics, it is impossible to minimize the effects of external factors (e.g., geographic distribution and time differences). This rules out evaluation by means of an experiment. On this ground, we selected the case study methodology to validate the feasibility of our proposal for adopting a usability technique in an OSS project.

There are several approaches to research based on case studies. Considering the adopted paradigm, we decided to use a positivist case study approach on two grounds. First, the user sample for each OSS project has different values. Therefore, the sample cannot be specified because the OSS users are volunteers. Second, the researcher must be a member of the OSS community to apply the usability techniques. A positivist case study is a qualitative method that is particularly suitable for researching information systems. Research based on positivist case studies is characterized by: (i) the study must focus on real situations, the phenomenon is studied in its real-world environment, (ii) only one or just a few entities (people, group, community) are examined, (iii) the phenomenon of interest is not isolated from its context and there is no controlled observation involving the manipulation of the experimental unit [61]. Our research has these characteristics.

We describe the case study following the guidelines set out by Runeson and Host [26]. According to these guidelines, we divide our research into two parts: an exploratory part and a descriptive part. We start by looking at what happens in a real-world scenario and then we describe what happens when we apply the adapted techniques to improve application usability [26]:

**RQ1:** How can the user profile, personas, direct observation and post-test information usability techniques be adopted in real OSS projects?

**RQ2:** Which are the types and characteristics of the OSS projects in which it is possible work with users and experts to adopt adapted usability techniques?

The usability techniques selected in this research are: (i) user profiles and personas (related to user analysis), and (ii) direct observation and post-test information (related to evaluation). Note that the last two techniques (direct observation and post-test information) were combined and applied in two OSS projects to gather better results and jointly test the usability of the selected applications.

The above techniques were applied to the QUCS, PSeInt, FreeMind and OpenOffice Writer projects. Initially, this research focused on three educational projects (PSeInt, QUCS and FreeMind). However, it was then extended to another special-purpose project (OpenOffice Writer). Thanks to this OSS project diversity, we were able to test and evaluate the usability technique integration framework in different fields and contexts. The selected case studies cover different application types (electronics, programming, mental map designer and text editor).

### 5.1. Case Study Projects

Below we describe the key characteristics of the OSS projects selected for applying the usability techniques.

**Case 1. QUCS.** The case study on the adoption of the user profiles technique was conducted on the QUCS project. QUCS is a multiplatform application for simulating electronic circuits. The size of this application is 504,526 lines of code, and it is written primarily in C++. It was developed to run on the GNU/Linux operating system, although it also operates on Windows, Solaris, NetBSD, FreeBSD, MacOS and Cygwin. We contacted Guilherme Brondarri Torri, the project administrator, who was receptive to our research. Even though he was unfamiliar with usability techniques, he was interested in the issue. The

project administrator could not provide a list of application user emails. However, he told us that QUCS application users are mostly students who are new to the electronics field.

**Case 2. PSeInt.** The personas technique case study was conducted on the PSeInt OSS project. PSeInt is software that uses a simple, intuitive pseudo language written in Spanish designed to help students without programming experience to understand basic and fundamental concepts of a computational algorithm. The size of the software is 35 thousand lines of code, and it is written mostly in C++. We did not manage to contact the project administrator who did not reply to our email asking for collaboration. Neither did we have access to an electronic mailing list of the most representative users of the application. However, when we asked about their level of usability knowledge, we received an email from Pablo Novara, member of the developer team stating that they had not addressed usability issues.

**Case 3. FreeMind.** The case study was conducted on the FreeMind project, and two techniques (direct observation and post-test information) were combined to conduct the usability evaluation. FreeMind is an OSS tool for building mind maps whose size is 255.6 thousand lines of code programmed in Java. We contacted the project administrator Christian Foltin, who was neither familiar with the most representative users of the application nor had a list of emails to start up our research. The administrator admitted that he did not have any usability background in order to improve project user interface.

**Case 4. OpenOffice Writer.** The case study was conducted on the OpenOffice Writer project, and two techniques (direct observation and post-test information) were combined to conduct the usability evaluation. OpenOffice is currently one of the most popular OSS projects. The size of the software is 11.2 million lines of code, and it is mainly written in C++. OpenOffice Writer is 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. We managed to contact one of the project administrators (Rob Weir), who did have an electronic mailing list of real users of this application. In the case of OpenOffice Writer, we had read that this project reckoned with usability experts. However, when we contacted the administrators, they told us that the usability team was not operational.

## 5.2. Preparation and data collection

The data collection protocol is more or less the same for all usability techniques. First, we contacted the OSS project administrators to express our interest in applying usability techniques. None of the administrators of the selected projects, except the OpenOffice Writer, had a list of users or had identified representative users. Second, we considered that the user was familiar with tool operation and we applied similar procedures to apply the usability techniques. In the first place, we used the official OSS community forums to send out the invitation to participate in technique application. Later, however, we used social networks to publicize the usability technique application in OSS projects and increase the user participation rate.

We created web artefacts to efficiently synchronize the necessary activities to apply the adapted usability techniques and improve communication with OSS community members. The web artefacts used to make the proposed adaptations of each usability technique were: online surveys, wikis and forums. For the personas technique, we ran an online survey created using Google Forms to gather the behavioural variables and create the possible OSS application user profiles. For the user profiles technique, we used a wiki to fine tune the questionnaire used as a basis for determining the user profiles. Additionally, we used the official project forum to invite users to participate in technique application. For the direct observation technique, we scheduled a remote meeting where we used a form to record the behaviour of the user and the time taken to complete previously defined tasks. For the post-information technique, we created an online survey to record the comments and suggestions about the evaluated tool taken from the interview of each subject. Thanks to the web artefacts, we were able to create a virtual meeting point with OSS users to apply the techniques because these users are geographically distributed all over the world. These web artefacts were designed to gather results separately for each case study because each usability technique requires specialized information to achieve its aim.

The data collected in all four case studies are mainly qualitative. We were unable to use the same data collection methods across all the case studies because: (i) differences in the characteristics of each of the selected OSS applications and (ii) the availability of OSS project users, as most are volunteers and do not have much spare time. On this ground, we used different data sources so as not to limit the results of the data interpretation. For example, we gathered the data provided by the developers and users in each OSS project by means of virtual meetings (such as wikis or forums) and online surveys. Table 7 gives an overview of the data collection methods and tools used in the four case studies.

**Table 7.** Methods and tools used for collecting data for each case study.

Case	Applied Usability Technique	Virtual Meetings	Online Surveys	Wikis	Forums	Online Interviews	Observation	Questionnaires
Case 1: <b>QUCS</b>	User profiles	X	X		X			X
Case 2: <b>PSeInt</b>	Personas	X	X	X	X			X
Case 3: <b>FreeMind</b>	Direct observation and post-test information	X	X		X	X	X	X
Case 4: <b>OpenOffice Writer</b>	Direct observation and post-test information	X	X			X	X	X

### 5.3. Data analysis and reports

The most important part of the data analysis process in this multiple case study was the creation of tables and forms to summarize the information gathered from online questionnaires, virtual meetings, interviews, observation and video calls. As the HCI prescriptions on usability techniques do not provide for any formalized document or special-purpose tool for gathering information and reporting results during the application of the usability techniques, we propose several new table and form templates for recording and presenting the information gathered from each of the case studies.

We analysed the data from each case study in response to the stated research questions. As the data collected from each of the case studies are qualitative, we do not report a statistical analysis. However, we do provide details on the adaptations made (Section 4) to the usability techniques for adoption in OSS projects. Additionally, we state the types and characteristics of the OSS projects in which these usability techniques can be adopted (see Section 5.1).

Our multiple case study addresses each case separately, where each case study provides its own conclusions, as well as supplying data for the research as a whole. It is important to accurately describe the information gathered from both users and developers of the selected OSS projects in order to elaborate the general principles of our research.

## 6. Results of the multiple case study

Below we describe the results of the case studies (QUCS, PSeInt, FreeMind and OpenOffice Writer) in which the adapted usability techniques (user profiles, personas, direct observation and post-test information) were adopted.

### 6.1. QUCS Case

The adapted user profiles technique was applied to the QUCS OSS project. QUCS application users are mostly students that are new to the electronics field. We made contact with the project administrator at the very beginning of our research, and he was receptive to the application of this technique. We had no trouble recruiting users, as QUCS users were interested in participating and finding out more about the user profiles technique.

To create the user profiles, we designed a questionnaire based on the proposal by Mayhew [55]. Additionally, we created the “circuitoselectricoslu”<sup>5</sup> wiki using the PBWorks<sup>6</sup> tool to encourage users to participate in the design of the preliminary questionnaire. Then we used the comments gathered from the wiki and the feedback provided by the administrator to improve the preliminary version of the survey. Later, we built an online survey using the improved version of the survey for publication in the SourceForge community forum. Finally, 14 people completed this survey over a four-month period. The online survey used for data collection is available at the web site<sup>7</sup>. Figure 2 shows an excerpt of the form summarizing the user profiles data created for later publication in QUCS project forums. By applying the adapted user profiles technique, we were able to form a rough idea of the QUCS application user types: a segment composed of students and another segment composed of teachers.

During the application of this technique, we detected three problems: (i) it is difficult to gather key common user profile characteristics due to their worldwide geographical distribution, (ii) it is hard to schedule a meeting with users to discuss the structure and content of the questionnaire due to the different time zones, and (iii) a usability expert needs to be on hand to analyse the gathered data.

<sup>5</sup> <http://circuitoselectricoslu.pbworks.com/w/page/123051369/Electric%20Circuits>

<sup>6</sup> <http://www.pbworks.com/>

<sup>7</sup> <http://goo.gl/EPsIz6>

QUCS PROJECT SUMMARY OF USER PROFILE DATA	
<b>User Category:</b> Intermediate	
<b>Demographic Characteristics</b>	
1.	Age: More than 25 years
2.	Gender: Male
3.	Employment Status: Teacher
4.	Professional Title: Electronic Engineer
<b>Attitude and motivation</b>	
5.	Place where you use your computer: At work.
6.	Computer user experience: More than 11 years
7.	Have you used the circuits application? Yes
8.	Place where you use QUCS: Work
9.	Knowledge area uses QUCS: Circuits
<b>Knowledge and experience</b>	
10.	Years of experience you have in issues related to circuits: Less than 5.
11.	How long you used QUCS? Less than 1 year
12.	Hours you spend in a week using QUCS: Less than 3 hours
13.	Level of competence with QUCS: Intermediate

Figure 2: Excerpt of the form summarizing the user profiles data

## 6.2. PSeInt Case

The adapted personas technique was applied to the PSeInt OSS project. It was hard to recruit real PSeInt users because the project administrator did not reply to our invitation to collaborate in the research. As a result, we did not have access to a list of user electronic mails and had no idea who the most representative users of the PSeInt application were.

With the aim of the gathering the necessary information to apply the adapted personas technique, we designed a preliminary survey in order to identify behavioural variables related to users' attitudes towards technology, computer literacy, frequency, motivation and purpose of application use, profession or trade. The PSeInt survey was build using Google Forms and is available at the web site<sup>8</sup>. The survey questions had a scalar format so that we could group a set of items by multiple values.

We published the PSeInt survey on the official OSS project site forum and received only six responses over the established four-month period. In view of the low participation rate, we opted to publicize the survey again, this time on social networks, achieving a participation of 55 users over a four-month period. Thanks to this high concentration of 55 responses plus the original six responses gathered from the official project forum, we were able to form a critical mass and cluster the results to draft personas. We used the Weka tool to analyse the resulting responses to the PSeInt survey, running the k-means algorithm to generate the clusters. K-means is a clustering method. We use k=2 with the aim of outputting two segments (cluster 1 and 2), where cluster 1 stands for the primary and dominant persona and cluster 2 for the secondary persona. Table 8 summarizes the variables and the dominant attribute in each cluster. The characteristics of the primary and secondary personas are similar, differing with respect to the computer literacy, tool user type and PSeInt expertise variables.

Table 8. Variables and dominant attribute for each cluster

Variables	Dominant attribute for Cluster 1	Dominant attribute for Cluster 2
Age	15-20	15-20
Educational level	University student	University student
Computer literacy	Average	High
PSeInt user type	Beginner	Intermediate
PSeInt place of use	Education	Education
PSeInt user type	Occasional	Occasional
PSeInt expertise	Beginner	Advanced

Having identified the primary and secondary personas, we checked for redundancy and completeness. To do this, we again surveyed a group of developers and users that were representative of these populations and selected at random from the PSeInt survey. The surveyed persons answered questions about the previously identified behavioural variables and a psychological test based on the Big Five model. We administered two new virtual surveys. The first survey, called Personas, is available at the web site<sup>9</sup>, and

<sup>8</sup> <http://goo.gl/forms/mjBTUwYfRA>

<sup>9</sup> <https://goo.gl/ffYnGr>

the second survey, called Personality, at the web site<sup>10</sup>. The Personas survey was useful as an instrument for validating the data gathered from the PSeInt survey administered out in Step 1 of the technique. Through psychological testing, we gathered more thorough knowledge of people's behavioural characteristics. Based on the data gathered from the surveys and their respective analysis, we drafted the Personas Foundation. This document contains a summary of the key characteristics and goals of the created personas. Figure 3 shows an excerpt of the Personas Foundation document for the primary persona. This document is the main output of the Personas technique.

PERSONAS FOUNDATION DOCUMENT	
PERSONA IDENTIFICATION	
a.	Full name: Ángel Peña
b.	Age: 18 years
c.	Glasses wearer
ROLES AND TASKS	
a.	Ángel is a third-semester computer engineering student.
b.	Ángel attends classes in the mornings.
c.	He is a video gamer and plays an instrument in his leisure time.
d.	He has a regular routine.
e.	His responsibilities are typical of professional training in computer sciences.
GOALS	
a.	Ángel Peña's current goals are to graduate as computer engineer and find a job
SEGMENT	
a.	Ángel Peña was born and lives in Puerto Ordaz, Venezuela.
b.	His marital status is single.
c.	His educational level is upper secondary education and he is currently at university.
SKILLS AND KNOWLEDGE	
a.	Ángel Peña likes to see programs running, and this is major motivation for programming.
b.	Ángel Peña is familiar with several programming languages.
c.	He is a computer user since he was 6 years old and now uses computers for education.
CONTEXT AND KNOWLEDGE	
a.	Ángel Peña used the PSeInt programming tool for learning purposes.
b.	Ángel Peña believes that a manual outlining PSeInt would be necessary to understand how this tool works.
c.	Ángel Peña currently uses similar tools for programming.
d.	Ángel Peña is not motivated to continue to use PSeInt for learning purposes.
PERSONAL/PSYCHOLOGICAL DETAILS	
a.	Ángel Peña is a sociable, friendly and outgoing person
b.	He is currently convinced that he wants to study computer science.
c.	He is a conscientious person.

**Figure 3:** Excerpt of the foundation document for the primary Persona

### 6.3. FreeMind Case

The adapted direct observation and post-test information techniques were applied to the FreeMind project. Before contacting users, we entered into conversation with the OSS project administrator to inform him of our interest in applying usability techniques and seek his support. The project administrator (Christian Foltin) responded positively to our request and showed an interest in collaborating with this type of initiatives. The project administrator did not have a list of representative user emails.

For our research, we considered two user groups: (i) our family and friends, and (ii) real users (registered in subscriber lists and who regularly use the OSS application). Additionally, real users were recruited from official community forums. Note that all users were classed into two user profiles (junior and senior). In both cases (friend and family users and real users), they were classed as junior or senior user according to the experience that they each had as application users.

The protocol enacted in the direct and remote (usability expert participates online) observation sessions is more or less the same. There were both junior and senior users to whom an almost identical procedure was applied. At the start of each session, arranged beforehand with the user, they were instructed that they were to perform a number of tasks using the application while an observer took notes. They were told that it was the application and not the user who was being evaluated. The recorded information reported the problems and difficulties encountered during the activity, as well as any gestures denoting feelings towards the application. Figure 4 shows an excerpt from the document designed to record the problems encountered by a junior FreeMind user.

<sup>10</sup> <https://goo.gl/WHWQom>

Tasks to be performed using the FreeMind tool (Observer's notes)				
Task to be performed by user		Does the user know how to perform the task?	Did the user encounter any problems?	Other comments
a.	Open the FreeMind application	Yes	No	
b.	Create a new map	Yes	No	The application is already open.
c.	Label the central node as <i>Mental map laws</i>	Yes	No	
d.	Insert a new sibling node labelled <i>Images</i>	Yes	Yes, the user has difficulty finding the sibling node.	
e.	Insert a new sibling node labelled <i>Words</i>	Yes	No	
f.	Insert a child node labelled <i>Symbols</i> within the <i>Words</i> node	Yes	No	
g.	Delete the node labelled <i>Images</i>	Yes	No	
h.	After deletion you realize that the node is necessary, create the <i>Images</i> node again	Yes	No	The user does not use Ctrl Z, but creates the node again
i.	Insert a child node labelled <i>Structure</i> within the central node	Yes	No	
j.	Insert an image of your choice in the node labelled <i>Words</i>	Yes	No	
k.	Insert a new node labelled <i>Style</i> in the central node	Yes	No	
l.	Insert the icon representing the exclamation mark (!) in the node labelled <i>Style</i>	Yes	No	The user uses the options displayed by a right-hand mouse click
m.	Insert another icon, this time representing a family, in the node labelled <i>Structure</i>	Yes	Yes, because it is very similar to the other step.	The user tries to search for the icon using the right-hand mouse button but the icon is not displayed.
n.	Delete the first icon inserted, that is, the exclamation mark (!)	Yes	Yes, the user has to be told where the delete option is	The user deletes the exclamation mark (!) using the respective icon located on the left-hand side of the interface.
o.	Save the mental map as Mental.Map in the Documents folder	Yes	No	

**Figure 4:** Document for recording problems during direct/remote observation technique application

Finally, after completing the tasks, the user was asked to take an interview to gather post-test information at the end of which he or she was thanked for participating. Contrary to HCI prescriptions, the interviewer was not a usability expert but a HCI student. The user merely answered the questions asked by the interviewer (for example, what are the key usability issues you came across?) and the interviewer took down the answers. The interview was always supervised by a mentor whose comments were designed to draw out the key points and help the interviewee provide as much information as possible. Figure 5 shows the document used to conduct the interview.

We should highlight some divergences in the methodology depending on the type of user participating in the session. All the communication with users participating in a remote observation had, of course, to take place online, including explanations, supply of materials, post-test interview and data collection. Specifically, we used two tools: Skype to speak to the user and see his or her reactions, and TeamViewer to visualize interaction with the application by remotely accessing the screen view. TeamViewer had a positive feature, namely, option of recording audio and video, thanks to which we were able to analyse the sessions as many times as necessary. This ultimately turned out to be a very useful resource because we had to replay the recordings several times in order to settle doubts (for example, determine whether or not a user performed one of the tasks as specified in the delivered document) that arose from analysis of the written report on the data collected.

Information Collection Document during Post-Test Information HCI Technique Application	
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">           Interview date: _____            Interviewer: _____            Interview type: <input type="checkbox"/> Direct (face to face) <input type="checkbox"/> Remote            Name of respondent _____         </div> <div style="border: 1px solid black; padding: 10px;"> <ol style="list-style-type: none"> <li>1. What did you find most difficult? _____</li> <li>2. What were the main usability problems that you had? _____</li> <li>3. Do you have any suggestions for improving the interaction with the application? _____</li> <li>4. Do you have any suggestions for improving the test? _____</li> <li>5. Do you have any criticism or complaint about the user interface? _____</li> <li>6. How do you think that the user interface (or part of the user interface) could be redesigned? _____</li> </ol> </div>	

**Figure 5:** Information collection document for applying the post-test information technique

A total of 22 FreeMind users participated in the application of the direct observation and post-test information techniques. The first half was composed of family and friends who we could contact personally. The second half included family and friends, plus real application users from different parts of the world. The FreeMind real user recruitment process for remote observation turned out to be doubly challenging, as the project administrator was neither acquainted with the most representative users of the application nor able to provide a list of user emails as a starting point. As a result, we decided, at the project administrator's suggestion, to search FreeMind help forums for user emails. As it turned out, not all the forum members were suitable targets for technique application as, from the type of messages that they published (for example, errors when compiling a particular Java class), some appeared to have overly technical profiles. Therefore, we decided to select users that published topics or subjects asking for help with the use of FreeMind.

A total of 100 users, including friends, family and real users recruited from the FreeMind project help forums, were contacted by email. The email sent to these users asked whether they would be interested in participating in the application of usability techniques. A total of 18 replied to the email. A second email was sent to this group of interested users describing the usability techniques, the activities that they would be expected to perform, the time it would take, the tools that they would need, informing them of the available time slots and asking whether and when they would be available to perform the activities. A total of 13 users replied to this second email. A third email was sent to these users to ask for their Skype user name, confirm the date and time of the meeting and specify the websites from which they could download the tools that they would need. We received a response from six users. Finally, we managed to set up a virtual meeting with four out of these six users, whereas another two failed to connect on the date and time as arranged. Table 9 summarizes the number of users that finally participated in the test by observation type.

**Table 9.** Number of FreeMind users by observation type.

Observation Type	Junior	Senior	Total
Direct Observation	7	5	12
Remote Observation	4	6	10
<b>TOTAL</b>	<b>11</b>	<b>11</b>	<b>22</b>

As a result of the adoption of the direct observation technique, we identified different usability issues regarding the FreeMind application. Table 10 shows an excerpt of the list of the most highly relevant problems encountered during technique application and the improvement suggested by users for each problem.



**Table 10.** Examples of problems encountered by users during direct observation.

Problem	User	No. of users	Improvements proposed by users
The insert new node is not readily visible.	Junior	2	Improve the visibility of the node insertion and deletion options.
Symbols are hard to find.	Junior	2	.Classify icons by categories.
The insert child node option is not readily visible.	Junior	1	Improve the visibility of the node insertion and deletion options
It is not possible to delete one of several icons within a particular node.	Senior	3	Enable deletion of an icon that is not necessarily the last icon inserted.
Symbols are very small.	Senior	1	Place the symbols in the tool bar and upsize.
The insert sibling node option is not readily visible.	Senior	1	Improve the visibility of the node insertion and deletion options.

Like Table 10, Table 11 shows some of the most highly relevant problems encountered by junior and senior users during the remote application of the direct observation technique.

**Table 11.** Examples of problems encountered by users during the remote application of the direct observation technique.

Problem	User	No. of users	Improvements proposed by users
The only option enabled by clicking on a node is edit associated text.	Junior	2	Add more actions to be performed on the node.
Icon deletion is confusing for the user because he or she cannot identify which icon was inserted last.	Junior	2	Enable the deletion of any icon.
Image is too big.	Junior	2	Resize image.
The node colour is not updated when a change is made.	Senior	2	Immediately update the node colour after a change.
Not all of the icons are accessible from the right-hand mouse button.	Senior	1	Enable access to all the icons by right clicking on the mouse.
The user has to save the document when an image is inserted.	Senior	1	Do not require the user to save when inserting an image.

#### 6.4. OpenOffice Writer Case

As for the FreeMind case, we applied the adapted direct observation and post-test information techniques to the OpenOffice Writer project. For the purposes of user classification (junior and senior), users were asked to complete a remote questionnaire which was emailed to them. This questionnaire was designed together with one of the project administrators. Based on a preliminary version and after several exchanges of electronic mails with improvement proposals for the questionnaire, we arrived at a final version that satisfied both parties (administrator-researchers). The information from this survey was used to classify users.

For OpenOffice Writer, 16 users participated in the application of the direct observation and post-test information techniques (half were family and friends, the other half were real users taken from the list of project emails). Unlike the other case studies (QUCS, PSeInt, FreeMind), OpenOffice Writer did have a list of user emails and we did not have to search forums. The OpenOffice Writer mailing list included 9,000 users. As the emails were confidential, one of the project administrators sent the first email to all the users. This email asked users to complete a survey and sought their permission to share the survey data with a group of researchers. One of the requested particulars was an email address to contact the user. A total of 1,121 users completed the survey, of which 956 gave permission to share their information, but only 644 users provided their email address. Of the 644 email addresses, nine were rejected by the mail server (non-existent addresses). Finally, we managed to contact 635 users, of which 132 replied.

Of these 132 users, 60 declined to participate, whereas 72 did show an interest. After reading the 72 emails, we made a distinction between two different user groups. On one hand, there were users who were interested in participating but, due to particular circumstances (for example, they do not have a web camera, slow internet connection, cannot speak due to an illness), were unable to do so. On the other hand, there were users interested in and able to participate. Each of these two groups (junior and senior) happened to be composed of 36 users. The classification was made based experience as application users. We sent an email to a preliminary group of 15 users at random, willing and able to participate, informing of the available time slots and asking whether and when they would be available for a virtual meeting to apply the techniques. Depending on the results for this first user group, e-mails would be sent to a second user group if more users were required. Finally, we did not need to email any more users for this case study. Of the 15 emails, 10 received a reply. Finally, we managed to meet with a total of eight users. Two of the users declined to participate in the end, even though they had originally showed an interest, whereas the other one cancelled the appointment at the last minute due to incident at work. Table 12 summarizes the number of users that finally

participated in the OpenOffice Writer case study. Figure 6 illustrates the instructions given to junior users for task performance.

**Table 12.** Number of OpenOffice Writer users by observation type.

Observation Type	Junior	Senior	Total
Direct Observation	5	3	8
Remote Observation	4	4	8
<b>TOTAL</b>	<b>8</b>	<b>8</b>	<b>16</b>

**Tasks to be performed using OpenOffice Writer**  
(Junior user)

Use the OpenOffice Writer application to perform the task outlined below.

Perform the following actions:

1. Open the OpenOffice Writer application.
2. Open a text document
3. Create a new document
4. Type the following text:  
Apache OpenOffice is an open-source office productivity software suite. It contains a word processor, a spreadsheet, a presentation application, a drawing application, a formula editor, and a database management application. It is available for several operating systems, like Microsoft Windows, GNU/Linux, BSD, Solaris and MacOS X. It can read and write a wide variety of file formats, including the default OpenDocument Format file format (ODF), an ISO/IEC standard, as well as over 110 languages since 2010.
5. Justify the margins of the above paragraph
6. Right hand mouse click to change the font of the above paragraph to Times New Roman and the font size to 11..
7. Add the following footnote after the Word Solaris: Unix-based operating system originally developed by Sun Microsystems.
8. Insert an image (representing a tourist) from the OpenOffice Writer Gallery.
9. Frame the image inserted above.
10. Change the left-hand page margin to 3 cm.
11. Number all the pages of the document in the bottom left hand corner.
12. Insert a table with two columns and seven rows. Use the Grey Table-Autoformat. The resulting table should be like the one below:

Application Name	Description
Writer	Word processor similar to Microsoft Word
Calc	Spreadsheet similar to Microsoft Excel or Lotus 1-2-3
Impress	Presentation program similar to Microsoft Power Point or Apple's Keynote
Base	Database program similar to Microsoft Access
Draw	Vector graphics editor and diagramming tools similar to Microsoft Visio.
Math	Application design to create and edit mathematical formulas.

13. Vertically centre the content of the Application Name column.
14. Save the file as OpenOffice-tasks in the OpenOffice text format (that is, with the .odt extension) to your desktop.
15. Export the file to .pdf format and save it as OpenOffice-tasks to your desktop.

**Figure 6:** Tasks to be performed by junior users with the OpenOffice Writer application.

As a result of the adoption of the direct observation technique, we identified several problems with the OpenOffice Writer application. Table 13 shows the most highly relevant problems encountered by junior and senior users.

**Table 13.** Examples of problems encountered by users during direct observation.

Problem	User	No. of users	Improvements proposed by users
Right-hand mouse click does not display all fonts.	Junior	5	Enable the selection of any font type using the right-hand mouse button.
Menu options are unclear.	Junior	3	Increase the visibility of the basic options.
The <i>insert</i> menu does not include the <i>gallery</i> option.	Junior	3	Include the <i>gallery</i> option in the <i>insert</i> menu.
Page numbers do not appear at the bottom of the page.	Junior	2	Automatically insert page numbers (in the footer).
Not all fonts are accessible using the right-hand mouse button.	Senior	2	Enable the selection of any font type using the right-hand mouse button.
Page numbers do not appear at the bottom of the page.	Senior	2	Automatically insert page numbers (in the footer).
The <i>insert</i> menu does not include the <i>gallery</i> option.	Senior	1	Include the <i>gallery</i> option in the <i>insert</i> menu.

Like Table 13, Table 14 shows some of the most highly relevant problems encountered by junior and senior users during the remote application of the direct observation technique. These problems are directly related to application use.

**Table 14.** Examples of problems encountered by users during the remote application of the direct observation technique.

Problem	User	No. of users	Improvements proposed by users.
The <i>insert</i> menu does not include the <i>gallery</i> option.	Junior	2	Include the <i>gallery</i> option in the <i>insert</i> menu.
It is hard to find the required options in the <i>tools</i> menu	Junior	2	Redesign the <i>tools</i> menu to make it more intuitive.
The <i>insert</i> menu does not include the <i>gallery</i> option.	Senior	1	Include the <i>gallery</i> option in the <i>insert</i> menu.
The Table-Autoformat tool is not readily visible and is only available during table creation.	Senior	1	Make it possible to change the autoformat after creating the table.

## 7. Discussion of Results

In this section, we discuss and answer the research questions stated in this research.

### RQ1: How can usability techniques be adopted in real OSS projects?

Table 15 below shows a comparative summary of the obstacles found during the application of the usability techniques.

**Table 15.** Summary of the obstacles and adaptations associated with the adapted usability technique

Usability Technique	Obstacle	Associated adaptation
User profiles	<ul style="list-style-type: none"> <li>Meetings between developers and users are hard to arrange.</li> <li>Data analysis requires a usability technique expert.</li> </ul>	<ul style="list-style-type: none"> <li>Developers participate online by electronic mail and users participate via a wiki.</li> <li>The usability expert is replaced by a team of junior experts supervised by a senior expert.</li> </ul>
Personas	<ul style="list-style-type: none"> <li>User participation is necessary to apply the technique.</li> <li>A usability expert is required to apply the technique.</li> </ul>	<ul style="list-style-type: none"> <li>Users participate online by means of an online survey.</li> <li>The usability expert is replaced by a team of junior experts supervised by a senior expert.</li> </ul>
Direct observation	<ul style="list-style-type: none"> <li>User participation is necessary.</li> <li>Meetings with users are hard to arrange.</li> <li>A usability expert is required to apply the technique.</li> </ul>	<ul style="list-style-type: none"> <li>Biased sampling including family and friends of the observer as users.</li> <li>The usability expert is replaced by a team of junior experts supervised by a senior expert.</li> <li>Observations/interviews of geographically distributed OSS users are remote.</li> </ul>
Post-test information		

Usability techniques were created for another type of software development, that is, were not designed taking into account the specific characteristics of the OSS development process. On this ground, it is necessary to adapt the techniques. These adaptations are based on the adverse conditions for technique application. Some of the adverse conditions are overcome using certain web artefacts (for example, wikis, forums, etc.), which are known to the OSS community. Consequently, OSS community members will be familiar with many of the adaptations, and this is likely to encourage the application of usability techniques. Below, we describe the major technique adaptations.

There are mainly two adaptations for the **user profiles** technique. First, users participate online via electronic mail and a wiki. Second, we suggest that the usability expert be replaced by a developer, expert user or a team of junior experts supervised by a senior expert. Again there were mainly two adaptations for the **Personas** technique. First, users participate online using electronic mail. Second, we suggest that the usability expert be replaced by a developer, expert user or a team of junior experts supervised by a senior expert. In both cases, the usability expert was replaced by a team of junior experts supervised by a senior expert.

Three adaptations were necessary to apply the **direct observation** technique. First, we propose the use of biased sampling including family and friends of the observer as users. Second, the role of usability expert is played by a team of junior experts supervised by a senior expert. Third, we suggest that geographically distributed OSS users be observed remotely. For such observations, we used different tools for text, voice and video transfer over the internet.

In view of the fact that OSS applications users are geographically distributed and, as a general rule, OSS projects do not have access to usability experts, it is necessary to carry out three adaptations to be able to apply the **post-test information** technique. First, we propose the use of biased sampling including family and friends of the observer as users. Second, geographically distributed OSS users should be observed remotely. Third, we replace the role of usability expert by a team of junior experts supervised by a senior expert. When applying the usability techniques to the OSS projects, we met with problems that vary depending on the technique and selected OSS project (PSeInt, QUCS, FreeMind and OpenOffice Writer). Below, we describe these obstacles.

During the application of the user profiles and personas techniques, we came up against two problems. First, it is difficult, due to time zone differences, to arrange meetings with users and developers in order to create and discuss documents associated with these techniques. Second, a usability expert is required to analyse and interpret the results of the application of the techniques. As a result of the application of both techniques, we discovered that to be able to define user segments, the material for gathering the information must be distributed using social networks, due to the low OSS community participation rate (i.e., of users contacted through the OSS forum). Additionally, OSS user participation was lower than expected on two grounds. First, we expected, based on statistics (user ratings and downloads this week) reported on the official website of the applications (QUCS and PSeInt), to get a large number of users to participate. Second, it was hard to contact and engage users to participate in this research.

When applying the direct observation and post-test information techniques, we met with three problems. First, it was difficult to arrange an appointment with each user due to time differences between the different countries where they live. Second, some users had problems with their Internet connection. This made it difficult to see what they were doing on screen or clearly hear what they were saying. Third, a minority of users did not have the necessary programs installed on their computers (even though they were repeatedly informed beforehand), on which ground they required help. This added to the technique application time. As a result of the application of both techniques, we discovered that the number of users interested in participating is greater when their participation does not require a big time investment. For example, if participants are asked to install additional software or enable remote access to their computer, the number of users interested in participating drops considerably. Note, however, that we always found committed users for which time was not an obstacle.

In the following, we describe the impact generated using the adapted usability techniques with the application in OSS projects. We suggested to the QUCS and PSeInt project developers that they should take into account the results of the application of the personas and user profiles techniques when developing features taking into account the characteristics of the users that they target. By applying the post-test information and direct observation techniques in the FreeMind case study, we identified several usability problems that were reported to the project developers. We later found that the developers of this project had solved some of the reported usability problems (for example, symbol visibility, icon deletion within a node containing more than one icon, poor visibility of *insert new node*, *insert child node* and *insert sibling node* options). We found that the OpenOffice Writer project developers did not solve the problems that we identified and reported. In this particular case, however, we created two plug-ins (the first plug-in<sup>11</sup> to insert a page number in the header/footer and the second plug-in<sup>12</sup> to apply the capital letter in a text paragraph) in order to improve some OpenOffice Writer features. Due to the small number of users, the plug-ins were not sent to the OpenOffice Writer project for feedback from developers at this preliminary stage of the research project, which is due to be conducted on a larger scale in the near future.

Considering the software life cycle, the user profiles and personas techniques, which are part of the user analysis activity during the requirements engineering stage, would be adapted and applied before requirements elicitation to provide details about prospective software users, as OSS user segments are not previously defined. On the other hand, the post-test information and direct observation techniques, which are part of the evaluation stage, would be adapted and applied after creating the software product so that users can execute the most relevant tasks and evaluate system usability.

Note that the results of the usability technique application were sent to the selected software project developers. Some responded that our results would be used at a later date. Note, however, that these results are potentially applicable by any OSS development team. On one hand, the results of the application of the user profiles technique are useful for developers to determine the profile of the users of their applications and thereby facilitating requirements elicitation tasks. Likewise, the results of the personas technique would facilitate the adoption of usability mechanisms as part of requirements engineering activities, thus helping to improve the usability of the software system under development. On the other hand, the post-test information and direct observation techniques can help developers understand how users perform their tasks and correct the problems encountered before releasing the software product.

---

<sup>11</sup> [https://drive.google.com/open?id=1\\_LKagXaysohK7PWjQB\\_YWaMNwEUDMc7A](https://drive.google.com/open?id=1_LKagXaysohK7PWjQB_YWaMNwEUDMc7A)

<sup>12</sup> [https://drive.google.com/open?id=1kjTlmeooF8\\_K2Jn4JgXjeTNo-S\\_rwhKR](https://drive.google.com/open?id=1kjTlmeooF8_K2Jn4JgXjeTNo-S_rwhKR)

**RQ2: Which are the types and characteristics of the OSS projects in which it is possible work with users and experts to adopt adapted usability techniques?**

There is a wide variety of OSS projects classified according to the level of development, size of community and resources: (i) large projects with an organized and structured user community and with company support (for example, Sun Microsystems contributed to GNOME [62]), (ii) medium-sized projects with a sizeable community of members (from 100 to 500 users) and a growing group of developers (from 5 to 10 developers) without company support (for example, FreeMind), (iii) small projects with very few developers and without a list of representative users or company support. Depending on the project type, the adaptations of the technique may vary. For example, for large projects that have HCI teams [19], it will not be necessary to make adaptations in this respect. Then again, it is necessary to implement usability technique adaptations for small projects that do not have many resources.

The QUCS, PSeInt and FreeMind projects are medium-sized communities and they are voluntary projects that do not have budgets, which means that they cannot afford external experts. In our research, we adapted usability techniques to medium-sized projects because we believe that developers do not have access to the resources (usability experts or lists of users to contact) in order to conduct usability testing. Rajanen et al. [45], who report how to introduce usability in OSS projects, offer an illustrative example of the shortcomings of this type of projects. On the above grounds, this type of projects could benefit from the strengths of applying usability techniques to write quality software outside a commercial environment.

Note that the results of adapting the user profiles and personas techniques may be applicable to similar (medium-sized) projects to the ones in which we participated as volunteers (PSeInt, QUCS and FreeMind). With regard to the extrapolation of our proposal to large-scale projects, however, we should bear in mind that the obstacles will be very different to the ones identified in our study on the following grounds: (i) large projects are very active and popular in OSS development, (ii) they report errors in multiples sources, where solutions to common problems are accessible across different web artefacts such as electronic mail lists, forums, chats or wiki, and (iii) they have documented their practices and tests. These web artefacts enable bug reporting and usability problem discussion within the community. Therefore, this type of OSS projects provide for dialogue between usability experts and developers, which facilitates the adaptation of usability techniques.

OpenOffice Writer is a large and longstanding community. It is a large-scale OSS project, which is well organized and well structured, and it also has a large number of users. Regarding the application of usability techniques (direct observation and post-test information), the project administrator had a list of users but had not identified which were representative. Generally, OSS developers working on large-scale projects are not acquainted with the profile of their users [24,63].

Finally, note that the adaptation of the techniques for the PseInt, QUCS and FreeMind projects mainly affected the projects. However, the adaptation of the techniques for the OpenOffice Writer project affected both the project and the team. Although the developer playing the role of PSeInt project administrator did not respond to our email asking for authorization to apply usability techniques, another developer did answer our questions about the project later on. With respect to QUCS and FreeMind, the developer playing the role of administrator gave us permission to apply the techniques and always answered our questions. In the OpenOffice Writer project, not only did one of the administrators give us permission to apply the usability techniques and answered our questions, he also directly participated in the design of the questionnaire used to apply the user profiles technique.

## **8. Study limitations**

There are some limitations on the validity of this study, because it is eminently qualitative [64]. We identified the limitations of our results using the guidelines provided by Runeson [26].

As regards construct validity, only one person provided feedback to improve the wiki-based survey design in the case of QUCS. The information gathered on survey design was sent to the OSS project administrator who had to make the respective corrections, that is, the data collection instrument was validated by the project administrator but no more feedback was received before administering the final online survey. No construct validation problems were identified for the other cases (PSeInt, FreeMind and OpenOffice Writer).

With respect to internal validity, age, educational level and tool user experience may have had a positive or negative influence on the use profiles usability technique in the case of QUCS. Initially, survey

credibility may vary depending on whether or not data are omitted or the meaning of the question is distorted. The online survey should be applied again at different times throughout the research process to ensure that the results are not biased. No internal validation problems were identified for the other cases (PSeInt, FreeMind and OpenOffice Writer).

With regard to external validity, the main limitation of our study in the case of QUCS and PSeInt, is that we adapt only one technique for application to only one OSS project. Therefore, more case studies need to be carried out applying the selected usability technique to other OSS projects in order to validate the proposed adaptations. On the other hand, the selected usability techniques were combined in the case of FreeMind and OpenOffice Writer to improve the application usability results. Finally, with regard to reliability, all the case studies should provide for the participation of all OSS user types, that is, we should not just recruit real users (regular OSS tool users that are registered with the community). Therefore, before applying the techniques, we suggest that other options for recruiting users who are genuinely interested in participating in this type of research (for example, social networks) should be explored.

## 9. Conclusions

The aim of this research was to adapt and evaluate the feasibility of adopting HCI usability techniques in real OSS projects. Technique adaptations are necessary to address some adverse conditions caused by OSS development characteristics that are an obstacle to the application of techniques prescribed by the HCI area. In particular, we made adaptations to the user profiles, personas, direct observation and post-test information for application in the QUCS, PSeInt, FreeMind and OpenOffice Writer OSS projects.

The main adverse conditions that we found the techniques to have in common are: (i) the need for a usability expert to apply the technique, (ii) user participation, and (iii) the need for several users to be present in person to apply the technique. To address these conditions, we propose four adaptations: (i) substitute the usability expert by a team of junior experts supervised by a senior expert (a MSc in Usability Engineering student supervised by two HCI expert researchers); (ii) remote user participation, that is, the users use web artefacts (for example, online surveys, wiki); (iii) use biased sampling including family and friends of the observer as users; and (iv) conduct remote observations/interviews of geographically distributed OSS users using different tools for text, voice and video transfer over the Internet.

Collaboration is one of the principles underlying OSS community [27,65]. However, we did not get much collaboration during the application of the usability techniques, perhaps because real users (i.e., users registered on the SourceForge project website) were short of time, unfamiliar with the importance of usability, or had no incentive to participate. On this ground, we suggest using social networks to publicize technique application and recruit as many participants as possible.

The results for these case studies (QUCS, PSeInt, FreeMind) will possibly be applicable to small- and medium-sized OSS projects like the ones in which we have participated as volunteers, because small- and medium-sized OSS projects do not have a large user base, do not report bugs in multiple sources, solutions to common problems are not accessible through online infrastructures (e.g., email lists, forums, etc.), and their work practices and tests are not documented.

In the following, we detail the key recommendations that researchers and developers should consider to apply usability techniques in OSS projects: (i) encourage the OSS community to start to attach importance to software development questions addressed by HCI, (ii) use other means of communication (like social networks) to promote the application of techniques and not only official OSS project forums, (iii) promote some sort of incentive to recruit as many participants as possible for these initiatives, and (iv) consider the software development status when selecting the OSS technique and project.

After analysing and applying the usability techniques in the requirements engineering and evaluation activities for OSS development projects, we found that all four of the implemented case studies had adverse conditions, such as the number of participants, biased information provided by developers, geographical and temporal distribution, and OSS community motivation. Despite all these problems, it is feasible to apply the adapted usability techniques in OSS projects. We believe that it is necessary for this community to start to attach importance and take note of the repercussion of the software development issues dealt with by HCI. Thus, as it is necessary to adapt the HCI techniques for adoption in OSS development projects, this community should take a broader view of software development to consider its usability and not just focus on feature development.

Below we detail how our research has advanced the state of the art in this field. Terry et al. [27] propose the reconceptualization of HCI methods for adaptation to OSS but do not explain how to carry out this

reconceptualization, that is, they do not define the steps required to carry out each HCI and the web artifacts to be used for adoption in OSS. However, we set out the adaptations for the adoption of some usability techniques in OSS, specifying the steps and tasks associated with each step. These adaptations are implemented in different case studies (real OSS projects). Some researchers have evaluated the usability of OSS applications without the participation of application users [42] [43]. However, our research was carried out with real users, as OSS users have different requirements and needs to developers. Additionally, the problems that we found in our evaluations were validated by experts (junior and senior) and then presented to the developers in order to improve their application usability. Castro [25] proposes a general framework for usability technique integration in OSS development projects. This technique framework is very generic and does not report the steps for applying the above techniques in OSS. The research reported here is an advance insofar as it specifies the usability techniques for application in the OSS projects, specifically describing the steps for each technique, as the procedures proposed by the original authors are not explicitly defined.

As future work, we intend to conduct further case studies to adapt and apply other usability techniques for OSS projects. These OSS projects will preferably be small- or medium-sized because, unlike large projects that do have online infrastructures for bug reporting and usability problems, they do not have the resources to achieve usability improvements. We will analyse other web artefacts that can be adapted to OSS communities in order to improve communication (for example, social networks) and gradually raise the awareness of OSS developers about the benefits of applying HCI usability techniques.

Considering that OSS has some characteristics in common with other types of software, such as global software development (GSD), where collaborative members with cultural differences are separated by distance and work in different time zones [12] and start-ups, which have small teams that are geographically distributed all over the world and a shortage of resources, the proposed adaptations could also be extrapolated to deal with some of the negative issues faced by other similar project types.

## Acknowledgements

This research was 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 Spanish Ministry of Education, Culture and Sports FLEXOR (TIN2014-52129-R) and TIN2014-60490-P projects and the eMadrid-CM project (S2013/ICE-2715). Finally, this research received funding from the University of Atacama “DIUDA 22316” project.

## References

- [1] G. Schryen, R. Kadura, Open Source Vs. Closed Source Software, in: 2009 ACM Symp. Appl. Comput. - SAC '09, ACM, 2009: pp. 2016–2023. doi:10.1145/1529282.1529731.
- [2] J. Noll, W.-M. Liu, Requirements Elicitation in Open Source Software Development: a Case Study, in: 3rd Int. Work. Emerg. Trends Free. Source Softw. Res. Dev. - FLOSS '10, ACM, 2010: pp. 35–40. doi:10.1145/1833272.1833279.
- [3] G. Madey, V. Freeh, R. Tynan, The Open Source Software Development Phenomenon: An Analysis Based on Social Network Theory, in: Eighth Am. Conf. Inf. Syst., 2002: pp. 1806–1813. [http://www3.nd.edu/~oss/Papers/amcis\\_oss.pdf](http://www3.nd.edu/~oss/Papers/amcis_oss.pdf).
- [4] D.M. Nichols, M.B. Twidale, The Usability of Open Source Software, First Monday. 8 (2003) 21. doi:<http://dx.doi.org/10.5210/fm.v8i1.1018>.
- [5] A. Raza, L.F. Capretz, F. Ahmed, Maintenance Support in Open Source Software Projects, in: Eighth Int. Conf. Digit. Inf. Manag. (ICDIM 2013), IEEE, 2013: pp. 391–395. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6694005>.
- [6] X. Ferré, N. Juristo, H. Windl, L. Constantine, Usability Engineering-Usability Basics for Software Developers, IEEE Softw. 18 (2001) 22–29.
- [7] A. Lisowska Masson, D. Lalanne, T. Amstutz, A Usability Refactoring Process for Large-Scale Open Source Projects, Proc. 2017 CHI Conf. Ext. Abstr. Hum. Factors Comput. Syst. - CHI EA '17. (2017) 1135–1143. doi:10.1145/3027063.3053345.
- [8] A. Hars, S. Ou, Working for Free? – Motivations of Participating in Open Source Projects, in: 34th Hawaii Int. Conf. Syst. Sci., IEEE, 2001: pp. 1–9. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=927045>.
- [9] A. Vourvopoulos, Usability and Cost-effectiveness in Brain- Computer Interaction : Is it User Throughput or Technology Related ? Usability and Cost-effectiveness in Brain-Computer Interaction : Is it User Throughput or Technology Related ?, (2016). doi:10.1145/2875194.2875244.

- [10] C. Garcia, F. Castro, J.I. Gomez, D. Chaver, J.A. Lopez-Orozco, OpenIRS-UCM: an Integral Solution for Interactive Response Systems, *Int. J. Eng. Educ.* 32 (2016) 873–885.
- [11] A. Mockus, R.T. Fielding, J.D. Herbsleb, Two Case Studies of Open Source Software Development: Apache and Mozilla, *ACM Trans. Softw. Eng. Methodol.* 11 (2002) 309–346. <http://portal.acm.org/citation.cfm?doid=567793.567795>.
- [12] S. O'Mahony, Guarding the Commons: How Community Managed Software Projects Protect their Work, *Res. Policy.* 32 (2003) 1179–1198. <http://www.sciencedirect.com/science/article/pii/S0048733303000489>.
- [13] W. Scacchi, Understanding Requirements for Open Source Software, in: *Lect. Notes Bus. Inf. Process.*, Springer-Verlag., 2009: pp. 467–494. doi:10.1007/978-3-540-92966-6\_27.
- [14] H. Assa, S. Chiasson, R. Biddle, Cesar : Visual Representation of Source Code Vulnerabilities, 2016 IEEE Symp. Vis. Cyber Secur. (2016) 1–8. doi:10.1109/VIZSEC.2016.7739576.
- [15] N. Vila Blanco, L. Rodríguez-Liñares, P. Cuesta, M.J. Lado, A.J. Méndez, X.A. Vila, gVARVI: A graphical software tool for the acquisition of the heart rate in response to external stimuli, *Comput. Methods Programs Biomed.* 132 (2016) 197–205. doi:10.1016/j.cmpb.2016.05.005.
- [16] N.S.M. Yusop, J. Grundy, R. Vasa, Reporting usability defects - do reporters report what software developers need?, 20th Int. Conf. Eval. Assess. Softw. Eng. - EASE '16. (2016) 1–10. doi:10.1145/2915970.2915995.
- [17] D. Brun, S.M. Ferreira, C. Gouin-Vallerand, S. George, CARTON Project: Do-It-Yourself Approach to Turn a Smartphone into a Smart Eyewear, *Proc. 14th Int. Conf. Adv. Mob. Comput. Multi Media.* (2016) 128–136. doi:10.1145/3007120.3007134.
- [18] W. Liu, K. Purdon, T. Stafford, J. Paden, X. Li, Open Polar Server (OPS)—An Open Source Infrastructure for the Cryosphere Community, *ISPRS Int. J. Geo-Information.* 5 (2016) 32. doi:10.3390/ijgi5030032.
- [19] C. Benson, M. Müller-Prove, J. Mzourek, Professional Usability in Open Source Projects: GNOME, OpenOffice.org, NetBeans, in: *CHI'04 Ext. Abstr. Hum. Factors Comput. Syst. - CHI EA'04*, ACM, 2004: pp. 1083–1084. doi:10.1145/985921.985991.
- [20] G. Çetin, M. Gokturk, A Measurement Based Framework for Assessment of Usability-Centricness of Open Source Software Projects, in: 4th Int. Conf. Signal Image Technol. Internet Based Syst. - SITIS'08, IEEE, 2008: pp. 585–592. doi:10.1109/SITIS.2008.106.
- [21] A. Raza, L.F. Capretz, F. Ahmed, Users' Perception of Open Source Usability: An Empirical Study, *Eng. Comput.* 28 (2012) 109–121. doi:10.1007/s00366-011-0222-1.
- [22] A. Raza, L.F. Capretz, F. Ahmed, An Empirical Study of Open Source Software Usability: The Industrial Perspective, *Int. J. Open Source Softw. Process.* 3 (2011) 1–16. doi:10.4018/jossp.2011010101.
- [23] A. Raza, L.F. Capretz, F. Ahmed, An Open Source Usability Maturity Model (OS-UMM)., *J. Comput. Hum. Behav.* 28 (2012) 1109–1121.
- [24] D.M. Nichols, M.B. Twidale, Usability Processes in Open Source Projects, *Softw. Process Improv. Pract.* 11 (2006) 149–162. doi:10.1002/spip.256.
- [25] J.W. Castro, Incorporating Usability in the Open Source Software Development Process, Tesis Doctoral. Departamento de Ingeniería Informática. Universidad Autónoma de Madrid, 2014.
- [26] P. Runeson, M. Host, A. Rainer, B. Regnell, Case Study Research in Software Engineering: Guidelines and Examples, John Wiley & Sons., 2012.
- [27] M. Terry, M. Kay, B. Lafreniere, Perceptions and Practices of Usability in the Free/Open Source Software (FOSS) Community, in: *Proc. 28th Int. Conf. Hum. Factors Comput. Syst. CHI 2010*, ACM, 2010: pp. 999–1008. doi:10.1145/1753326.1753476.
- [28] H.A. Al-Odan, Ahmad A. Al-Daraiseh, Open Source Data Mining Tools, 2015 Int. Conf. Electr. Inf. Technol. (2015) 369–374. doi:10.1109/EITech.2015.7162956.
- [29] A. Ternauciuc, R. Vasiiu, Testing usability in Moodle: When and How to do it, in: 2015 IEEE 13th Int. Symp. Intell. Syst. Informatics, 2015: pp. 263–268. doi:10.1109/SISY.2015.7325391.
- [30] M. Rajanen, N. Iivari, Examining Usability Work and Culture in OSS, in: *Proc. 11th IFIP WG 2.13 Int. Conf.*, 2015: pp. 58–67. <http://link.springer.com/10.1007/978-3-319-17837-0>.
- [31] G. Çetin, D. Verzulli, S. Frings, An Analysis of Involvement of HCI Experts in Distributed Software Development: Practical Issues, in: D. Shuler (Ed.), *Online Communities Soc. Comput. - OCSC'07*, Springer., 2007: pp. 32–40. doi:10.1007/978-3-540-73257-0\_4.
- [32] H. Hedberg, N. Iivari, M. Rajanen, L. Harjumaa, Assuring Quality and Usability in Open Source Software Development, in: *Proc. First Int. Work. Emerg. Trends FLOSS Res. Dev. - FLOSS'07*, IEEE, 2007: pp. 1–5. doi:10.1109/FLOSS.2007.2.
- [33] S.T. Acuna, J.W. Castro, O. Dieste, N. Juristo, A systematic mapping study on the open source software development process, 16th Int. Conf. Eval. Assess. Softw. Eng. (EASE 2012). (2012) 42–46. doi:10.1049/ic.2012.0005.
- [34] J.W. Castro, S.T. Acuña, Diferencias entre las Actividades de Mantenimiento en los Procesos de Desarrollo Traduicional y Open Source, in: *Jornadas SISTEDES''2012*, 2012: pp. 651–664.



- [35] J.W. Castro, S.T. Acuña, Differences between Traditional and Open Source Development Activities, *Proc. 13th Int. Conf. Prod. Softw. Process Improv.* (2012) 131–144.
- [36] W. Scacchi, Free and Open Source Development Practices in the Game Community, *IEEE Softw.* 21 (2004) 59–66.
- [37] B. Kitchenham, D. Budgen, O. Pearl Brereton, Using Mapping Studies as the Basis for Further Research-A Participant-Observer Case Study, *Inf. Softw. Technol.* 53 (2011) 638–651. <http://dx.doi.org/10.1016/j.infsof.2010.12.011>.
- [38] L. Llerena, Transformación de Técnicas de Usabilidad Relacionadas con las Actividades de Ingeniería de Requisitos para su Incorporación en el Proceso de Desarrollo Open Source Software., Trabajo de Fin de Master. Departamento de Ingeniería Informática. Universidad Autónoma de Madrid, 2015.
- [39] E. Reitmayr, B. Balazs, J. Mühligh, Integrating Usability with Open Source Software Development: Case Studies from the Initiative OpenUsability, in: *TOSSad Work. Gov. Educ. Usability, Leg. Issues Toward Open Source Softw. Adopt. an Enlarg. Eur., Italy*, 2006.
- [40] G. Çetýn, M. Gokturk, Assessing Usability Readiness of Collaborative Projects, *Comput. Syst. Sci. Eng.* 26 (2011) 259–274. <http://www.mehmetgokturk.com/wp-content/uploads/2012/04/cetin-gokturk-crl2010.pdf>.
- [41] S. Faily, J. Lyle, Guidelines for Integrating Personas into Software Engineering Tools, in: *Proc. 5th ACM SIGCHI Symp. Eng. Interact. Comput. Syst. - EICS'13*, 2013: pp. 69–74. doi:10.1145/2494603.2480318.
- [42] J. Pruett, N. Choi, A Comparison Between Select Open Source and Proprietary Integrated Library Systems, *Libr. Hi Tech.* 31 (2013) 435–454. <http://www.emeraldinsight.com/journals.htm?issn=0737-8831&volume=31&issue=3&articleid=17097034&show=html>.
- [43] E. Gallinger, K.L. Neville, Usability in the Pika discovery layer : an academic and public library case study, 63 (2016) 261–265.
- [44] X. Jing, J.J. Cimino, G. Del Fiol, Usability and acceptance of the librarian infobutton tailoring environment: An open access online knowledge capture, management, and configuration tool for openinfobutton, *J. Med. Internet Res.* 17 (2015) 1–10. doi:10.2196/jmir.4281.
- [45] M. Rajanen, N. Iivari, E. Kesitalo, Introducing Usability Activities into Open Source Software Development Projects: A Participative Approach, in: *Proc. 7th Nord. Conf. Human-Computer Interact. Mak. Sense Through Des. Nord. '12*, ACM, 2012: pp. 683–692. doi:10.1145/2399016.2399120.
- [46] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, *J. Empir. Softw. Eng.* 14 (2009) 131–164. doi:10.1007/s10664-008-9102-8.
- [47] J. Blitzer, W. Schrefft, P.J.H. Schröder, Intrinsic Motivation versus Signaling in Open Source Software Development, *J. Comp. Econ.* 35 (2007) 160–169.
- [48] M. Rajanen, N. Iivari, Open Source and Human Computer Interaction Philosophies in Open Source Projects – Incompatible or Co-Existent?, in: *Proc. Int. Conf. Mak. Sense Converging Media*, ACM Press, 2013: pp. 67–74. <http://dl.acm.org/citation.cfm?id=2523429.2523463> (accessed August 19, 2015).
- [49] E.E. Northrop, H.R. Lipford, Exploring the Usability of Open Source Network Forensic Tools, in: *Proc. 2014 ACM Work. Secur. Inf. Work.*, 2014: pp. 1–8.
- [50] M. Rajanen, N. Iivari, Power, Empowerment and Open Source Usability, in: *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst. - CHI 2015*, 2015: pp. 3413–3422. [http://www.researchgate.net/publication/270899400\\_Power\\_Empowerment\\_and\\_Open\\_Source\\_Usability](http://www.researchgate.net/publication/270899400_Power_Empowerment_and_Open_Source_Usability).
- [51] J. Hall, The Usability of GNOME, *Linux J.* (2014).
- [52] L. Nielsen, M. Bødker, To Do or Not to Do: Usability in Open Source Development., *Interfaces (Providence)*. 71 (2007) 10–11.
- [53] R.K. Yin, *Case Study Research: Design and Methods.*, 5th Ed., SAGE Publications, 2013.
- [54] X. Ferré, Marco de Integración de la Usabilidad en el Proceso de Desarrollo Software., Tesis Doctoral. Facultad de Informática. Universidad Politécnica de Madrid, 2005.
- [55] D.J. Mayhew, *The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design*, Morgan Kaufmann., 1999.
- [56] A. Cooper, R. Reinmann, D. Cronin, *About Face 3.0: The Essentials of Interaction Design*, Wiley, 2007. <http://ivi.sagepub.com/lookup/doi/10.1057/palgrave.ivs.9500066>.
- [57] J. Nielsen, *Usability Engineering*, Morgan Kaufmann, 1993. <https://goo.gl/hT9BSz> (accessed July 6, 2015).
- [58] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, T. Carey, *Human-Computer Interaction*, 1st Ed., Addison-Wesley Pub. Co, 1994.
- [59] L.L. Constantine, L.A.D. Lockwood, *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*, ACM, 1999.

- [60] M. Genero Bocco, J.A. Cruz Lemus, M.G. Piattini Velthuis, *Métodos de investigación en ingeniería del software*, Ra-Ma, 2014.
- [61] Line Dubé and Guy Paré, Rigor in Information Systems Positivist Case Research: Current Practices, Trends, and Recommendations, *MIS Q.* 27 (2003) 597–636.
- [62] D.M. German, The GNOME Project: A case study of open source, global software development, *Softw. Process Improv. Pract.* 8 (2003) 201–215. doi:10.1002/spip.189.
- [63] M. Müller-Prove, User Experience for OpenOffice.org., *Interfaces (Providence)*. 71 (2007) 8–9.
- [64] J. Pérez, Dialnet-El Estudio de Casos como Estrategia de Construcción Teórica, *Cuad. Econ. y Dir. la Empres.* (1999) 123–140.  
<http://dialnet.unirioja.es/servlet/articulo?codigo=195459&info=resumen&idioma=SPA> (accessed August 18, 2015).
- [65] E.S. Raymond, *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, O'Reilly & Associates., 2001.