Interaction Design in Free/Libre/Open Source Software Development: a systematic mapping

Daniel Domingos Alves
Department of Computer Science
Federal University of Bahia (UFBA)
Salvador, Bahia, Brazil
daniel.domingos@ufba.br

Ecivaldo de Souza Matos
Department of Computer Science
Federal University of Bahia (UFBA)
Salvador, Bahia, Brazil
ecivaldo@ufba.br

ABSTRACT

Approaches for integrating interaction design into software development processes do not consider the specific development characteristics of free/libre/open source software (FLOSS). Researchers know the importance of integrating good practices of interaction design into the software development process. This paper aims to present a summary and analysis of methods, techniques, tools, strategies and approaches (MTTSA) to interaction design that have been proposed/used in the context of FLOSS development. A systematic mapping was performed to identify MTTSAs of interaction design proposed or used for/in the development of FLOSS. The results show that few studies have used MTTSA of interaction design in FLOSS context. No methods or techniques of interaction design proposed specifically for the development of FLOSS have found, and the majority of the selected papers do not present any type of validation through empirical studies. We hope that this paper provides an overview of studies that have used MTTSA of interaction design in FLOSS context, and becomes an initial effort to conduct new research proposals involving interaction design MTTSA and FLOSS development.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Author Keywords

Interaction Design, FLOSS development, Free Software, Open Source Software, Human-Computer Interaction, HCI.

INTRODUCTION

Free/libre/open source software (FLOSS) projects are virtual organizations/communities formed by geographically distributed developers [41]. Generally, the development of FLOSS is based on voluntary contributions. Developers using tools to coordinate and communicate work over the Internet, such as mailing lists and version control systems [41, 49]. The adoption of FLOSS has grown in the private sector, in public agencies and by end-users [11, 39, 49]. The number of FLOSS projects and contributions made by developers and employees paid by private companies has increased in recent years [11, 39]. The importance of FLOSS has increased [5]. This can be verified by the increase in research publications on FLOSS in the last years [11].

Scientific literature presents some problems of human-computer interaction (HCI) related to the development of FLOSS. There are problems related to usability and accessibility in FLOSS [1, 23]; despite the general impact of FLOSS, the HCI community has done few studies on FLOSS and there are few research collaborations of HCI researchers with the FLOSS community [5]; usability aspects are late considered in FLOSS projects [43]; most FLOSS projects have requirements defined fundamentally by their authors and little emphasis is given to usability [41].

A good process of interaction design is very important for development of interactive products. There are many studies to improve this process, specially with the development of new interaction design methods and techniques [53]. There are also approaches for integrating interaction design into software development processes [10, 46]. However, these approaches do not consider specific characteristics of FLOSS development such as distributed development environment, cultural differences and time zones.

FLOSS development presents some typical characteristics. One of them is the distributed software development, which makes it fundamentally different from co-localized software development [57]. The mainly models of interaction design process propose to engage users in diverse tasks with high level of communication and coordination of activities. Some interaction design techniques and approaches have been applied in co-localized software development, but which are not applicable or have to be adapted to distributed software development. This is because there are specific difficulties and challenges in FLOSS development context. Despite the importance of good interaction design practices in the software development process, little is known about methods, techniques, tools, strategies and approaches (MTTSA) of interaction design proposed/used in the context of FLOSS.

This paper aims to present a summary and analysis of methods, techniques, tools, strategies and approaches (MTTSA) to inter-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. IHC'17, Proceedings of the 16th Brazilian symposium on human factors in computing systems. October 23-27, 2017, Joinville, SC, Brazil. Copyright 2017 SBC. ISBN 978-85-7669-405-2 (online).

action design that have been proposed/used in the context of FLOSS development. A systematic mapping was performed to identify MTTSAs of interaction design proposed/used for/in the development of FLOSS.

In this paper, we present a discussion of the identified studies, providing evidence related to the proposals and/or use of MTTSAs of interaction design in FLOSS development. It is important to emphasize that this mapping is part of an ongoing research project, in which the authors are investigating how interaction design can be performed collaboratively in distributed software development environments. Thus, the results of this mapping will contribute to develop a collaborative interaction design process model for distributed software development environments.

BACKGROUND

In this section, we present a brief background on FLOSS, interaction design, and research that have been conducted involving the areas of HCI and FLOSS development.

Free/Libre/Open Source Software

FLOSS is a term that covers a variety of types of software and refers to computer programs that are distributed with their source code and allows freedom to use, study, copy, modify, and redistribute them [41, 49]. The term "free" is used with the sense of "freedom"; it does not refer to price. The four freedoms are described below [48, p.3]:

- the freedom to run the program as you wish, for any purpose;
- the freedom to study how the program works, and change it so it does your computing as you wish. Access to the source code is a precondition for this;
- the freedom to redistribute copies so you can help your neighbor;
- the freedom to distribute copies of your modified versions to others. By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

Questions about motivation for participation in FLOSS development were raised by [30, p. 198] and [8, p. 1243], respectively: "Why should thousands of top-notch programmers contribute freely to the provision of a public good?" and "Why do programmers write Open Source codes if no one pays them to do it?". According to Raymond [40], the participants in the FLOSS project have at least three reasons to write or contribute to writing FLOSS code. First, participants are interested in using the FLOSS product. For this reason, they contribute to its development and can benefit directly from the software developed by them. Second, they can enjoy the programming work itself. The last reason is related to the reputation of the participant in the FLOSS community. For Lerner *et al.* [30], the interest in the development of FLOSS has been motivated by three reasons:

- rapid diffusion of FLOSS, in which several FLOSS products, such as web server and operating system distributions, have become popular and potential competitors of proprietary software;
- significant capital investments in FLOSS projects, in which large corporations have launched projects to develop and use FLOSS products;
- the new organizational structure, in which the collaborative nature of the development of FLOSS has been praised by the commercial and technical press as an important organizational innovation.

FLOSS development process has the following characteristics [41, 49]:

- developed by communities of geographically distributed developers;
- voluntary contributions via the Internet;
- cultural and time zones differences;
- use of tools to enable communication among the geographically distributed teams;
- use of version control tools and software repositories;
- most projects have a small team, and the average number of individuals per team is five.

According to Reis *et al.* [41], most FLOSS projects have requirements that are fundamentally defined by their authors and a significant part of the projects is based on other pre-existing software. In addition, little emphasis is placed on usability as well as on conventional quality assurance activities.

Some studies have shown that many aspects of FLOSS development processes differ from traditional software engineering processes [33, 37, 42, 50] and others describe the way in which FLOSS development phases have occurred [27, 40, 41, 45]. In addition, software process models for the FLOSS development have also been proposed as, for example, Open Source Maturity Model (OSMM) - Navica [18], Qualification and Selection of Open Source software (QSOS) [44], Open Business Readiness Rating (OpenBRR) [52] and Open Source Maturity Model (OMM) - Qualipso [54].

Although there is a distinction between the terms "free software" and "open source", this systematic mapping encompasses studies related to free software as well as studies related to open source software. The differences between the movements of free software and open source software will not be discussed here. In addition, studies will not be classified as being of one or another movement, since the fundamental difference between the two movements lies in their values and their ways of looking at the world, rather than in their software development processes [47].

Interaction Design

Preece *et al.* [38, p. 319] states that "Design is a practical and creative activity that aims developing a product that helps its users to achieve their goals". Various definitions of interaction design can be found. Lowgren [34] describes in a simple but comprehensive way what interaction design is: "(...) about

shaping digital things for people's use". Preece *et al.* [38, p. 8] understand that interaction design is "(...) to support the way people communicate and interact in their everyday and working lives". The Interaction Design Association [3] states that "Interaction Design (IxD) defines the structure and behavior of interactive systems. Interaction Designers strive to create meaningful relationships between people and the products and services that they use, from computers to mobile devices to appliances and beyond".

Whittaker [53] reports that there is consensus in the HCI area regarding the interaction design process: "Everyone in HCI agrees that design involves four iterative steps: understanding users, generating designs, prototyping, and evaluation.". For Preece *et al.* [38] the interaction design process involves four basic activities [38]:

- establishing requirements - activity that aims to understand who the users are, their activities and the context of these activities, so that the developed product can support them in achieving their goals;
- designing alternatives - this activity begins when some requirements have been established and aims to suggest ideas to satisfy the requirements;
- prototyping - activity that involves the construction and experimentation of prototypes in an iterative way in order to test ideas regarding the viability and acceptance of the user;
- evaluating - activity that seeks to ensure the quality and performance of tests to verify that the final product is in accordance with its purpose.

When performing these activities, it is important to involve the users so that developers can understand their goals and thus build a more suitable and usable product [18].

Although FLOSS developers and interaction designers aim to build quality software that meet the needs of users, they each have their own perspective. FLOSS developers are generally concerned with the activities related to the technical part of the development and the functionalities of the software [23], while the designers approach interaction design activities with the participation of the users, and their concern is that the users achieve their objectives [38].

Relationships between HCI and FLOSS Development

Some studies [1, 2, 15, 23, 36], workshop [5] and special interest group (SIG) [4, 43] have highlighted the gap between software design approaches in HCI and FLOSS development. Although the HCI and FLOSS areas are complementary, these studies highlight the lack of integration between them. However, some initiatives and proposals have involved the areas of HCI and FLOSS development. These studies are presented below.

Hedberg *et al.* [23] proposed an extended FLOSS development project organization model that adds a new level of communication and roles for attending human aspects of software. The proposed model makes the existence of HCI specialists visible in a projects, and promotes interaction between developers and the HCI specialists.

Andreasen *et al.* [2] states that FLOSS developers are interested in usability. However, usability is not a priority. FLOSS projects rarely employ systematic usability evaluation. Most developers have a very limited understanding of usability and there is a lack of resources and evaluation methods in FLOSS paradigm.

Bach *et al.* [4] organized a SIG with the purpose of bringing together HCI professionals and researchers to discuss current issues in FLOSS. The SIG discussed usability, the role of HCI expertise, and design rationale in FLOSS projects. The authors state that HCI professionals are working on usability issues in FLOSS, but the HCI community has not yet organized with respect to FLOSS. Schwartz *et al.* [43] also organized a SIG, but with the aim of encouraging the participation of the user experience (UX) community and identifying solutions for better integration of UX into the FLOSS development process.

Nichols *et al.* [36] reviewed the existing evidence of the usability of FLOSS and discussed how the characteristics of FLOSS development influence the usability quality. The paper addresses how existing HCI techniques can be used to leverage distributed networked communities, of developers and users, to address issues of usability.

Bach *et al.* [5] organized a workshop that brought together researchers and practitioners from the HCI and FLOSS communities to establish an agenda for future research and collaboration between the two communities. The workshop research agenda included the study of the FLOSS community, tools development and methods for the community, and collaboration to solve shared problems.

We note that there are few scientific studies involving the development of FLOSS and HCI. Among the existing studies, most are related to usability in the development of FLOSS.

METHODOLOGY

This systematic mapping was based on the protocol model presented by Wohlin [55] and the recommendations provided by Kitchenham [28]. The protocol for this mapping is available at https://goo.gl/hAaYB6.

Research Questions

The mapping sought to answer the following research question:

> What are the methods, techniques, tools, strategies and interaction design approaches proposed or used specially for/in FLOSS development?

The main research question was organized according to the structure Population, Intervention, Comparison, Outcome, Context (PICOC) [28].

P- publications of scientific papers dealing with interaction design in FLOSS development;

I - MTTSA of interaction design proposed/used for/in FLOSS development;

C - not applicable;

O - overview of studies that consider the use of MTTSA of interaction design in FLOSS development;

C- Academic studies.

The main research question has been broken down into the following more detailed research sub-questions:

sub-RQ1 - Was the MTTSA of interaction design crafted specifically for the context of FLOSS or was an existing one used?

sub-RQ2 - Is the MTTSA related to which activity(ies) of the interaction design process?

sub-RQ3 - Has the proposed MTTSA been validated through empirical studies?

Study Selection Process

The following steps were performed in the selection of studies for systematic mapping:

1. search for studies in digital libraries;
2. selection of preliminary studies (first filter: title and abstract reading);
3. selection of studies (second filter: complete reading of selected studies in step 2);
4. extraction of data from included studies;
5. tabulation and analysis of the remaining selected studies.

Search Strategy

The digital libraries used to search for primary studies were: ACM Digital Library[1], Engineering Village[2], IEEEXplore[3], and ScienceDirect[4]. These digital libraries were chosen for indexing the majority of qualified publication sources in Computer Science, and due to the free access to papers in the major Brazilian universities.

As the research was conducted in 2017, the papers published in this year were not considered in this systematic mapping.

In this mapping, the following keywords were used: "interaction design", "design of interaction", "FLOSS", "free software", "open source". According to the keywords, the string to be used in the digital library searches was defined. The string has been defined in English only, as follows:

("interaction design" OR "design of interaction") AND (FLOSS OR "free software" OR "open source").

Data Extraction Strategy

In this step, data were extracted from each of the primary studies included in this systematic mapping according to the classification of possible predefined responses.

[1] http://dl.acm.org/
[2] http://www.engineeringvillage.com/
[3] http://ieeexplore.ieee.org/Xplore/home.jsp
[4] http://www.sciencedirect.com/

The data extraction strategy was based on Fernandez *et al.* [17]. This strategy provides a set of possible answers for each research sub-question and, therefore, ensures the application of the same extraction criteria assigned to all selected papers [17].

With regard to main research question, a paper could be classified into one or more categories:

- Method - If the paper proposes or uses at least one interaction design method for the context of FLOSS. We consider a "method" as a set of rational procedures, based on rules aimed at achieving the objectives of interaction design. [12];
- Technique - If the paper proposes or uses at least one interaction design technique for the context of FLOSS. We consider technique as a "[...] practical application of theoretical scientific knowledge to a specific field of human activity." [12, own translation];
- Tool - If the paper proposes or uses at least one interaction design tool for the context of FLOSS. We consider all tools (automated or not) that support interaction design activities.
- Approach/strategy - If the paper proposes or uses at least one approach or strategy for interaction design for the context of FLOSS. We consider approach when the authors of the paper propose or use some interaction design paradigm such as user-centered design, activity-centered design, systems design, and genius design [38]. Already strategy, it was considered a general plan or set of plans intended for interaction design [13].

With regard to sub-RQ1, a paper could be classified into one of the following categories:

- New - If the paper presents at least one new MTTSA of interaction design crafted specifically for the context of FLOSS; or
- Existing - If the paper uses at least one existing MTTSA from the interaction design field in the FLOSS development.

With regard to sub-RQ2, a paper could be classified into one or more basic activities of the interaction design process [38]:

- Establishing requirements - If the MTTSA proposed/used is related to the activity of requirements;
- Designing alternatives - If the MTTSA proposed/used is related to the attempt to satisfy the requirements and needs of the users;
- Prototyping - If the MTTSA proposed/used is related to prototyping activities;
- Evaluating - If the MTTSA proposed/used is related to evaluation of interaction design;
- Other - If the MTTSA proposed/used is related to another activity or practice of interaction design.

With regard to sub-RQ3, a paper could be classified according to the following indications:

Source	*Search DL*	*PreSt*	*SelectSt*
ACM	79	22	8
Engineering Village	55	7	1
IEEEXplore	15	2	1
ScienceDirect	168	3	1
Total	**317**	**34**	**11**

Table 1. Number of papers returned by search in digital libraries (SearchDL), selection of preliminary papers (PreSt) and selection of papers (SelectSt), classified by digital libraries

- Yes - If the author(s) validated the proposal with empirical studies; or
- No - If information about empirical validation is not provided.

Inclusion and Exclusion Criteria

For the inclusion of a paper, it was determined its relevance in relation to the research question. Papers that met the following inclusion criteria were included:

- paper presenting MTTSA of interaction design in the context of FLOSS development;
- papers published until 2016.

Papers that met at least one of the following exclusion criteria were excluded:

- introductory papers for special issues and summary of events, such as workshops;
- papers not written in English;
- duplicated papers;
- papers without free availability (especially in cases where the access is paid or not available in digital library);
- papers not published in conference or scientific journals.

RESULTS

With the application of the systematic mapping protocol, we obtained the results shown in Table 1.

Search for Papers in Digital Libraries

The application of the search string in the digital libraries returned 317 papers, separated by source, as follows: ACM Digital Library, 79 papers; Engineering Village, 55 papers; IEEEXplore, 15 papers; ScienceDirect, 168 papers.

The search in the IEEEXplore and Engineering Village digital libraries were carried out with the advanced search engine of the databases, since the basic search of each digital library returned few papers. The search for papers in the other databases was performed through basic search.

To facilitate the selection of studies, storage and management of the papers, the software Mendeley©[5] and StArt[6] were used.

[5]Mendeley©- http://www.mendeley.com

[6]StArt - http://lapes.dc.ufscar.br/tools/start_tool

Questions	**Classification**	**# of papers**	**Percentage (%)**
Research question	method	1	9.1%
	technique	2	18.1%
	tool	5	45.5%
	strategy / approach	6	54.6%
Sub-RQ1	new	8	72.7%
	existing	3	27.3%
Sub-RQ2	establishing requirements	1	9.1%
	designing alternatives	2	18.2%
	prototyping	6	54.6%
	evaluating	4	36.4%
	other	4	36.4%
Sub-RQ3	yes	5	45.5%
	no	6	54.5%

Table 2. Results of the answers for main research question and the sub-questions according to the classification of predefined responses

Selection of Preliminary Papers

In this step, carried out in January 2017, titles and abstracts of the papers found in the paper search stage were read. Based on their analysis, and according to the established inclusion and exclusion criteria, 34 papers were selected for the next step, 283 were rejected, 23 of which were duplicated.

Complete Reading of Selected Papers

After the selection of preliminary papers, the next step was to read the complete text of the selected papers. Out of 34, only 11 papers were included.

Data Extraction Results

The overall results of the answers for main research question and the sub-questions are presented in Table 2.

The overall results of the data extraction revealed that around 9% of the papers proposed/used interaction design methods for/in the FLOSS development; around 18% were related to techniques; around 45% were related to tools and around 55% were related to strategies/approaches.

Regarding the results for sub-RQ1, the majority of the papers (73%) presented at least one MTTSA of interaction design conceived in the context of FLOSS. Papers that used at least one MTTSA of existing interaction design represented around 27% of the total.

As for sub-RQ2, which deals with the interaction design activities in which the MTTSA is related, the results were as follows: around 9% of the papers were related to establishing requirements; around 18% of the papers to designing alternatives; around 55% of the papers to prototyping; around 36% to evaluating and around 36% to other activities. We observed that in other interaction design activities, all papers were related to investigating and/or supporting the participation of users in the development of FLOSS.

Considering the answers for sub-RQ3, which is related to empirical studies to validate MTTSA of interaction design

around 45% of the selected papers performed some kind of empirical study. The majority of the papers (55%) did not report on empirical studies.

The next subsections present the results related to each method, technique, tool and strategy/approach of interaction design for/in the FLOSS development in detail.

Methods of Interaction Design

The results for main research question revealed that only one of the selected papers, Lichtner *et al.* [32], have used at least one method of interaction design in the FLOSS development.

Lichtner *et al.* [32] described a case study on the use of an online user diary for remote evaluation of a software prototype. The use of open source software enabled the implementation of the online user diary for prototype evaluation. The online diary contributed to gathering data on the interaction design, system performance, and user experiences. It also enabled communication among users, researchers and system developers, giving voice to users in the evaluation and redesign processes of the software.

The evaluation was carried out collaboratively in four different organizations, in different European countries. The online diary was built with the open source phpBB software. Real-time access allowed users to be monitored and researchers were able to request input from participants when necessary or monitor problems (technical or organizational) for solutions to be found promptly.

In response for sub-RQ1, Lichtner *et al.* [32] used an existing method for the development of software. None of the selected papers proposed a method of interaction design for the development of FLOSS.

Regarding the responses for sub-RQ2, in Lichtner *et al.* [32], the method was related to the evaluating activity.

With regard to validation through empirical studies, sub-RQ3, Lichtner *et al.* [32] conducted a case study.

Techniques of Interaction Design

The results for main research question revealed that only two of the selected papers, Lichtner *et al.* [32] and Hepting *et al.* [51], mentioned the use of interaction design techniques for the development of open source software.

Lichtner *et al.* [32] presented a case study on the use of a diary in real world system evaluation by a geographically distributed team and different cultures and practices. The techniques used were field visits, interviews, observation, focus group, scenario walkthroughs and questionnaire. Hepting *et al.* [51] discussed the experience of offering a course in the HCI area with FLOSS development projects. The techniques used were scenario-based design, questionnaire, and prototyping.

Regarding sub-RQ1, Lichtner *et al.* [32] and Hepting *et al.* [51] used pre-existing techniques.

Regarding the answers for sub-RQ2, Lichtner *et al.* [32] used the techniques in the evaluating activity. Hepting *et al.* [51] used the techniques in the activities of establishing requirements, designing alternatives, prototyping and evaluating.

With regard to validation through empirical studies, sub-RQ3, only Lichtner *et al.* [32] presented information on empirical studies: a case study.

Tools of Interaction Design

The results for the main research question revealed that around 45% of the papers proposed/used at least one tool of interaction design for/in the development of FLOSS.

The authors who proposed/used at least one tool of interaction design for/in the development of FLOSS are Lichtner *et al.* [32], Lawson *et al.* [29], Li *et al.* [31], Hansen *et al.* [20] and Mcgee-Lennon *et al.* [35].

While Lichtner *et al.* [32] used the open source phpBB tool to perform the interaction design activities, Lawson *et al.* [29], Li *et al.* [31], Hansen *et al.* [20] and Mcgee-Lennon *et al.* [35], respectively, developed the following tools: SKEMMI, software workbench for supporting the effective and dynamic prototyping of multimodal interactive systems; ActivityStudio, an Open source toolset for prototyping and in-situ testing of ubiquitous computing application prototypes; PyMT, a multi-touch interface toolkit for the Python programming language; and Open Interface Development Environment, a rapid prototyping tool that is embedded in an open source framework for the rapid development of multimodal interactive systems.

In response for sub-RQ1, Lichtner *et al.* [32] used existing open source software tool for software development. Lawson *et al.* [29], Li *et al.* [31], Hansen *et al.* [20] and Mcgee-Lennon *et al.* [35] proposed new software prototyping tools.

Regarding the responses for sub-RQ2, the existing tool was used by Lichtner *et al.* [32] in the prototyping activity of the interaction design process. The tool proposed by Li *et al.* [31] can be used in prototyping and evaluating activities. The tools proposed by Lawson *et al.* [29], Hansen *et al.* [20] and Mcgee-Lennon *et al.* [35] can be used in the prototyping activity.

With regard to the validation of the tools through empirical studies, sub-RQ3, the SKEMMI tool of Lawson *et al.* [29], was evaluated using a group of six experts (programmers, application designers). Lichtner *et al.* [32] and Li *et al.* [31] carried out a case study and Mcgee-Lennon *et al.* [35] carried out a study of users.

Strategies/Approaches of Interaction Design

Interaction design strategies/approaches for FLOSS development were found in around 55% of selected papers.

The authors who proposed at least one interaction design strategy/approach for the development of FLOSS were Hepting *et al.* [25], Jetter *et al.* [26], Cheung *et al.* [9], Bach *et al.* [6], Barcellini *et al.* [7] and Ebenreuter [16].

Hepting *et al.* [25] defined a strategy to offer a usability course with open source software projects. With the experience of combining the usability course with open source software design, the authors proposed a framework for course organization.

Jetter *et al.* [26] presented the zoomable object-oriented information landscape, a new approach to interaction design, and

an open source software framework for post-WIMP distributed user interfaces in interactive spaces.

The study by Cheung *et al.* [9] identified use cases related to enabling and facilitating user participation in the development of open source software.

Bach *et al.* [6] addressed the reflective practitioner to consider how end users can directly contribute to design (interaction) discussions in open source software projects.

Barcellini *et al.* [7] examined a "push-by-users" design proposal through discussions that took place on two discussion lists, one user-oriented and the other developer-oriented. The study provided insights on how design and usage are articulated in open source software projects.

Ebenreuter [16] proposed the extension of the concept of open source development to collaborative interaction design practices. For this, the author presented an alternative approach for the presentation of design information as rationale instances. The author also discussed the usefulness of rationale instances as a raw material for collaboration in open design projects.

Regarding the answers for sub-RQ1, new approaches or strategies were proposed by Hepting *et al.* [25], Jetter *et al.* [26], Cheung *et al.* [9] and Ebenreuter [16]. Bach *et al.* [6] and Barcellini *et al.* [7] used existing approaches with the purpose of investigating users' participation in mailing lists for open source software projects.

According to the answers for sub-RQ2, we note that the framework proposed by Hepting *et al.* [25] contemplated the following interaction design activities: establishing requirements, designing alternatives, prototyping, and evaluating. The approach proposed by Jetter *et al.* [26] contemplated the activities of designing alternatives, prototyping and evaluating. The proposals of Cheung *et al.* [9] and Ebenreuter [16] included, respectively, user participation in the open source software development process and the provision of interaction design documentation to improve collaboration in open design projects. In Bach *et al.* [6] and Barcellini *et al.* [7], the approaches were used in other activities involving investigating and/or supporting the participation of users in the FLOSS development process.

With regard to validation by means of empirical studies, sub-RQ3, only Jetter *et al.* [26] presented information on empirical studies, which they used in an approach to usability evaluation based on conceptual maps with designers and developers.

ANALYSIS

This section summarizes the main considerations and discussions from this systematic mapping, highlighting the limitations that may pose threats to its validity.

Main Findings and Discussions

The main findings of this mapping are the following ones:

- there are few studies on MTTSA of interaction design proposed or used for/in FLOSS development;
- methods of interaction design proposed specifically for the development of FLOSS were not found; the studies found used existing methods of interaction design in the context of FLOSS;
- techniques of interaction design, proposed specifically for the development of FLOSS, were not found; one of the selected papers, Lichtner *et al.* [32], used pre-existing techniques and did not consider the distributed development environment of FLOSS;
- the principal interest of the selected studies is in the activities of prototyping and evaluating; few studies have addressed the activities of establishing requirements and designing alternatives;
- the majority of the selected studies do not present any type of validation through empirical studies.

The results of this systematic mapping suggest the need for broad support for FLOSS projects and communities by the HCI community, through research efforts in the area of interaction design for the availability of MTTSA of interaction design considering the characteristics of FLOSS development. Therefore, it is necessary to develop and publish research on interaction design in the context of FLOSS.

On the other hand, the studies selected in this mapping show that researchers are aware of the need for scientific investment in interaction design in FLOSS development. We can also mention that there is evidence that FLOSS can contribute to the HCI, such as FLOSS tools that can be used in prototyping and evaluation activities in the interaction design process.

Despite the fact that there were few studies, successful results are verified when using MTTSA of interaction design in the development of FLOSS. There are several contributions found in selected papers, including:

- the active participation of end users in the FLOSS development process;
- greater concern with users;
- improvements in communication between users and developers;
- the use of distributed development environments and FLOSS for the interaction design process;
- proposal of source code customization with HCI principles to facilitate and improve understanding of the source code by inexperienced programmers;
- encouraging the use of FLOSS development strategy with HCI disciplines in the academic environment;
- contribution of FLOSS in the design process of interaction of software projects, both for the development of FLOSS, and for the development of proprietary software.

The only paper selected in this systematic mapping that reported the use of existing method and techniques for interaction design in the development of FLOSS was Lichtner *et al.* [32]. Lichtner *et al.* [32] cite that "This paper reports and reflects on one of the methods used for the formative evaluation of this prototype: the online diary.". The main contribution is

in the use of a method that made possible the remote interaction between users and developers in a FLOSS development project. Data collect and redesign evaluation were performed remotely through the online user diary with the phpBB tool. User participation in the redesign process has become possible, although the development team and users are geographically distributed. Lichtner *et al.* [32] also used interaction design techniques for the prototype evaluation process. Unlike the interaction design method, which was used remotely, interaction design techniques were used in a co-localized way. The developers visited the companies where the users worked and applied the techniques of field visits, interviews and observation. A tool for interaction design, phpBB, was used to implement the online diary method. Although it addresses only the evaluation activity of the interaction design process, Lichtner *et al.* [32] showed that it is possible to apply good interaction design practices in the development of FLOSS.

During the execution of this mapping, we found several software development initiatives [19, 21, 22, 24, 56] that used some FLOSS product combined with some interaction design practice. At first, it was selected these papers for complete reading. When these papers were read in full, we decided not to include them because the software were not developed in the context of FLOSS. Despite major interaction design contributions in distributed software development, such studies are outside the scope of this systematic mapping.

The results of this systematic mapping are expected to provide researchers and FLOSS community with information on MTTSA of interaction design for the development of FLOSS. The gaps and lack of studies on this subject may suggest that further research to be undertaken.

Threats to Validity

In order to guarantee an impartial selection process, all the steps of the systematic mapping process were followed according to the protocol. The authors independently selected the studies and, in the end, eventual disagreements and doubts were corrected. The digital libraries used include most of the main journals and conferences in the area of Computer Science. However, due to the limited number of digital libraries used, it is possible that relevant studies have not been included.

CONCLUSIONS

This paper presented a systematic mapping that summarizes the existing information about MTTSA of interaction design that have been proposed or used by researchers in the context of FLOSS development. Eleven primary papers were selected in this systematic mapping that were classified into four main categories: method, technique, tool and strategy/approach.

This systematic mapping identified few studies that proposed/used MTTSA of interaction design in the context of FLOSS. Despite this fact, the selected studies were successful and brought important contributions by applying MTTSA of interaction design in the development of FLOSS.

As future work, we intend to review some of the concepts related to ISO 9241-210 [14] and participatory design to reflect on the main principles and how they were addressed or not in the papers surveyed. Moreover, with the result obtained in this systematic mapping, the gaps and the lack of studies involving the areas of interaction design and development of FLOSS, we intend to advance in the research on this subject. Therefore, the next step of our research will be to expand this systematic mapping to identify approaches, methods, techniques, and tools for participatory interaction design in distributed software development environments. With this, we intend to develop a participatory interaction design process model for distributed software development environments. Finally, we must extend the Open Source Maturity Model [54] with the proposed interaction design model. Considering the inherent advantages of software development following a software process model, we believe that interaction design activities will be considered during the different stages of the FLOSS development process, particularly in the early stages.

Finally, we emphasize on the relevance of the theme and highlight the importance of the integration of techniques, methods and approaches interaction design to the development of FLOSS. This mapping is expected to provide an overview of the studies that have proposed/used MTTSA of interaction design in the context of FLOSS, and it is an initial effort for new researches and proposals related to this topic to be carried out.

ACKNOWLEDGMENTS

We thank the financial support of the Foundation for Research Support of the State of Bahia (FAPESB). We also thank the Computer Science Graduate Program (PGCOMP)/UFBA and the Research and Extension Group in Informatics, Education and Society Onda Digital/UFBA.

REFERENCES

1. Daniel D. Alves, Maria I. Cagnin, and Débora M. B. Paiva. 2014. Accessibility in development of free software projects. In *Computing Conference (CLEI), 2014 XL Latin American*. IEEE, 1–12.
2. Morten S. Andreasen, Henrik V. Nielsen, Simon O. Schrøder, and Jan Stage. 2015. Usability in open source software development: Opinions and practice. *Information technology and control* 35, 3 (2015).
3. Interaction Design Association. 2017. About & History - Interaction design definition. (2017). Retrieved February 02, 2017 from http://ixda.org/ixda-global/about-history.
4. Paula M. Bach, Bryan Kirschner, and John M. Carroll. 2007. Usability and free/libre/open source software SIG: HCI expertise and design rationale. In *CHI'07 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2097–2100.
5. Paula M. Bach and Michael Terry. 2010. The future of FLOSS in CHI research and practice. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*. ACM, 4473–4476.
6. Paula M. Bach and Michael Twidale. 2010. Involving reflective users in design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2037–2040.

7. Flore Barcellini, Françoise Détienne, and Jean-Marie Burkhardt. 2008. User and developer mediation in an Open Source Software community: Boundary spanning through cross participation in online discussions. *International Journal of Human-Computer Studies* 66, 7 (2008), 558–570.
8. Andrea Bonaccorsi and Cristina Rossi. 2003. Why open source software can succeed. *Research policy* 32, 7 (2003), 1243–1258.
9. Gifford Cheung, Parmit Chilana, Shaun Kane, and Braden Pellett. 2009. Designing for discovery: opening the hood for open-source end user tinkering. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. ACM, 4321–4326.
10. Joelma Choma. 2015. *ScrumUX: uma abordagem para integrar design de interação do usuário ao processo Scrum*. Ph.D. Dissertation. PhD Dissertation, Federal University of São Carlos.
11. Kevin Crowston, Kangning Wei, James Howison, and Andrea Wiggins. 2012. Free/Libre open-source software development: What we know and what we do not know. *ACM Computing Surveys (CSUR)* 44, 2 (2012), 7.
12. Dicionário Básico de Filosofia. 1996. Japiassú, Hilton and Marcondes, Danilo. (1996).
13. Collins Dictionaries. 2017. English Dictionary. *Retrieved January 11, 2017 from https://www.collinsdictionary.com/dictionary/english/strategy* (2017).
14. ISO DIS. 2009. 9241-210: 2010. Ergonomics of human system interaction-Part 210: Human-centred design for interactive systems. *International Standardization Organization (ISO). Switzerland* (2009).
15. Ana P. O. dos Santos. 2012. *Aplicação de práticas de usabilidade ágil em software livre*. Ph.D. Dissertation. PhD Dissertation, Universidade de São Paulo.
16. Natalie Ebenreuter. 2009. Working towards an open source design approach for the development of collaborative design projects. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7*. ACM, 285–288.
17. Adrian Fernandez, Emilio Insfran, and Silvia Abrahão. 2011. Usability evaluation methods for the web: A systematic mapping study. *Information and Software Technology* 53, 8 (2011), 789–817.
18. Bernard Golden. 2008. Open Source Maturity Model. *The Open Source Business Resource* (2008), 4.
19. Collin Green, Irene Tollinger, Christian Ratterman, Guy Pyrzak, Alex Eiser, Lanie Castro, and Alonso Vera. 2009. Leveraging open-source software in the design and development process. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. ACM, 3061–3074.
20. Thomas E. Hansen, Juan Pablo Hourcade, Mathieu Virbel, Sharath Patali, and Tiago Serra. 2009. PyMT: a post-WIMP multi-touch user interface toolkit. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. ACM, 17–24.
21. Lisa Haskel. 2014. Bridging the gaps: building on experience with participatory design. In *Proceedings of the 13th Participatory Design Conference: Short Papers, Industry Cases, Workshop Descriptions, Doctoral Consortium papers, and Keynote abstracts-Volume 2*. ACM, 233–236.
22. Lisa Haskel and Paula Graham. 2014. Hublink: a case study in participatory design and open source in the third sector. In *Proceedings of the 13th Participatory Design Conference: Short Papers, Industry Cases, Workshop Descriptions, Doctoral Consortium papers, and Keynote abstracts-Volume 2*. ACM, 129–132.
23. Henrik Hedberg and Netta Iivari. 2009. Integrating HCI specialists into open source software development projects. In *IFIP International Conference on Open Source Systems*. Springer, 251–263.
24. David G. Hendry. 2008. Public participation in proprietary software development through user roles and discourse. *International Journal of Human-Computer Studies* 66, 7 (2008), 545–557.
25. Daryl H. Hepting, Lijuan Peng, Timothy J. Maciag, David Gerhard, and Brien Maguire. 2008. Creating synergy between usability courses and open source software projects. *ACM SIGCSE Bulletin* 40, 2 (2008), 120–123.
26. Hans-Christian Jetter, Michael Zöllner, Jens Gerken, and Harald Reiterer. 2012. Design and implementation of post-WIMP distributed user interfaces with ZOIL. *International Journal of Human-Computer Interaction* 28, 11 (2012), 737–747.
27. Kim Johnson. 2001. *A descriptive process model for open-source software development*. PhD thesis, University of Calgary.
28. Barbara Kitchenham and Stuart Charters. 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering. In *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*.
29. Jean-Yves L. Lawson, Ahmad-Amr Al-Akkad, Jean Vanderdonckt, and Benoit Macq. 2009. An open source workbench for prototyping multimodal interactions based on off-the-shelf heterogeneous components. In *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems*. ACM, 245–254.
30. Josh Lerner and Jean Tirole. 2002. Some simple economics of open source. *The journal of industrial economics* 50, 2 (2002), 197–234.
31. Yangi Li and James A Landay. 2008. Into the wild: low-cost ubicomp prototype testing. *Computer* 41, 6 (2008).

32. Valentina Lichtner, Angela P. Kounkou, Amir Dotan, José P Kooken, and Neil AM Maiden. 2009. An online forum as a user diary for remote workplace evaluation of a work-integrated learning system. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2955–2970.

33. John W. C. Llanos and booktitle=International Conference on Product Focused Software Process Improvement pages=131–144 year=2012 organization=Springer Castillo, Silvia T. A. Differences between traditional and open source development activities.

34. Jonas Lowgren. 2013. *Interaction Design - brief intro In The Encyclopedia of Human-Computer Interaction*. The Interaction Design Foundation.

35. Marilyn R. McGee-Lennon, Andrew Ramsay, David McGookin, and Philip Gray. 2009. User evaluation of OIDE: a rapid prototyping platform for multimodal interaction. In *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems*. ACM, 237–242.

36. David Nichols and Michael Twidale. 2003. The usability of open source software. *First Monday* 8, 1 (2003).

37. Vidyasagar Potdar and Elizabeth Chang. 2004. Open source and closed source software development methodologies. In *26th International Conference on Software Engineering*. IET, 105–109.

38. Jenny Preece, Helen Sharp, and Yvonne Rogers. 2015. *Interaction design: beyond human-computer interaction* (4 ed.). Wiley.

39. Nicholas P. Radtke, Marco A. Janssen, and James S. Collofello. 2009. What makes Free/Libre Open Source Software (FLOSS) projects successful? An agent-based model of FLOSS projects. *International Journal of Open Source Software and Processes (IJOSSP)* 1, 2 (2009), 1–13.

40. Eric Raymond. 1999. The cathedral and the bazaar. *Philosophy & Technology* 12, 3 (1999), 23.

41. Christian R. Reis and Renata P. de M. Fortes. 2003. *Caracterizaç ao de um Processo de Software para Projetos de Software Livre*. Ph.D. Dissertation. PhD thesis, University of Sao Paulo, Brazil.

42. Walt Scacchi, Chris Jensen, John Noll, and Margaret Elliott. 2005. Multi-modal modeling of open source software requirements processes. In *First International Conference on Open Source Systems, Genova, Italy*. Citeseer, 1–8.

43. Daniel Schwartz and Allen Gunn. 2009. Integrating user experience into free/libre open source software: CHI 2009 special interest group. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2739–2742.

44. Raphaël Semeteys. 2008. Method for qualification and selection of open source software. *Open Source Business Resource* May 2008 (2008).

45. Bruno C. C. Silva and Ricardo de A. Falbo. 2006. Definição de um Processo Padrão para Software Livre. *V Simpósio Brasileiro de Qualidade de Software–SBQS* (2006), 159–173.

46. Tiago S. da Silva. 2012. *A framework for integrating interaction design and agile methods*. Ph.D. Dissertation. PhD thesis, Pontifical Catholic University of Rio Grande do Sul, Brazil.

47. Richard Stallman. 1998. Why free software is better than open source. (1998). Retrieved February 06, 2017 from http://www.fsf.org/philosophy/free-software-for-freedom.html.

48. Richard Stallman. 2002. Free software, free society: Selected essays of Richard M. Stallman. (2002). Retrieved January 05, 2017 from https://www.gnu.org/philosophy/fsfs/rms-essays.pdf.

49. Ramanath Subramanyam and Mu Xia. 2008. Free/Libre Open Source Software development in developing and developed countries: A conceptual framework with an exploratory study. *Decision support systems* 46, 1 (2008), 173–186.

50. Yuhong Tian. 2006. *Developing an open source software development process model using grounded theory*. University of Nebraska at Lincoln.

51. Michael B. Twidale and David M. Nichols. 2005. Exploring usability discussions in open source development. In *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*. IEEE, 198c–198c.

52. Anthony I. Wasserman, Murugan Pal, and Christopher Chan. 2006. The Business Readiness Rating: a framework for evaluating open source. *EFOSS-Evaluation Framework for Open Source Software* (2006).

53. Steve Whittaker. 2013. Interaction design: what we know and what we need to know. *interactions* 20, 4 (2013), 38–42.

54. Marion Wittmann, Ranga Nambakam, Felipe Ortega, Viviane Malheiros, and Davide Tosi. 2009. Working Document WD6. 3.1-CMM-like model for OSS, v. 1. (2009).

55. Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media.

56. Marcelo Worsley, Michael Johnston, and Paulo Blikstein. 2011. OpenGesture: a low-cost authoring framework for gesture and speech based application development and learning analytics. In *Proceedings of the 10th International Conference on Interaction Design and Children*. ACM, 254–256.

57. Didar Zowghi. 2002. Does global software development need a different requirements engineering process. In *Proceedings of International Workshop on Global Software Development*, Vol. 2002. Citeseer.