

2-2012

# Users' Perception of Open Source Usability: An Empirical Study

Arif Raza

*National University of Science and Technology - Pakistan, arif\_raza@mcs.edu.pk*

Luiz Fernando Capretz

*University of Western Ontario, lcapretz@uwo.ca*

Faheem Ahmed

*Thompson River University, fahmed@tru.ca*

Follow this and additional works at: <https://ir.lib.uwo.ca/electricalpub>

 Part of the [Software Engineering Commons](#)

---

## Citation of this paper:

@article{DBLP:journals/ewc/RazaCA12, author = {Arif Raza and Luiz Fernando Capretz and Faheem Ahmed}, title = {Users' perception of open source usability: an empirical study}, journal = {Eng. Comput. (Lond.)}, volume = {28}, number = {2}, year = {2012}, pages = {109-121}, ee = {http://dx.doi.org/10.1007/s00366-011-0222-1}, bibsource = {DBLP, http://dblp.uni-trier.de}}

# Users' Perception of Open Source Usability: An Empirical Study

Arif Raza

*Department of Electrical & Computer Engineering, University of Western Ontario, London, Ontario, Canada N6A 5B9*  
 araza7@uwo.ca, 1 226 268 1054 (Tel), 1 519 850 2436 (Fax)

Luiz Fernando Capretz

*Department of Electrical & Computer Engineering, University of Western Ontario, London, Ontario, Canada N6A 5B9*

Faheem Ahmed

*Faculty of Information Technology, United Arab Emirates University*  
 P. O. Box 17551, Al Ain, United Arab Emirates

**Abstract**— The number of Open Source Software (OSS) users has increased in recent years. No longer are they limited to technically adept software developers. Many believe that the OSS market share could increase tremendously provided OSS had systems that were easier to use. Although examples of good usable open source software exist, it is agreed that OSS can be made more usable. This study presents an empirical investigation to study the impact of some key factors on OSS usability from the end users' point of view. The research model studies and establishes the relationship between the key usability factors from the users' perspective and OSS usability. A data set of 102 OSS users from 13 open source projects of various sizes was used to study the research model. The results of this study provide empirical evidence by indicating that the highlighted key factors play a significant role in improving OSS usability.

**Index Terms** — Usability testing, Software Quality, Statistical methods, User issues.

----- Φ -----

## 1 INTRODUCTION

Software plays an ever-increasing role in our society. However, software systems often fail to deliver as promised. Usability concerns need to be addressed in many of the software systems that we use every day. Open source software allows its users to use, inspect, modify and distribute it in modified or unmodified form to others [1]. von Krogh and Spaeth propose that such an open license is one of the major reasons for OSS's existence [2]. They believe that to get the best out of open source phenomena, *"information systems research should remain open to dialog with other areas and disciplines."* Levesque argues that for OSS to be widely accepted it must address issues like user interface design, documentation, feature-centric development, self-programming and *"religious blindness"* [3]. Polancic et al., however, question the quality of OSS projects [4]. They propose an assessment model to evaluate an OSS product. The empirical study of Paulson et al. supports the belief that defects are found and fixed more rapidly in OSS as compared to closed source software products [5].

Although it is not presumed that every OSS has a bad interface, it is believed there are issues related to usability in OSS. In the words of Nichols and Twidale, *"the existence of a problem does not necessarily mean that all OSS interfaces are bad or that OSS is doomed to have hard to use interfaces, just a recognition that the interfaces ought to be and can be made better"* [6]. Hedberg et al. also identify that due to the constant growth in number of novice / non-technical users of OSS, its usability needs to

be improved [7]. According to Çetin and Göktürk, OSS usability is a multidimensional problem area particularly due to the fact that usability is not a prime goal of OSS projects, OSS developers are not aware of the importance of usability and users' requirements and there is a lack of interaction between developers and the Human Computer Interaction (HCI) community [8].

This research work contributes to understanding the effects of some key usability factors on OSS usability through an empirical investigation. A quantitative survey of OSS users of different OSS projects is conducted and is reported here. The survey is to analyze the conceptual model and the hypotheses of the study. The results provide evidence that the stated key factors play an important role towards OSS usability.

The literature review that motivates this research work is presented in the next section. It also helps in the selection of the key factors for the study. Section 3 illustrates the research model and the hypotheses of this study. Section 4 explains the research methodology, data collection process and the experimental setup in its first part, reliability and validity analysis of the measuring instrument in the second part and data analysis procedures in its third part. In Section 5, the hypotheses are tested and the results are analyzed followed by the discussion of results in Section 6 that also includes the limitations of the study. Finally, the paper concludes in Section 7.

## **2. LITERATURE REVIEW**

### **2.1 Open Source Software Quality – In General**

Porter et al. observe the inconsistency in open source software mainly due to "*short feedback loops between users and core developers*" [9]. They feel that the frequent release of beta versions of the software, although it may satisfy some users, does frustrate many others who would like more stable software. They also identify unsystematic and undocumented software testing in OSS. Yang and Wang consider free and open source software as a reliable fighting force against the monopoly of proprietary software [10]. They believe that its popularity will ultimately reduce users' dependency on proprietary software. Stol et al. categorize OSS as a suitable field for research, in particular for empirical studies due to easy and freely available data through sourceforge.net, freshmeat.com and mailing lists [11]. However they identify the need to systematically review these empirical analyses and their results. Stol et al. have conducted a systemic review of OSS related empirical research and have presented their results. Ferenc et al. realize the importance of OSS in the software industry and emphasize the need to measure the quality and reliability of OSS code through the use of proper tools [12]. They present a toolset to extract facts that are used to calculate object oriented metrics of real-world software. Hedgebeth considers OSS a valuable collaborative source of knowledge management [13]. He highlights different misconceptions about open source such as "*security concerns, a lack of a customer support apparatus, and a perceived inability that businesses*

*cannot make a profit using open source technology*" and asserts that OSS is here to stay. This fact is now acknowledged even by big commercial software organizations like Microsoft, IBM, and Apple etc.

Golden et al. have come up with a Usability-Supporting Architectural Pattern (USAP) that supports specific usability issues at the architectural level [14]. They observe that usability concerns could be better addressed if "*implications of usability heuristics for software design*" are made clear and explicit to the software designers. Nakagawa et al. also believe in the direct relationship between software architecture and OSS quality [15]. They point out to OSS developers that architectural knowledge, styles, patterns and evaluation methods should be applied in order to achieve high quality and success of OSS projects.

Stamelos et al.'s empirical study concludes that user satisfaction is related to the average size of components in an open source application [16]. They also support the idea of OSS having its own quality standards. According to Feller and Fitzgerald, OSS users have traditionally been the developers, testers and documenters themselves or the experts in the field resulting in an overlap of developers and users [17]. However, with the entrance of OSS into the mainstream and the interest and support of big corporate players like IBM, Apple and Oracle, non-expert users are attracted to OSS. No cost (generally), high quality and support have been the motivating forces as well.

Analyzing the effects of different development practices on product quality in OSS, Koch and Neumann found a significant relationship between process attributes and product quality [18]. Gyimothy et al. underline the need to study the quality and reliability of OSS due to non-conventional development and management methodologies employed by the OSS community [19]. They have come up with a toolset to calculate metrics from C++ source code of real-world software. Ferenc et al. are curious about the quality and reliability of OSS as it is developed outside the controlled environment and without proper company management [12]. Mansfield-Devine, however has a different viewpoint. He states, "*By monitoring the development of an OSS project, joining mailing lists, seeing how quickly issues are fixed, how often releases are made and so on, users can verify how well the project is being managed*" [20].

Samoladas et. al. present a hierarchical quality model SQO-OSS to evaluate source code and related processes in open source software [21]. The model supports an automated software evaluation system and is based on calculation of metric values. The authors state, "*Our model evaluates all aspects of OSS development, both the product (code) and the community.*" The evaluation process also provides a profile based evaluation algorithm.

Golden proposes open source maturity model (OSMM) and relates the OSS quality to its maturity by considering "*Product software, Support, Documentation, Training, Product integrations and Professional services*" [22]. Each of these factors is evaluated to find an accumulated score. Although OSMM is simple and thus easy to apply, it overlooks some important

software parameters, such as the source code.

In the last decade, two well-known methodologies have emerged namely, Open Business Readiness Rating (OpenBRR) by Carnegie Mellon West and Intel [23] and Qualification and Selection Open Source (QSOS) sponsored by Atos Origin [24]. Deprez and Alexandre have performed a detailed comparative study of these two assessment methodologies on their *"overall approaches, their scoring procedures and their evaluation criteria"* [25]. However their comparison is based on the description of the methodologies and not on their empirical application.

Çetin and Göktürk have proposed a metric model using literature research and survey findings to measure and analyze the usability of an OSS project. However, no validation of the proposed metrics has been presented, as they state that *"the main lacking part of this paper is that no validation of the proposed metrics has been done. It's required to apply the provided metrics to various F/OSS projects for which their usability is known to a specific extent. Another method could be to measure the usability of these applications and try to find a correlation between them."*

## **2.2 Usability Factors: Literature Review of Concepts**

In the ISO 9241-11 Standard, usability is defined as *"the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use"* [26]. The International Organization for Standardization and The International Electro Technical Commission ISO/IEC 9126-1 places software quality attributes into six categories namely functionality, reliability, usability, efficiency, maintainability and portability [27]. In the standard, usability is defined as *"The capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions."*

While examining usability practices in OSS, Nichols and Twidale highlight the need to address usability issues more arduously [28]. These authors discuss the failure of certain commercial closed source software projects, which result from unusable systems or poor handling usability issues. Specifically, they believe that these failures are an indication of usability being an *"unresolved"* issue even in proprietary software, which is considered more usable than OSS, as closed source software is more mature than OSS, and it is equipped with more resources, both in terms of experienced manpower and financial resources. Aberdour contrasts the *"formal and structured testing"* that is typical in closed software development with the *"unstructured and informal testing"* in OSS development [29]. On the other hand, Hedberg et al. maintain that the processes of *"test coverage, test-driven development and testing performed by developers"* require more attention in OSS projects through formal and detailed test plans that ensure errors are caught before the release of the software [7].

Sampson criticizes OSS usability by wondering whether OSS developers ever consult a usability expert or explore any of

the standards [30]. He refers to the same old dilemma of OSS developers who consider themselves as end users of their product, *"too often the team presumes that all the skills required are already present among team members."* Twidale and Nichols while exploring usability related issues in open source development conclude that *"Scarcity of expertise, Bug reporting and classification, and Heterogeneity in usability discussions"* are the areas that need to be focused on by the OSS developers [31]. The more that the users' expectations and requirements are addressed in open source software, the more acceptances it will receive. Bodker et al. believe that there is a gap between the OSS *"developer-users"* and their potential users [32]. They posit that enabling the employees of organizations and institutions to come up with the right requirements of OSS systems can be a way *"to avoid developer-centric systems that perform poorly in terms of real-world usability."* Nichols and Twidale also highlight few related problems in this scenario such as the traditional approach by OSS developers to develop software for an *"elite"* class of technically adept users [6]. However, they observe that with the increasing involvement of non-technical users in the OSS community, software developers are starting to realize the importance of usability and hence the need to ensure that new users find their products usable and adaptable. It is time for OSS architects, designers and developers to realize that they are not the ultimate users of their applications.

Zhao and Deek highlight the problem of reporting bugs by OSS users [33]. They observe that when an average user wants to report an error, particularly related to usability, s/he does not know how to do it effectively. Observing a similar problem, Nichols and Twidale identify the difficulties faced by the users in reporting usability bugs as *"difficulties that a user may experience with a Graphical User Interface may not be easy to describe textually"* [28]. They observe that there is a bias in treating usability bugs as compared to functionality bugs. Usability issues, as expected, are more subjective in nature and more debatable. As an example, a user interface (UI) element may be more confusing to some people and less to others. Such issues could prolong the discussion of analyzing and fixing usability bugs. Cetin et al. observe that due to *"the lack of a suitable usability reporting interface"* usability issues are reported less than when they really exist [34]. It is proposed that effective feedback from end users can be one of the ways to improve OSS usability. This can be achieved by providing them with an easy and convenient way to report the errors they encounter while using the software.

Interactive help features in software that addresses users' problems dynamically can be a step toward addressing the usability problem. However to provide help, it is necessary to know exactly what sort of help is actually expected and may be asked for. Furthermore, such help features need to be designed to provide help to all sorts of users, not only to the expert ones. This obviously is not an easy task as Shneiderman highlights that designing software for any expert computer user is difficult as it is, let alone designing for anyone to use [35]. He states that lowering the cost of hardware and computer accessories is providing access for more people but interface and information design has to play its role. Viorres

et al. argue that the majority of disabled users prefer to use proprietary software due to better accessibility along with assistive technology, even though OSS claims to have the *"right to access for all"* [36]. While discussing the user's problems, it should not be forgotten that there are users having problems or the elderly or children who are novices to the computer technology.

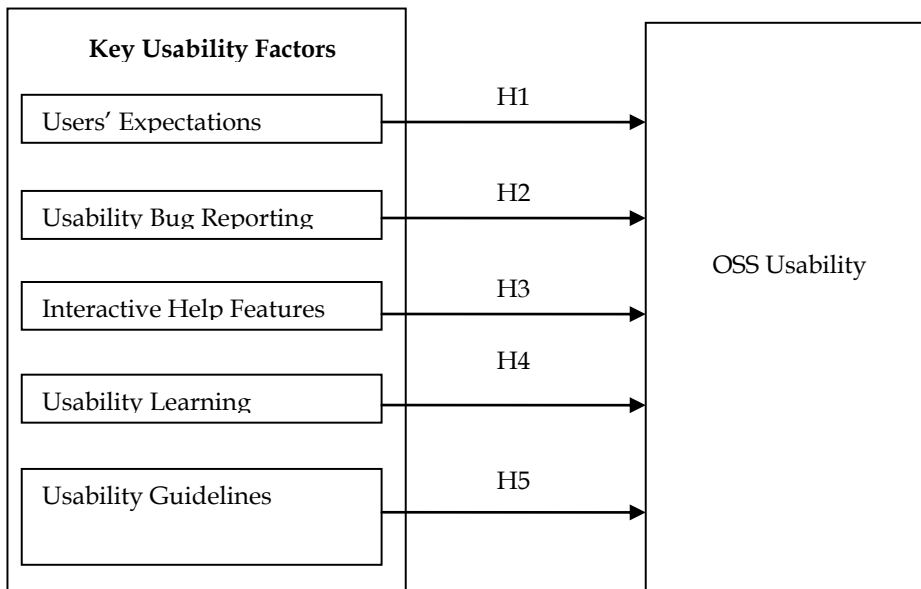
Formal usability learning by software architects, designers and developers can be an acknowledgement of the problem as well as a part of the solution. Practice of HCI is commonly referred to as Usability Engineering. To better understand the user's point of view, software developers need to learn HCI and usability principles. Rusu et al. highlight the importance of HCI education for software professionals [37]. They identify that in computer science (CS) design courses, HCI issues are generally given a secondary level importance. According to Faulkner and Culwin, HCI and Software Engineering educators are usually in different camps [38]. The authors feel that there is a need for more interaction between HCI and SE. According to them, HCI should be adopted as the underlying principle to systems development. Zhao and Deek however suggest that OSS users need to be trained too so that they are able to do usability inspections in an effective and efficient way [33].

Hedberg et al. propose the incorporation of usability guidelines, active participation of usability experts in OSS projects, usability testing and bug reporting [7]. In-depth empirical research is needed to understand the challenges related to usability and quality assurance in OSS. Nichols and Twidale refer to Human Interface Guidelines (HIGs) that can not only prevent confusion about usability issues but can also be considered as an authority on what shall be done [28]. Çetin and Göktürk also identify that there is no consensus of usability guidelines for OSS developers from usability experts [8]. Iivari et al. believe that the growing user population of OSS is more interested in usable systems than in their development [39]. They recommend the involvement of HCI experts in OSS development. Therefore it is first necessary to understand how usability issues are practiced in OSS environments and then to come up with a method to evaluate and measure usability of OSS projects.

Benson et al. call usable software *"a win-win situation for developers, the corporations, and – most importantly – the users"* [40]. Folmer and Bosch in their survey about usability engineering practices identify that usability issues are typically discovered late in the software development process and hence are expensive to implement [41]. They advocate that *"usability should drive design at all stages"* and assert that usability issues can best be addressed if incorporated at the architectural design level. Hedberg et al. believe that high quality and usability can be ensured in OSS through proven methods and processes, however there is a need to find ways to adapt user centered design methods that can fit in a distributed environment of OSS development [7].

### 3. RESEARCH MODEL AND THE HYPOTHESES

This work presents a research model for analyzing the relationship between key usability factors and open source software usability, as shown in Figure 1. The model derives its theoretical foundations by combining previous work in OSS usability, SE and HCI; it includes five key usability factors: *Users' Expectations*, *Usability Bug Reporting*, *Interactive Help Features*, *Usability Learning* and *Usability Guidelines*. The dependent variable of this study is *OSS Usability*, and the five independent variables are referred to as "*Usability Factors*" hereafter.



**Fig. 1 Research Model**

Overall, the objective of this study is to investigate the answer to the following question:

*“Do key usability factors have an impact on OSS usability from the perspective of end users?”*

The multiple linear regression equation of the model is as follows:

$$\text{OSS Usability} = f_0 + f_1v_1 + f_2v_2 + f_3v_3 + f_4v_4 + f_5v_5 \quad (1)$$

where  $f_0, f_1, f_2, f_3, f_4$  and  $f_5$  are the coefficients and  $v_1, v_2, v_3, v_4$  and  $v_5$  are the five independent variables. In order to empirically investigate the research question, the five hypotheses are derived as presented below:

- H1:** The software developers' understanding of user expectations and requirements is positively related to improving usability in OSS.
- H2:** Convenient usability bug reporting has a positive impact on usability in OSS.
- H3:** Interactive help features in software have a positive impact on usability in OSS.



**H4:** The designers' knowledge of usability and user-centered design methods is positively related to improved software.

**H5:** Usability guidelines for the developers help to improve OSS usability.

#### 4. RESEARCH METHODOLOGY

Open source software projects deal with different categories of applications, such as Database, Desktop Environment, Education, Finance, Games / Entertainment, and Networking. In order to collect the data, we sent emails from users' mailing lists to OSS users of 13 different projects on sourceforge.net. The projects differed in size and ranged from small-scale to large-scale. The questionnaires, which are presented in Appendix A, were sent to the end users of projects with activity levels of 90% and above in the categories of Games / Entertainment, Database, Education, Office/Business and Scientific/Engineering as shown in Fig 2.

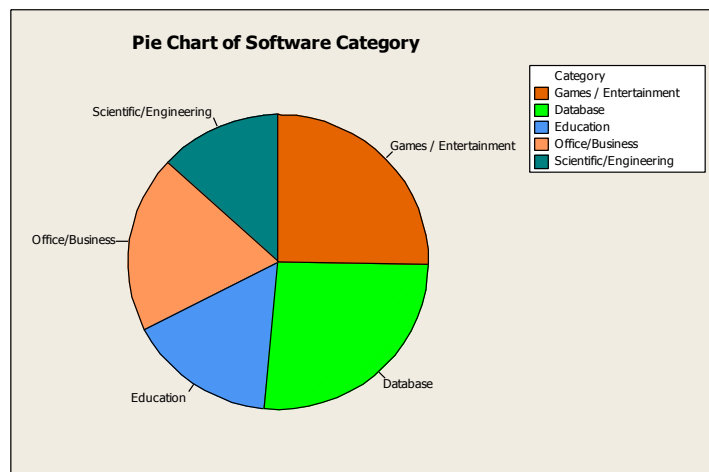


Fig 2: Respondents' Distribution

We assured the participants that our survey was confidential and that their identity would not be disclosed. However, in order to support our data analysis of user experience, we asked the respondents to reveal their experience with computers. Unlike the mandatory questions related to OSS usability, this particular question was optional. Out of the 102 responses that we received, 101 individuals chose to respond to this question. Among these responses, two respondents categorized themselves as novice computer users, ten considered themselves as average computer users and eighty-nine of the respondents believed that they were experienced users, as demonstrated in Figure 3. These statistics indicate the dominance of experienced computer users in the OSS arena.

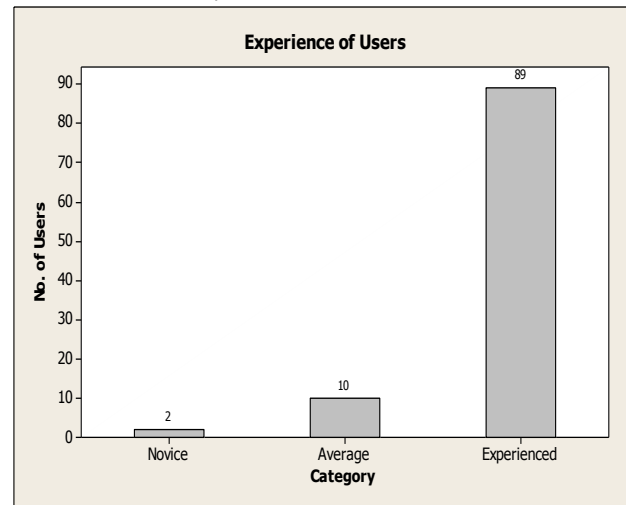


Fig 3: Experience of Users

#### 4.1. Data Collection and the Measuring Instrument

In this study, we collected data on the key usability factors and the perceived level of usability by OSS users. The measuring instruments presented in Appendix A were used to learn the perceived level of OSS usability as well as the extent to which these usability factors were important for the users of the OSS projects. Specifically, we used twenty separate items to measure the independent variables and four items to measure the perspective of OSS users regarding usability. The questionnaire required respondents to indicate the extent of their agreement or disagreement with statements using a five-point Likert Scale. For all of the items associated with each variable, the scale ranged from "Strongly Agree" (1) to "Strongly Disagree" (5). The four items for each independent variable were designed to measure the extent to which the variable was practiced within each project. The items for all five usability factors were labeled sequentially in Appendix A and numbered one through twenty. Additionally, the dependent variable, OSS Usability, was also measured on the multi-item, five-point Likert Scale. The items were specifically designed to collect measures for this variable and were labeled sequentially from one through four in Appendix A.

#### 4.2 Reliability and Validity Analysis of Measuring Instrument

The reliability and validity of a measurement are two integral features of an empirical study. Reliability indicates the reproducibility of a measurement, whereas validity refers to the agreement between the experimental value of a measurement and its true value. The most commonly used approaches in empirical studies were used to conduct reliability and validity analyses of the measuring instruments of the study. The reliability of the multiple-item measurement scales of the five usability factors was evaluated using an internal-consistency analysis which was performed using coefficient alpha [42]. In our analysis, the coefficient alpha ranged from 0.56 to 0.60 as shown in Table 1/ Fig 4. van de Ven and Ferry stated that a reliability coefficient of 0.55 or higher was satisfactory [43], and Osterhof

suggested that 0.60 or higher was satisfactory [44]. Therefore, it was concluded that the variable items developed for this empirical investigation were reliable.

Table 1: Coefficient Alpha and Principal Component Analysis (PCA) of variables

Usability Factors	Item no.	Coefficient $\alpha$	PCA Eigen value
Users' Expectations	1 - 4	0.60	2.34
Usability Bug Reporting	5 - 8	0.56	2.12
Interactive Help Features	9 - 12	0.57	2.07
Usability Learning	13 - 16	0.57	1.23
Usability Guidelines	17 - 20	0.60	1.02

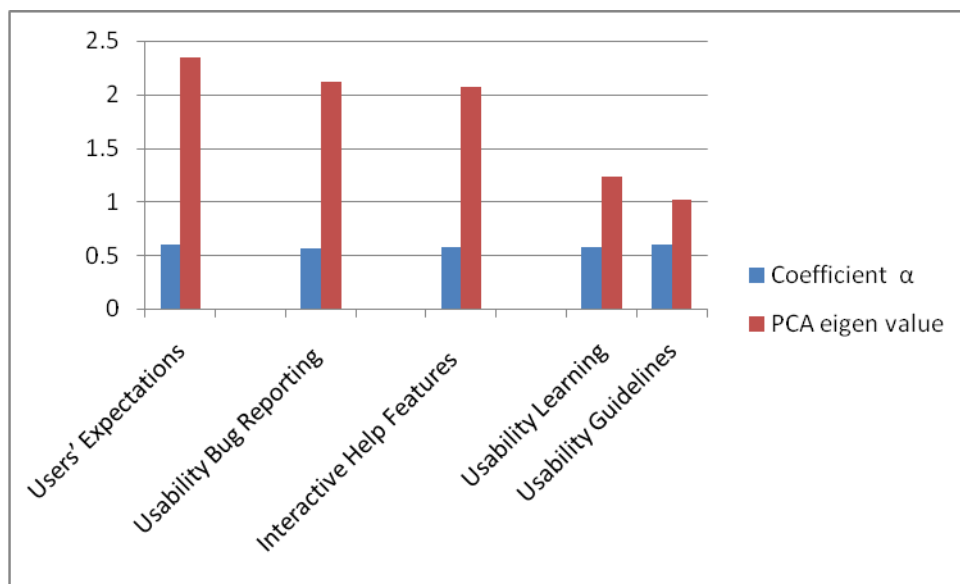


Fig. 4: Coefficient Alpha and Principal Component Analysis of variables (Users' Perspective)

According to Campbell and Fiske, convergent validity occurs in a given assembly when the scale items are correlated and move in the same direction [45]. The principal component analysis was performed for all five usability factors [46], and is reported in Table 1/ Fig 4. The Eigen-value from Kaiser was used as a reference point to observe the construct validity using principal component analysis [47]. We used Eigen-value-one-criterion, also known as Kaiser Criterion ([48], [49]) here, which means any component having an Eigen- value greater than one is retained. Eigen-value analysis revealed that all five variables completely formed a single factor. Therefore it is concluded that the convergent validity can be regarded as sufficient.

### 4.3 Data Analysis Procedure

We analyzed the research model and the significance of hypotheses H1-H5 using different statistical techniques in three phases. In Phase I we used normal distribution tests and parametric statistics, whereas in Phase II we used non-parametric statistics. Due to the relatively small sample size, both parametric as well as non-parametric statistical approaches were used to reduce the threat to external validity. Since our measuring instruments had multiple items for all five independent variables as well as the dependent variable (refer to Appendix A), their respondents' ratings were summed to get a composite value for each of them. Tests were conducted for hypotheses H1-H5 using parametric statistics to determine the Pearson correlation coefficient. For non-parametric statistics, Spearman correlation coefficient tests were determined for hypotheses H1-H5. To deal with the limitations of relatively small sample size and to increase the reliability of the results, hypotheses H1-H5 of the research model were tested using the Partial Least Square (PLS) technique in Phase III. According to Fornell and Bookstein and Joreskog and Wold, the PLS technique is helpful in dealing with issues such as complexity, non-normal distribution, low theoretical information, and small sample size [50], [51]. The statistical calculations were performed using minitab-15.

## 5. HYPOTHESES TESTING AND RESULTS

### 5.1 Phase-I

To test hypotheses H1-H5 of the research model, as shown in Figure 1, parametric statistics were used to examine the Pearson correlation coefficient between individual independent variables, the usability factors, and the dependent variable, OSS usability. The results of the statistical calculations for the Pearson correlation coefficient are displayed in Table 2/Fig. 5. *"In statistical hypothesis testing, the p-value is the probability of obtaining a test statistic. The lower the p-value, the less likely the result is if the null hypothesis is true, and consequently the more "significant" the result is, in the sense of statistical significance"* [52].

The Pearson correlation coefficient between the users' expectations and OSS usability was found to be positive (0.221) at  $P < 0.05$ , and hence justified the hypothesis H1. A Pearson correlation coefficient of 0.367 was observed at  $P < 0.05$  between usability bug reporting and OSS usability and hence was found to be significant. The hypothesis H3 was accepted based on the Pearson correlation coefficient (0.224) at  $P < 0.05$ , between the interactive help features and OSS usability. The positive correlation coefficient of 0.497 at  $P < 0.05$  was also observed between the OSS usability and usability learning which meant that H4 was accepted. However, hypothesis H5 was found to be insignificant after analyzing the Pearson correlation coefficient of 0.109 at  $P = 0.275$  between usability guidelines and OSS usability. Therefore, hypothesis H5 that deals with usability guidelines and OSS usability was rejected. Hence, as observed and reported above, hypotheses H1,

H2, H3 and H4 were found to be statistically significant and were accepted whereas H5 was not supported and therefore was rejected.

### 5.2 Phase II

Non-parametric statistical testing was conducted in this phase by examining the Spearman correlation coefficient between individual independent variables (usability factors) and the dependent variable (OSS usability). The results of the statistical calculations for the Spearman correlation coefficient are also displayed in Table 2/ Fig 5. The Spearman correlation coefficient between the users' expectations and OSS usability was found to be positive (0.281) at  $P < 0.05$ , and hence justified the hypothesis H1. The Spearman correlation coefficient of 0.371 in hypothesis H2 was observed at  $P < 0.05$  between usability bug reporting and OSS usability and hence was significant. The hypothesis H3 was accepted based on the Spearman correlation coefficient (0.271) at  $P < 0.05$ , between the interactive help features and OSS usability. The positive Spearman correlation coefficient of 0.422 at  $P < 0.05$  was also observed between the OSS usability and usability learning which meant that H4 was accepted as well. However, for hypothesis H5 the Spearman correlation coefficient of 0.025 was observed at  $P = 0.799$  between the usability guidelines and OSS usability. As no significant relationship was found between the usability guidelines and OSS usability in this test at  $P < 0.05$  hypothesis H5 was rejected.

Hence, as observed and presented above H1, H2, H3 and H4 were found to be statistically significant and were accepted whereas H5 was not supported and hence rejected in the non parametric analysis as well.

Table 2: Hypotheses testing using parametric and non-parametric correlation coefficients

Hypothesis	Usability Factor	Pearson Correlation coefficient	Spearman Correlation coefficient
H1	Users' Expectations	0.221*	0.281*
H2	Usability Bug Reporting	0.367*	0.371*
H3	Interactive Help Features	0.224*	0.271*
H4	Usability Learning	0.497*	0.422*
H5	Usability Guidelines	0.109**	0.025**

\* Significant at  $P < 0.05$ . \*\* Insignificant at  $P > 0.05$ .

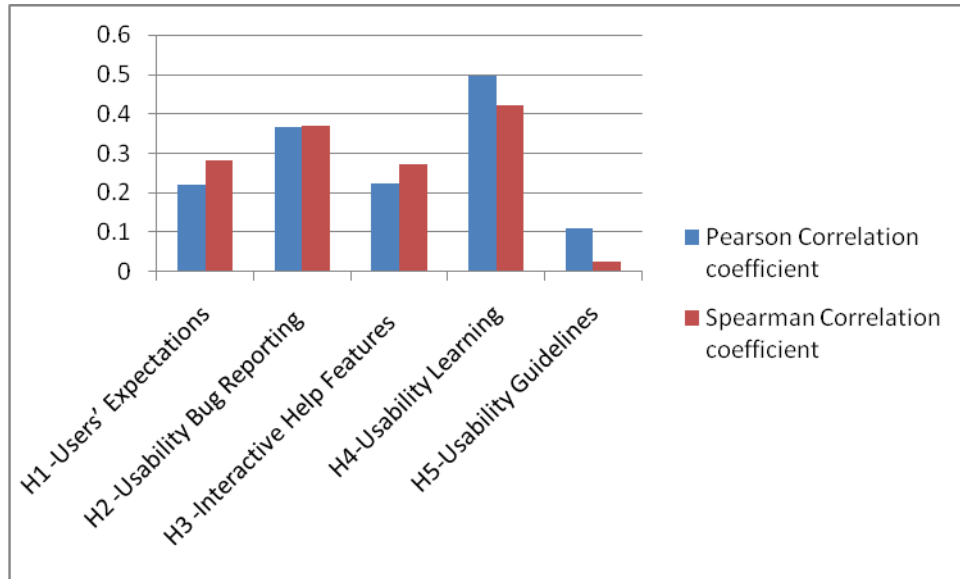


Fig. 5: Hypotheses testing (Users' Perspective)

### 5.3 Phase III

In order to perform the cross validation of the results obtained in Phase I and Phase II, the Partial Least Square (PLS) technique was used in this phase of hypotheses testing. The direction and significance of hypotheses H1-H5 were examined. In PLS, the dependent variable of the research model i.e. OSS usability was used as the response variable and independent key usability factors as predicates. The test results containing observed values of path coefficient, R<sup>2</sup> and F-ratio are shown in Table 3/Figure 6. The users' expectations were observed to be significant at P < 0.05 with path coefficient 0.221, R<sup>2</sup>: 0.049 and F-ratio as 5.15. Usability bug reporting had a path coefficient of 0.584 with R<sup>2</sup>: 0.135 and F-ratio of 15.57 and was found to be significant at P < 0.05. Interactive help features were observed to have the same direction as those proposed in hypothesis H3 with path coefficient: 0.463, R<sup>2</sup>: 0.050 and F-ratio: 5.26 at P < 0.05. Usability learning was also found to be in conformance with hypothesis H4 with observed values of path coefficient: 0.619, R<sup>2</sup>: 0.247 and F-ratio: 32.82 at P < 0.05. Finally, the usability guidelines were found to have path coefficient: 0.217, R<sup>2</sup>: 0.012 and F-ratio: 1.20 with observed P = 0.275. Hence, in this phase, as in Phase I and Phase II, hypothesis H5 that deals with usability guidelines and OSS usability was not found to be statistically significant at P < 0.05 and hence was rejected.

Table 3: Hypotheses testing using Partial Least Square (PLS) regression

Hypothesis	Usability Factor	Path Coefficient	R <sup>2</sup>	F- Ratio
H1	Users' Expectations	0.221	0.049	5.15*

H2	Usability Bug Reporting	0.584	0.135	15.57*
H3	Interactive Help Features	0.463	0.050	5.26*
H4	Usability Learning	0.619	0.247	32.82*
H5	Usability Guidelines	0.217	0.012	1.20**

\* Significant at  $P < 0.05$ . \*\* Insignificant at  $P > 0.05$

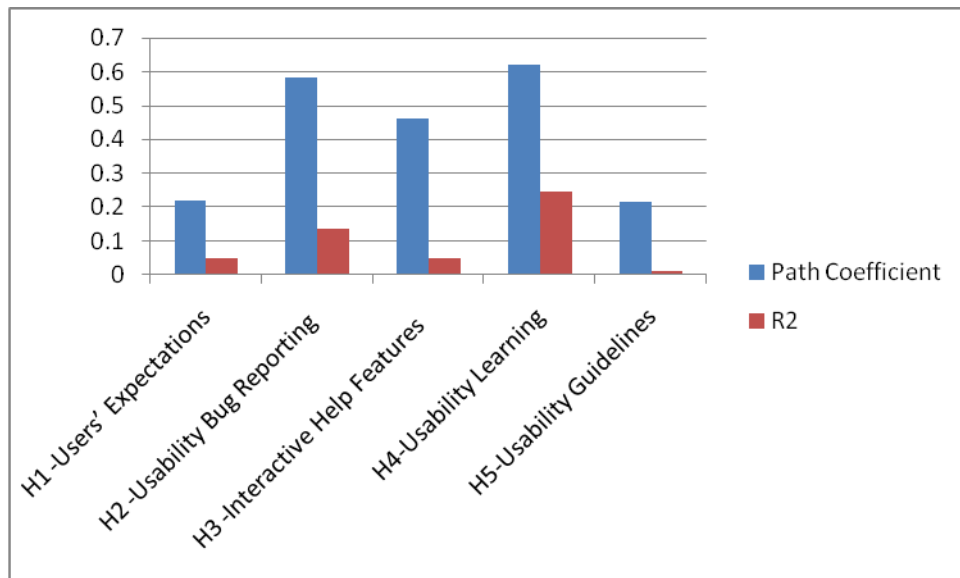


Fig. 6: Hypotheses testing using PLS regression (Users' Perspective)

### 5.4 Testing of the Research Model

The multiple linear regression equation of our research model is depicted by (1). The testing process included the regression analysis, which yielded the values of the model coefficients and their direction of association. OSS usability was used as response variable and the usability factors as predictors. Table 4 displays the regression analysis results of the research model. The path coefficient of all five variables was found to be positive whereas the t-statistics of four out of five variables namely users' expectations, usability bug reporting, interactive help features and usability learning were found to be statistically significant at  $P < 0.05$ . The t-value of the usability guidelines was observed as 0.40 at  $P = 0.688$ , thus making usability guidelines statistically insignificant in this research model.  $R^2$  and adjusted  $R^2$  of the overall research model were observed as 0.322 and 0.286 with an F-ratio of 9.10 significant at  $P < 0.05$ .

Table 4: Multiple Linear Regression Analysis of the research model.

Model coefficient Name	Model coefficient	Coefficient value	t-value
Users' Expectations	$f_1$	0.220	2.55*
Usability Bug Reporting	$f_2$	0.199	1.79*
Interactive Help Features	$f_3$	0.188	1.69*
Usability Learning	$f_4$	0.382	4.40*
Usability Guidelines	$f_5$	0.036	0.40**
Constant	$f_0$	3.92	0.45*

\* Significant at  $P < 0.05$ . \*\* Insignificant at  $P > 0.05$

Recapping Equation 1 by inserting the model coefficient values, we get:

$$\text{OSS Usability} = 3.92 + 0.22v_1 + 0.20v_2 + 0.19v_3 + 0.38v_4 + 0.04v_5$$

where  $v_1, v_2, v_3, v_4$  and  $v_5$  are the five independent variables.

## 6. DISCUSSION OF THE RESULTS

The emerging use of open source software in recent years is due to reasons such as easy (and mostly free) access and availability of the internet. However, many believe that post-release maintenance, quality control and management are among those areas where closed proprietary software has the upper hand. Twidale stresses the need for “*participatory usability involving the coordination of end users and developers*” [53]. With the popularity of OSS among organizations as well as common novice users, the OSS community is no longer limited to “technically adept” people alone. Hence, the requirements and expectations are not the same as they were a decade ago, when only software developers were supposed to be the only OSS users. Through empirical investigation this research highlights the relationship of key factors in the present research model and the OSS usability process from actual OSS users’ points of view.

In their empirical study to measure success of OSS projects, Lee et al. conclude that “*OSS use was significantly influenced by software quality and user satisfaction*” [54]. We believe that to achieve user satisfaction, software designers and developers need to understand their expectations and requirements. Furthermore, software developers need to communicate more



with the target audience of their product to better understand their perspective [55]. In our survey, forty-one percent of the respondents agreed, twenty-nine percent remained neutral and the rest thirty percent disagreed with the statement, *"User interface of OSS should follow standards and norms of proprietary software to make them easy to use,"*. On the other hand, only seventeen percent agreed with the statement, *"I believe OSS is not meant for novice non-technical users,"* five percent remained neutral and seventy-eight percent disagreed. About the statement, *"Formal feedback from users is missing and thus needed in an OSS environment,"* thirty-six percent agreed, thirty-one percent remained neutral and the rest chose to disagree with the statement. And for the last statement related to *"Users' Expectations"*, which was, *"Proprietary software addresses users' expectations and requirements better than OSS,"* seventeen percent agreed, twenty-seven percent remained neutral and the rest disagreed. We have found in the empirical investigation a positive relationship between users' expectations and the OSS usability. *"Users' Expectations"* could thus be taken by OSS developers community to be a key issue necessary to improve usability of their projects.

A subjective matter like software usability cannot be directly measured. Users also find it difficult to report such errors. Nichols and Twidale identify that it is not easy for a user to describe and hence report the difficulties s/he faces in software Graphical User Interface (GUI) [28]. Software designers and developers need to make it convenient for their users to report usability related bugs. Seventy-eight percent of the respondents in our survey agreed that the ease of reporting errors in software would increase their level of satisfaction. Ninety percent respondents did not agree with the statement *"I never report an error, so usability bug reporting does not affect me."* However, only nineteen percent agreed, forty-two percent remained neutral and the rest disagreed with our survey statement, *"For effective usability inspections, user training is required."* Moreover, sixty-three percent agreed that, *"Usability bugs reflect users' expectations; therefore, they need to be fixed on priority."* Consequently, our empirical analysis also confirms a positive association between usability bug reporting and OSS usability. Thus, *"usability bug reporting and fixing"* is considered to have a positive impact on OSS usability.

While discussing software users, their expectations and problems, the fact should not be overlooked that users do include elderly, children and people having some disability. Thus, to increase the acceptance level of OSS among different categories of users, help features of software need to be made interactive and dynamic. Sixty-six percent of our respondents believed that *"interactive help would increase ease-of-use of open source software."* Sixty-eight percent however did not agree with the statement, *"A novice user needs only basic features of software so interactive help features would not have much impact."* Moreover, seventy-one percent respondents of our survey agreed that interactive help would increase their learnability of open source software. Furthermore, our empirical investigation also supports our hypothesis, *"Interactive help features in software have a positive impact on usability in OSS."*

Benson et al. observed that quite often OSS was *"created by engineers for engineers"* [40]. Hence, in order to have a better market share and accessibility, they emphasized that OSS developers need to make suitable usability tactics their top priority. OSS managers and developers thus need to consider the importance of usable systems more seriously. Seventy-nine percent of our survey's respondents agreed that OSS developers must learn how to incorporate users' requirements and usability aspects in their software designs. Forty percent agreed that *"lack of usability knowledge is the main cause of poor usability of OSS systems."* Fifty-four percent respondents believed that *"the realization of the fact that a software system is for end users is more important than the formal usability learning."* And finally thirty-six percent agreed, fifty percent remained neutral, and thirteen percent of 102 respondents disagreed with the statement, *"OSS developers should use quantifiable usability metrics to measure it objectively and effectively."* Furthermore, our empirical analyses confirmed the positive relationship between usability learning and OSS usability as well. Usability learning was thus determined to be one of the key issues, needed to improve usability of OSS projects.

According to Hedberg et al. if existing and already proven methods are adopted, high quality and usability could be ensured in OSS [7]. As already mentioned above, Nichols and Twidale advocate that Human Interface Guidelines (HIGs) have a consensus about usability issues in terms of an authority on what shall be done [28]. Thirty-five percent respondents felt that *"there should be a standardized user interface and usability guidelines that OSS developers should follow in their designs."* Forty-one percent, however disagreed with the statement and the rest remained neutral. Sixty-seven percent were of the opinion that *"usability guidelines could act as a standard and checklist against which software could be inspected."* Similarly sixty-three percent of the respondents in our survey felt that the strict implementation of usability guidelines takes away the OSS developer's freedom. And twenty-seven percent agreed that *"standardized usability guidelines are impracticable in an OSS environment."* Considering our empirical analysis, in the parametric and non parametric statistical analysis as well as in PLS and multiple regression testing, the results did not support the positive relationship between usability guidelines and OSS usability (refer to Table 2, Table 3 and Table 4 above). Therefore, it is concluded that our study was not able to prove a positive association of usability guidelines and OSS usability.

To assess the dependent variable, OSS usability, we presented four statements to our respondents as well. Fifty-two percent respondents of our survey agreed with the statement, *"OSS with standardized usability features will help users to compare the usability of different software."* Sixty percent believed that *"poor usability being the major hurdle, improved usable OSS systems will result in switching users from proprietary software to OSS."* Fifty-nine percent were of the opinion that,

*"usable software with satisfied users guarantees its success."* Eighty-one percent OSS users of our survey showed their agreement with the statement, *"open source software having improved usability and adaptability for less technical and novice users will end up benefiting all users,"* thus indicating their strongly support towards improved usability in OSS environment.

### **6.1 Limitations of the Study & Threats to External Validity**

Like any empirical investigation, this study has certain limitations. The researcher's ability to generalize the experimental outcome to industrial practice is generally limited by threats to external validity [56], which was the case with this study as well. We took specific measures to support external validity such as using a random sampling technique to select the respondent from the population in order to conduct experiments. We retrieved the data from the most active and well-known OSS reporting website, sourceforge.net, which has a large number of projects listed. The increased popularity of empirical methodology in software engineering has also raised concerns of an ethical nature [57], [58]. The recommended ethical principles were followed in order to ensure that the empirical investigation conducted would not violate any form of recommended experimental ethics. Another aspect of validity concerned whether or not the study results reported correspond to previous findings. First of all there is the selection of independent variables in this work. We have used five independent variables to relate to the dependent variable of OSS usability. Of course there could be other key factors that influence usability but the scope of this study was kept within the area of open source software and end users' point of view. Some other contributing factors like users' educational and cultural background, their experience of using OSS projects in comparison to closed proprietary software projects developed in big organizations have not been considered in this study. It is worth mentioning here that we are also conducting three more studies, on the same format, to study developers, contributors and industry's perspective on open source software usability. Consequently many more usability factors will also be empirically investigated.

Another limitation of this study is its relatively small sample size. Although we sent our survey to a large number of OSS users who subscribed to different (13) categories of software, we received only 102 responses. The relatively small sample size in terms of number of respondents was a potential threat to the external validity of this study. We followed appropriate research procedures by conducting and reporting tests to improve the reliability and validity of the study, and certain measures were also taken to ensure the external validity.

## **7. CONCLUSION**

This study investigates the effects of key factors on OSS usability from the end user's perspective. Results of the empirical analysis show that the stated key factors of our research model assist in the improvement of OSS usability. The results support the hypotheses that users' expectations, usability bug reporting and fixing, interactive help features and usability

learning have a positive impact on usability of OSS projects. However, we could not find any significant statistical support for usability guidelines on OSS usability. The study conducted and reported here can assist OSS designers and developers to better comprehend the efficacy of the relationships of the stated key factors and usability of their projects. The stated usability factors may be considered by the OSS development community to address usability issues of their projects. This empirical investigation provided some justification for us to consider these key factors as measuring instruments for a maturity model to assess the usability of open source software project which we have recently developed.

## **APPENDIX A. KEY USABILITY FACTORS FROM OSS USER'S POINT OF VIEW (MEASURING INSTRUMENT)**

### *Users' Expectations*

1. User interface of OSS should follow standards and norms of proprietary software to make them easy to use.
2. I believe OSS is not meant for novice non-technical users.
3. Formal feedback from users is missing and thus needed in an OSS environment.
4. Proprietary software addresses users' expectations and requirements better than OSS.

### *Usability Bug Reporting and Fixing*

5. Ease of reporting errors in software would increase my level of satisfaction.
6. I never report an error, so usability bug reporting does not affect me.
7. For effective usability inspections, user training is required.
8. Usability bugs reflect users' expectations; therefore, they need to be fixed on priority.

### *Interactive Help Features*

9. I believe interactive help would increase ease-of-use of open source software.
10. A novice user needs only basic features of software so interactive help features would not have much impact.
11. Users of OSS are technically sophisticated; they do not need interactive help.
12. Interactive help would increase learnability of OSS.

### *Usability Learning*

13. OSS developers must learn how to incorporate users' requirements and usability aspects in their software designs.
14. Lack of usability knowledge is the main cause of poor usability of OSS systems.
15. The realization of the fact that a software system is for end users (not for the developers themselves) is more important than the formal usability learning.
16. OSS developers should use quantifiable usability metrics to measure it objectively and effectively.

### *Usability Guidelines for the developers*

17. There should be a standardized user interface and usability guidelines that OSS developers should follow in their designs.
18. Usability guidelines could act as a standard and checklist against which software could be inspected.
19. Strict implementation of usability guidelines will take away the OSS developer's freedom.
20. Standardized usability guidelines are impracticable in an OSS environment.

### *OSS Usability*

1. OSS with standardized usability features will help users to compare the usability of different software.
2. Poor usability being the major hurdle, improved usable OSS systems will result in switching users from proprietary software to OSS.
3. Usable software with satisfied users guarantees its success.
4. Open source software having improved usability and adaptability for less technical and novice users will end up benefiting all users.

## **REFERENCES**

- [1] E.S. Raymond The cathedral and the bazaar, O'Reilly, Sebastopol, CA, 1999.
- [2] G. von Krogh and S. Spaeth, "The open source software phenomenon: characteristics that promote research", Journal of Strategic Information Systems, vol. 16, no. 3, pp. 236-53, Sep. 2007
- [3]. M. Levesque, "Fundamental issues with open source software development", (originally published in Volume 9, Number 4, April 2004) First Monday [Online], vol. 0, no. 0, 3 Oct. 2005
- [4]. G.Polancic, R.V. Horvat and T. Rozman, "Comparative assessment of open source software using easy accessible data", In Proceedings of the 26th International Conference on Information Technology Interfaces - ITI 2004 (IEEE Cat. No.04EX794), vol. 1, pp. 673-8, 2004
- [5]. J. W. Paulson, G.Succi and A. Eberlein, "An empirical study of open-source and closed-source software products", IEEE Transactions on Software Engineering, vol. 30, no. 4, pp. 246-56, 2004
- [6] D. M. Nichols and M.B. Twidale, "The Usability of Open Source Software", First Monday vol. 8, no. 1, 2005 <http://firstmonday.org/issues/issue8.1/nichols/>
- [7] H. Hedberg, N. Iivari, M. Rajanen and L. Harjumaa, "Assuring Quality and Usability in Open Source Software Development", In Proceedings of the First international Workshop on Emerging Trends in FLOSS Research and Development, FLOSS. IEEE Computer Society, Washington, DC, 20 – 26 May, 2007
- [8] G. Çetin and M. Göktürk, "A Measurement Based Framework for Assessment of Usability-Centricness of Open Source Software Projects", In Proceedings of IEEE International Conference on Signal Image Technology and Internet Based Systems, SITIS '08, 2008
- [9] A. Porter, C. Yilmaz, A.M. Memon, A.S. Krishna, D.C. Schmidt and A. Gokhale, "Techniques and processes for improving the quality and performance of open-source software", Software Process Improvement and Practice, vol. 11, no. 2, pp. 163-76, 2006
- [10] J. Yang and J. Wang, " Review on free and open source software", In Proceedings of the 2008 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), pp. 1044-9, 2008
- [11] K. J. Stol, M.A.Babar, B. Russo and B. Fitzgerald B," The use of empirical methods in open source software research: facts, trends and future directions", 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development (FLOSS 2009),pp. 19-24, 2009
- [12] R. Ferenc, I. Siket and T. Gyimothy, "Extracting facts from open source software", In Proceedings of the 20th IEEE International Conference on Software Maintenance, pp. 60-9, 2004
- [13] D. Hedgebeth, "Gaining competitive advantage in a knowledge-based economy through the utilization of open source software", VINE: The journal of information and knowledge management systems, vol. 37, no. 3, pp. 284-94, 2007
- [14] E. Golden, B.E. John and L.Bass, "The Value of a Usability-Supporting Architectural Pattern in Software Architecture Design: A Controlled Experiment", In Proceedings of the 27th International Conference on Software Engineering, 15-21 May, 2005, St. Louis, MO, USA.
- [15] E. Y. Nakagawa, E.P.M. de Sousa, M.K. de Brito, A.G. de Faria, L.B. Morelli and J.C. Maldonado, "Software architecture relevance in open source software evolution: a case study", In Proceedings of the 2008 IEEE 32nd International Computer Software and Applications Conference (COMPSAC), pp. 1234-9, 2008
- [16] L. Stamelos, L. Angelis, A. Oikonomou and G.L. Bleris, "Code quality analysis in open source software development", Information Systems Journal, vol. 12, no. 1, pp. 43-60, 2002

- [17] J. Feller and B. Fitzgerald, "A Framework Analysis of the Open Source Software Development Paradigm", In Proceedings of the 21st Annual International Conference on Information Systems, Brisbane, Queensland, Australia, pp. 58–69, 2000
- [18] S. Koch and C. Neumann, "Exploring the effects of process characteristics on product quality in open source software development", *Journal of Database Management*, vol. 19, no. 2, pp. 31-57, 2008
- [19] T. Gyimothy, R. Ferenc and I. Siket, "Empirical validation of object-oriented metrics on open source software for fault prediction", *IEEE Transactions on Software Engineering*, vol. 31, no. 10, pp. 897-910, 2005
- [20] S. Mansfield-Devine, "Open source: does transparency lead to security?", *Computer Fraud & Security*, vol. 9, pp. 11-13, 2008
- [21] I. Samoladas, G. Gousios, D. Spinellis and Ioannis Stamelos, "The SQO-OSS quality model: measurement based open source software evaluation", *Open Source Development, Communities and Quality — OSS 2008: 4th International Conference on Open Source Systems*, pages 237–248.
- [22] B. Golden, "Making Open Source Ready for the Enterprise, The Open Source Maturity Model", Extracted From *Succeeding with Open Source*, Addison-Wesley Publishing Company, 2005.
- [23] Business Readiness Rating. Business readiness rating for open source. <http://www.openbrr.org>.
- [24] QSOS. Method for qualification and selection of open source software (qsos) version 1.6. <http://www.qsos.org>.
- [25] J.C. Deprez, and S. Alexandre, "Comparing Assessment Methodologies for Free/Open Source Software: OpenBRR and QSOS", Jedlitschka, A., Salo, O. (eds.) *PROFES 2008. LNCS*, vol. 5089, pp. 189–203. Springer, Heidelberg, 2008
- [26] ISO 9241. Ergonomics Requirements for Office with Visual Display Terminals (VDTs), 1997
- [27] International Standard ISO/IEC 9126-1 (2001) Software Engineering – Product Quality – Part 1: Quality Model, 1<sup>st</sup> ed., pp. 9-10, 15 Jun. 2001
- [28] D.M. Nichols and M.B. Twidale, "Usability Processes in Open Source Projects", *Software Process: Improvement and Practice*, vol. 11, no. 2, pp. 149- 162, 2006
- [29] M. Aberdour, "Achieving quality in open-source software", *IEEE Software*, vol. 24, no. 1, pp. 58–64, Jan./Feb. 2007, doi:10.1109/MS.2007.2
- [30] F. Sampson, "Who said "usability is free"?", *Interactions*, vol. 14, no. 4, pp. 10-11, 2007
- [31] M.B. Twidale and D.M. Nichols, "Exploring usability discussions in open source development", In Proceedings of the 38th Annual Hawaii International Conference on System Sciences, pp. 198-208, 2005
- [32] M. Bodker and L. Nielsen and R.N. Orngreen, "Enabling user centered design processes in open source communities: Usability and Internationalization, HCI and Culture", In Proceedings of the Second International Conference on Usability and Internationalization, UI-HCII 2007. Held as Part of HCI International 2007, part I, pp. 10-18, 2007
- [33] L. Zhao and F.P. Deek, "Exploratory Inspection – A Learning Model for Improving Open Source Software Usability", In Proceedings of the Conference on Human Factors in Computing Systems CHI '06, pp. 1589 – 1594, 2006
- [34] G. Cetin, D. Verzulli and S. Frings, "An analysis of involvement of HCI experts in distributed software development: practical issues", *Online Communities and Social Computing*. In Proceedings of the Second International Conference, OCSC 2007. Held as Part of HCI International 2007, pp. 32-40, 2007
- [35] B. Shneiderman, "Universal Usability", *Communications of the ACM*, vol. 43, no. 5, pp. 84-91, May 2000
- [36] N. Viorres, P. Xenofon, M. Stavrakis, E. Vlachogiannis, P. Koutsabasis and J. Darzentas, "Major HCI Challenges for Open Source Software Adoption and Development", In Proceedings of Schuler, Douglas (ed.) OCSC 2007 - Online Communities and Social Computing, Second International Conference, Beijing, China, pp. 455-464, 22-27 Jul. 2007
- [37] C. Rusu, V. Rusu and S. Roncagliolo, "Usability Practice: The Appealing Way to HCI", In Proceedings of First International Conference on Advances in Computer-Human Interaction, achi-2008, pp. 265-270, 2008
- [38] X. Faulkner and F. Culwin, "Integrating HCI and SE", *ACM SIGCSE Bulletin*, vol. 32, no. 3, pp. 61-64, 2000
- [39] N. Iivari, H. Hedberg, and T. Kirves, "Usability in company open source software context. Initial findings from an empirical case study", In Proceedings of the 4th International Conference on Open Source Systems (co-located with the World Computer Congress), Milan, 7-10 Sep. 2008.
- [40] C. Benson, M. Muller-Prove and J. Mzourek, "Professional usability in open source projects: GNOME, OpenOffice.org, NetBeans", In Proceedings of the CHI '04 extended abstracts on Human factors in computing systems, Vienna, Austria, Apr. 2004
- [41] E. Folmer and J. Bosch, "Architecting for usability: A survey", *The Journal of Systems and Software*, vol.70, pp. 61–78, 2004
- [42] L.J. Cronbach, "Coefficient alpha and the internal consistency of tests", *Psychometrika*, vol. 16, pp. 297–334, 1951
- [43] A.H. van de Ven and D.L. Ferry, "Measuring and Assessing Organizations", John Wiley & Son, NY, 1980.
- [44] A. Osterhof, "Classroom Applications of Educational Measurement", Prentice Hall, NJ, 2001.
- [45] D.T. Campbell and D.W. Fiske, "Convergent and discriminant validation by the multi-trait multi-method matrix", *Psychological Bulletin* vol. 56, no. 2, pp. 81–105, 1959
- [46] A.L. Comrey and H.B. Lee, "A First Course on Factor Analysis", second ed. Hillsdale, 1992
- [47] H.F. Kaiser, "A second generation little jiffy", *Psychometrika*, vol. 35, pp 401–417, 1970
- [48] H.F. Kaiser, "The application of electronic computers to factor analysis", *Educational and Psychological Measurement*, vol. 20, pp. 141–151, 1960
- [49] J. Stevens, "Applied Multivariate Statistics for the Social Sciences", Hillsdale, NJ, 198.
- [50] C. Fornell and F.L. Bookstein, "Two structural equation models: LISREL and PLS applied to consumer exit voice theory", *Journal of Marketing Research*, vol. 19, pp. 440–452, 1982
- [51] K. Joreskog and H. Wold, "Systems Under Indirect Observation : Causality, Structure and Prediction", North Holland, The Netherlands, 1982
- [52] Wikipedia, <http://en.wikipedia.org/wiki/P-value>, 2010
- [53] M. Twidale, "Silver bullet or fool's gold: supporting usability in open source software development", In Proceedings of the 27th International Conference on Software Engineering, pp. 35, 2005
- [54] S.Y.T. Lee, H.W. Kim and S. Gupta, "Measuring open source software success" *Omega*, vol. 37, no. 2, pp. 426-438, 2009
- [55] H. Koppelman and B. Van Dijk, "Creating a Realistic Context for Team Projects in HCI", *SIGCSE Bulletin*, vol. 38, no. 3, pp. 58-62, 2006

[56] C. Wohlin, P. Runeson, M. Host, M.C. Ohlsson, B. Regnell and A.Wesslen, "Experimentation in Software Engineering", Kluwer Academic Publishers, Norwell, MA, 2000

[57]. R.R. Faden, T.L. Beauchamp and N.M.P. King, "A history and theory of informed consent", Oxford University Press, 1986

[58]. J. Katz, "Experimentation with human beings", New York: Russell Sage Foundation, 1972

**Arif Raza** received his M.Sc. (1994) in Computing Science from Birkbeck College, University of London, U.K. and Ph.D. in Software Engineering from the University of Western Ontario, Canada. Dr. Raza has several years of teaching and research experience. He has authored and co-authored several research articles in peer reviewed journals and conference proceedings in the area of software engineering. His current research interests include empirical studies, usability assessment of open source software projects and human factors in SE. He can be reached at [araza7@uwo.ca](mailto:araza7@uwo.ca).

**Luiz Fernando Capretz** has almost 30 years of international experience in the software engineering field as a practitioner, manager and educator. Having worked in Brazil, Argentina, U.K., Japan, Italy and the UAE, he is currently an Associate Professor and the Director of the Software Engineering Program at the University of Western Ontario, Canada. He has published over 100 peer-reviewed research papers on software engineering in leading international journals and conference proceedings, and has co-authored two books in the field. His present research interests include software engineering (SE), human factors in SE, software estimation, software product lines, and software engineering education. Dr. Capretz received his Ph.D. in Computing Science from the University of Newcastle upon Tyne (U.K.), his M.Sc. in Applied Computing from the National Institute for Space Research (INPE, Brazil), and his B.Sc. in Computer Science from State University of Campinas (UNICAMP, Brazil). He is an IEEE senior member, ACM member, MBTI certified practitioner, Professional Engineer in Ontario (Canada), and he can be contacted at [lcapretz@eng.uwo.ca](mailto:lcapretz@eng.uwo.ca).

**Faheem Ahmed** received his M.E.Sc. (2004) and Ph.D. (2006) in Electrical Engineering from the University of Western Ontario, London, Canada. Currently, he is an assistant professor at the College of Information Technology, UAE University, Al Ain, United Arab Emirates. Ahmed has many years of industrial experience holding various technical positions in software development organizations. During his professional career he has been actively involved in the life cycle of the software development process including requirements management, system analysis and design, implementation, testing, delivery and maintenance. Dr. Ahmed has authored and co-authored many peer-reviewed research articles in leading journals and conference proceedings in the area of software engineering; His current research interests are Software Product Line, Software Process Modeling, Software Process Assessment, and Empirical Software Engineering. He is a member of IEEE and can be reached at [f.ahmed@uaeu.ac.ae](mailto:f.ahmed@uaeu.ac.ae)

Usability: An Empirical Study

23