

# THE IMPACT OF IN-HOUSE SOFTWARE DEVELOPMENT PRACTICES ON SYSTEM USABILITY IN THE SOCIAL SECURITY FUNDS IN TANZANIA

Hamis Fredrick Eliazer<sup>1</sup> and Michael John Haule<sup>2</sup>

<sup>1</sup>Postgraduate Student Institute of Accountancy Arusha, Tanzania

<sup>2</sup>Senior Lecturer, Institute of Accountancy Arusha, Tanzania

## ABSTRACT

*In-house software development is a critical phenomenon for the production of efficient and effective software in generating requisite job output. A few studies have devoted efforts towards establishing the impact of in-house software development on software. Therefore, this paper is an effort towards establishing the impact of in-house software development practices on system usability. In pursuit of this paper, a sample of a sample size of 169, at 95% confidence level, with margin error of 5% was drawn from bold software users, i.e. 300 employees who used the all software including those dealing with the main stream activities. A total of 102 respondents actually responded to the questionnaires. The Online Sample Calculator was used to draw the sample. Quantitative data were collected using semi-structured questionnaires and processed using the SPSS. Descriptive statistics were applied in the analysis. Findings of the study indicate that software development practices, specifically usability test and user involvement in software designing and development had an impact on determining software usability for in-house software. The paper concludes that software development practices shape the design of the software; hence influence usability of the software produced. Recommended is therefore that software usability test and user involvement in software designing and development be promoted for effective software production.*

## KEYWORDS

*In-house software development, Software development practices and Software usability.*

## 1. INTRODUCTION

Software development is the process of planning and actual production of software tailored to processing certain expected output. The process of software development is closely linked to the software use for a specific software is basically meant to perform specific tasks for specified quality results. Literature defines software usability as a degree of ease of system software of web application [1]. It is the controlled aspect of user experience design that ensures the user does not encounter problems when using the software product or website [2][3]. Whether in-house software development practices applicable in Tanzania have taken into account software usability is an aspect covered by this paper.

Bevan and others [4] define usability in terms of effectiveness, efficiency and satisfaction in a particular context of use. The same definition is held by Chaudhary and Upadhyav [1] and Abran and others [5]. Whether usability test is being applied in in-house software development in the context of Tanzania is an important aspect this paper would like to shed light on to. The design aspect of software development process which is an articulated software development practice is considered indispensable aspect is enhancing software usability and evaluation [3][6]. One would

be interest to establish which specific software development practices are adopted and applicable in Tanzania so as to ensure higher software usability levels.

Software usability testing or valuation, a step towards improving software qualities is important for minimizing design costs, saves production time and raises user comfortability [3][7]. How the situation persists in Tanzania, concerning this stand-point demands to be uncovered by this paper. From the above background, it may be inferred that the purpose of this paper is to determine whether software development practices applicable in Tanzania, within the NSSF in particular, have an effect on software usability.

## 2. LITERATURE REVIEW

In-house software development refers to initiatives for an organization to produce software internally for use within the organization. The need to develop such software may arise depending on many circumstances which may include non-availability of the software in the market, potentiality, or ability of the corporation to develop such software or to customize the software based on the corporate organization's need. There are high possibilities of software development failure for outsourcing the software development due to miscommunication and unclear understanding of the system needs and business process of the organization [8]. Even though both in-house software development and offshore outsourcing face challenges, offshore-outsourced projects are especially prone to failure. Information Technology managers are constantly seeking ways to increase their internal developments for better software products and cost-effectiveness [8].

The majority of organizations go for outsourcing due to some factors, such as allowing the company or organization to focus on their primary business. Outsourcing would allow organizations to focus on their areas of specialization, leaving software developments to software companies. Others consider cost, although this has been discussed as a predominant reason [9]. However, Jennings [10] cautions that the outsourcing decision requires the recognition of a wide range of contextual factors and the implication of their long-term development. Some light as shed on what happens to usability and occupational health issues in a be-spoke software development project, where cautioned that poor usability may increase cognitive workload and deter the workers from using their cognitive capacity to efficiently achieve their goals thus causing mental stress [11].

The International Organization of Standardizations, ISO, defines usability of a software product to be a key software quality factor and has developed a variety of models to specify and measure software usability [3]. According to Bevan and others [4] usability is defined in terms of effectiveness, efficiency, and satisfaction in a particular context of use. *“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”* [3]. It is about learnability, efficiency, memorability, errors, and satisfaction [12] [6]. Chaudhary and Upadhyav [1] emphasize the same attributes regarding software usability.

Software usability refers to the degree of ease of system software or web application. It is a controlled aspect of user experience design that ensures that the user does not encounter problems with the use of a product or website's user interface [2]. The usability engineers ensure the design makes it easy for the users to interact with the system application or web application. Usability engineering refers to the discipline concerned with the design, evaluation, and implementation of interactive computing systems for human use and the study of major phenomena surrounding them [7]. Usability testing or evaluation is important in software development processes as it

improves software, saves time and cost and it minimizes software engineering designing costs by avoiding the redesigning of the software if the user is not be comfortable with the usability[7]. Usability has been reported to be economically beneficial for increased sales, productivity, decreased training costs, and decreased support needs [13].

Moreover, it is argued that system usability refers to the ease of operation and usefulness as serving the intended purpose [3][14]. This means the system software has to observe the intended functionalities and purposes, and should always stick to meeting these intentions while making it easy for the users to interact and operate the system effectively. According to Jeng [15], Usability is a multidimensional construct that can be examined from various perspectives which can be determined by the tasks, the users, the product, and the environment. Usability is determined by how the user can effectively use the system through the well-designed interface to perform the specified tasks accurately. This means the system interface has a lot to do with software usability. Rubin [16] argues that usability has a theoretical base on human-computer interaction. Therefore interface is the most important aspect of usability since it is the medium through which users communicate and interact with the system [6][16].

However, it is not easy for developers to comply with the available usability resources due to some reasons. It has been argued that usability sources are not straightforward for developers and evaluators due to their uncertainties, hardship in selecting appropriate guidelines to a particular context of use, and some guidelines being not usable by themselves [17]. In this study, software usability refers to the extent of ease of system software product or web application by its users. It is determined by the ability of the system software product users to interact with the system through the well-designed interface to perform their operations to produce the intended output. Task completion and learn ability to use the system make special concerns in software product usability.

Usability Testing refers to the process of evaluating a software product by testing it with selected users to examine whether the users can complete a task and to determine user satisfaction with the product [18]. Software developers, before deployment, organize usability testing to find the flaws in the usability of the product. Unlike the User Acceptance Test (UAT) which refers to the testing by the client to make sure the product work as per the agreed requirements, Usability Testing strives to ensure the user can use the product with ease [19]. Chaudhary and Upadhyav [1] advocate for provision of software user guidelines to promote effective software usability. Usability testing employs a variety of methods and techniques that help to assess the software product usability by sampling users to determine how effective they can use the product with ease and satisfaction [9]. According to Nielsen [12], the following are some of the Usability testing methods.

Heuristic evaluation is the most informal method and involves having a usability specialist judge whether each dialogue element follows established usability principles. Users are given a test and someone assesses to ensure if each element follows the defined usability principles. Cognitive walkthroughs make use of a more explicitly detailed procedure to simulate a user's problem-solving process at each defined step through the dialogue, checking the simulated user's goals and memory content can be assumed to lead the next correction action[12]. According to Nielsen [12], formal usability inspection is a method which uses six steps procedure with strictly defined roles to combine heuristic evaluation and a simplified form of cognitive walkthroughs. The combination of both the cognitive walkthroughs methods and the heuristic evaluation forms the formal usability inspection [12]. Unlike the Cognitive Walkthroughs, the Pluralistic walkthroughs conduct meetings where users, developers, and human factors people step through a scenario, discussing each dialogue element to determine whether it meets the defined usability principles or not [12].

Feature inspection lists a sequence of features used to accomplish typical tasks, checks for long sequences, cumbersome steps, steps that would not be natural for users to try, and steps that require extensive knowledge/ experience to assess a proposed feature set[9] Consistency inspection involves the designers representing multiple projects to inspect an interface to see whether it does things in the same way as their designs[12]. The standard inspection, unlike the consistency inspection, has an expert on some interface standards inspect the interface for compliance [12]. In *Investigating Software Development Methodologies and Practice in Software Industry in Tanzania* provided is an analysis of the software development methodologies that are in practice by software developers in Tanzania. The investigation strived to establish the methodologies that have been adopted by the software development firms in Tanzania, where they found that a large percentage of software developers in Tanzania has adopted the traditional software development methodology which is Water Fall Methodology [20]. In the paper titled “*Why usability gets lost or usability in in-house software development*” observes the existence of the tendency of loss of usability for in-house software development and analyze some factors behind the losses. According to their findings, the usability of the software is not assigned any economic values, hence contributing to the loss of system usability in organizations [11]. In *Usability Testing in Digital Libraries*, usability testing issues have been concerned in the digital libraries (DL) community [16]. This tells how important Usability Testing is for software.

### **3. CONCEPTUAL FRAMEWORK AND MODEL**

#### **3.1. Conceptual Framework**

The study conceptualizes that the applicable software development practices tend to mould the effectiveness of software developed hence influencing its usability [16]. The actual process of software development is considered have an influence on the manner that software usability would be rated.

Having the involvement of the users in the process of identifying features of the software to be developed is considered the linking and important factor towards promoting and enhancing utility of the software produced [15]. This kind of dependency constitutes the core quality of any type of software developed. In-house developed software has a higher chance of being compatible with the job procedures of a particular organization, hence ensuring its usability. Therefore, in this study, software development practices becomes an independent variable while software usability remains to be a dependant variable. Therefore conducting software usability testing for in-house software developments in various organizations is considered critical towards the production of useful software for a particular organization. The process ensures determining the level of software usability in in-house software products for any type of an organization [15].

### 3.2. Conceptual Model

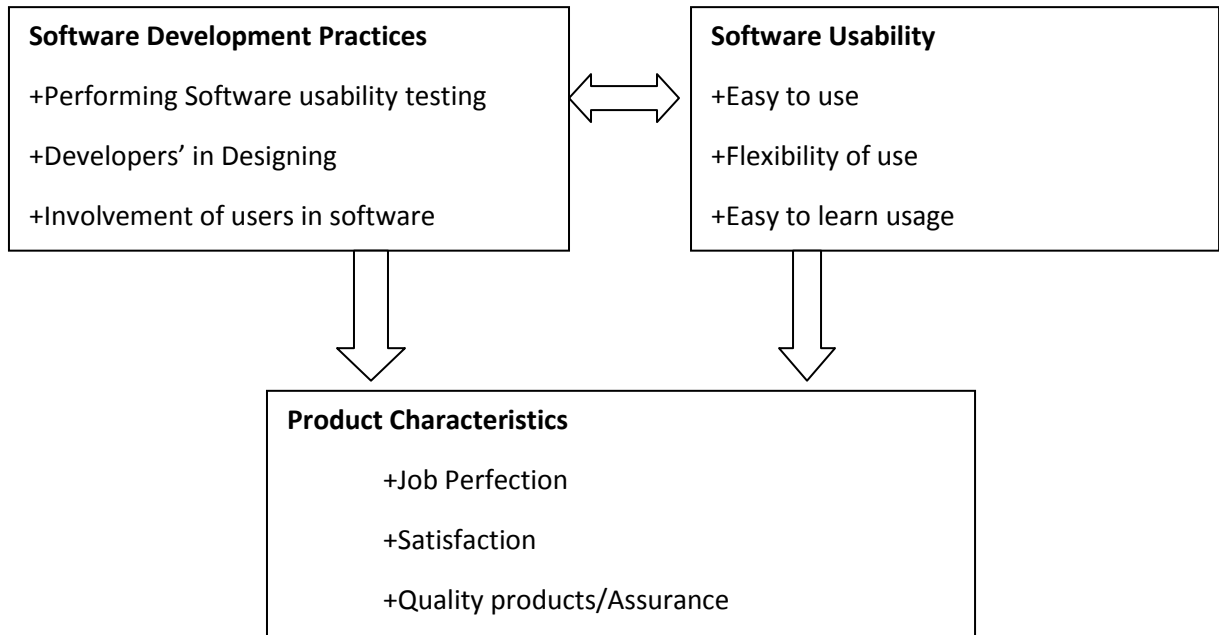


Figure 1. Conceptual Model

Source: Researcher's own Construct, 2021

## 4. METHODOLOGY

The quantitative study was conducted at the National Social Security Head Office in Dar es Salaam analyzed the quantitative data collected from both software developers and software users working with the National Social Security Fund. A sample size of 169, at 95% confidence level, with margin error of 5% was to be drawn from bold users, i.e. 300 employees who used the all software including those dealing with the main stream activities. However, 102 respondents actually responded to the questionnaires. Data were collected for three weeks in August, 2020. The Online Sample Calculator was used to draw the sample. The NSSF has a total of 1,435 employees countrywide. This study adopted a descriptive research design that illustrates the characteristics of a population or a phenomenon being studied. This research interviewed different people involved with software development and usability testing. The descriptive study used *Likert scale* in the analysis whereby percentages and cross tabulation were used to describe and conclude the findings.

## 5. PRESENTATION AND DISCUSSION OF FINDINGS

### 5.1. Demographic Characteristics of the Respondents

Survey data in Table 1.0 indicate that the user-respondents who participated in this study were from different age groups. The majority of the respondents were from age group 25-44 they were in their middle age. While fewer, i.e. about 8% were youth aged between 18 and 24; another smaller proportion of about 8% were aged over 45 years, hence consisting of the elderly and above. Generally, it is most likely that the people in the younger age category constituted the main users of the software and related information technologies.

Table 1. Age of the Respondents

Age-groups	Frequency	Percent	Cumulative Percent
18-24	8	7.8	7.8
25-34	60	58.8	66.7
35-44	26	25.5	92.2
45+	8	7.8	100.0
<b>Total</b>	<b>102</b>	<b>100.0</b>	

Source: Field Data, 2021

The study visualized the need to categorize the user respondents by gender to unveil the gender technology divide among NSSF staff, regarding software use in their daily work performance. Survey data in Table 1 indicate that about 83% of in-house software users were males, while 17% were females. Though the number of males tended to dominate, it does not infer that females were laggards in technology use, though the male/female ratio in employment in the National Social Security Funds is 50:50, where males are 716 and females are 719.

## 5.2. Applied Software Development Practices

### 5.2.1. Software Usability Testing

Survey data in Table 2 indicate that about 73.9% of the respondents “agree” that they perform usability testing as one of the in-house software development practices, which suggests that the user satisfaction is contributed by this practice and other in-house software development practices.

Table 2. Ranking by Usability Testing

Responses	Frequency	Percent	Cumulative Percent
YES	17	73.9	73.9
NO	9	26.1	100
<b>Total</b>	<b>23</b>	<b>100</b>	

Source: Field Data, 2021

### 5.2.2. Involvement of Software Developers in Software Designing

Software designing is a critical step in software development. It entails understanding of user requirement and output expected. The quality of any software relies very much on the designing and its suitability for the purpose it is intended to address. Data in table 3 demonstrate that not all IT staff, who are software developers, were engaged in software designing activities. In fact the majority of them, i.e. about 66% were involved in the process.

Table 3. Respondents by Roles they Played in Software Development

<b>Responses</b>	<b>Frequency</b>	<b>Percent</b>	<b>Cumulative Percent</b>
<b>Requirements</b>	5	21.7	21.7
<b>Designing</b>	15	65.2	87.0
<b>Testing</b>	3	13.0	100.0
<b>Total</b>	23	100.0	

Source: Field Data, 2021

In the software development process, software users were found to be less involved. Only a few, i.e. software developers, were asked to provide information that was tailored to the software production. The results were software products which could not fully suffice the need of the organization. The dictation of software developers and the attitude of “knowing what is required” without obtaining information from the users tended to undermine the software development process in the NSSF. This fact is underlined by the challenges software developers observed from the users’ perspective.

### 5.2.3. Involvement of Users in In-House Software Development Process

To establish whether users were involved in the software development process the reverse analysis was conducted through reviewing the challenges IT staff, i.e. software developers faced while implementing the introduction of in-house software in the NSSF. The fact portrayed by data in Table 4 indicate that a significant section of the users, i.e. 39% indicated that they lacked exposure to the software implies that they were less involved in the development process. About 17 % had resistance to adopt use of the same software while another 17% lacked awareness of the software used. Another 17% found software used as un-clear regarding their capability to meet specific job requirements. Smaller proportions, i.e. about 4% indicated to be slow in adoption of technology, while other % had slow user response. The entire scenario would be the opposite in case users were involved in the process of software development in the initial and other critical stages.

Table 4. Distribution of Respondents by Challenges Emanating from Users

<b>Responses</b>	<b>Frequency</b>	<b>Percent</b>	<b>Cumulative Percent</b>
<b>User Resistance</b>	4	17.4	17.4
<b>Slow User Response</b>	1	4.3	21.7
<b>Technology</b>	1	4.3	26.1
<b>Poor Training and exposure</b>	9	39.1	65.2
<b>Unclear requirements</b>	4	17.4	82.6
<b>Less User awareness</b>	4	17.4	100.0
<b>Total</b>	<b>23</b>	<b>100.0</b>	

Source: Field Research 2021

### 5.3. Respondents' Ranking of Software Usability for In-house Developed Software in the NSSF

#### 5.3.1. Ranking Easy to Use of the In-House Developed Software

When asked about flexibility for use of the in-house developed software, the respondents had different rankings. According to the data in table 5, those who “*agreed*” constituted 49%, while 48% *strongly agreed* that the in-house developed software was *easy to use*, therefore, facilitating users' when used the same. It is imperative that being *easy to use* made such software be used by the majority for different purposes.

Table 5. Ranking Respondents by Easy To Use

Responses	Frequency	Percent	Cumulative Percent
<b>Strongly Agree</b>	49	48.0	48.0
<b>Agree</b>	50	49.0	97.1
<b>Neutral</b>	3	2.9	100.0
<b>Total</b>	102	100.0	

Source: Field Data, 2021

#### 5.3.2. Ranking In House Software in terms of Flexibility of Use

When asked about flexibility for use of the in-house developed software, the respondents had different rankings. According to data in table 6, those who “*agreed*” constituted 49%, while 40% strongly agreed that the in-house developed software was flexible in use, therefore, facilitating users' flexibility when used the same. It is imperative that being flexible made such software be used by the majority for different purposes.

Table 6. Ranking In House Software -Flexible to Use

Responses	Frequency	Percent	Cumulative Percent
<b>Strongly Agree</b>	44	43.1	43.1
<b>Agree</b>	50	49.0	92.2
<b>Neutral</b>	8	7.8	100.0
<b>Total</b>	102	100.0	

Source: Field Data, 2021

#### 5.3.3. Ranking Software in terms of Easy to Learn Usage

Easy-to-learn usage derives from having easy instructions due to the logical steps which can be easily understandable and applicable by the users. Data in table 7 demonstrate the presence of the majority, i.e. about 57% of the respondents who “*agreed*” that in-house developed software was easy to learn in terms of how to use them. A proportionate section of the respondents, i.e. about 35% “*strongly agreed*”. Therefore the vast majority of the respondents agreed that software that was developed from within the organization was easily tailored to users through their being easy to learn. Only a smaller section of the respondents, i.e. 8% found it to be normal.



Table 7. Ranking in House Software as Easy To Learn its Usage

Responses	Frequency	Percent	Cumulative Percent
<b>Strongly Agree</b>	36	35.3	35.3
<b>Agree</b>	58	56.9	92.2
<b>Neutral</b>	8	7.8	100.0
<b>Total</b>	<b>102</b>	<b>100.0</b>	

Source: Field Data, 2021

## 5.4. The Impact of In-house Software Development Practices with Software Usability in the NSSF

### 5.4.1. Software Usability Testing and Easy to Use

The testing for the usability of in-house software, ranking various aspects of the software was performed analyzed basing on *easy to use*, *quick accomplishment of tasks*, *the flexibility of use*, and *easy to learn usage and* Results of the same sections are as indicated hereunder.

To ascertain that software usability testing which was mentioned by 74% of the software developers that it was being performed in the process of software development at the NSSF, it was found plausible and necessary to test some aspects of the same software basing on users experiences on the in-house software. Cross tabulation was run for *easy to use* and *quick accomplishment of tasks*. The logical assumption being that in case the software development process was pre-tested for usability then its users will find it *easy to use* and that *task would be accomplished quickly*.

Results, as demonstrated in Table 8 indicate that the majority of the respondents, i.e. 96/102 agreed that the in house software were both *easy to use* and *expedited job completion*. This suffices one to conclude that usability of the produced software is to the acceptable levels.

Table 8. Easy To Use and Software’s ability for Quick Accomplishment of Tasks-Cross tabulation

Easy To Use	Ability for Quick Accomplishment of Tasks				Total
	Strongly Agree	Agree	Neutral	Disagree	
<b>Strongly Agree</b>	30	16	2	1	<b>49</b>
<b>Agree</b>	6	42	1	1	<b>50</b>
<b>Neutral</b>	0	2	1	0	<b>3</b>
<b>Total</b>	<b>36</b>	<b>60</b>	<b>4</b>	<b>2</b>	<b>102</b>

Source: Field Data, 2021

### 5.4.2. User's Involvement in Software Development and Generation of Correct Output by the Software

Involvement of users in software development was mentioned to have taken place during initial software development phase where critical information was collected from the potential users so as to be tailored to the software to be produced.

To test whether users were actually involved in software development, cross tabulation of data on *flexibility of use* and *ability to generate correct output* was performed. Results, as in table 9 indicate that the vast majority of respondents (74/102) agreed that in-house software developed within the NSSF were both flexible to use and had the ability to generate correct output, hence indicating the positive impact of software development practice on usability.

Table 9. Flexibility of Use and its Ability to Generate Correct Output – Cross tabulation

Flexibility of Use	Ability to Generate Correct Output				Total
	Strongly Agree	Agree	Neutral	Disagree	
Strongly Agree	27	17	0	0	44
Agree	8	40	1	1	50
Neutral	0	4	3	1	8
<b>Total</b>	<b>35</b>	<b>61</b>	<b>4</b>	<b>2</b>	<b>102</b>

Source: Field Data, 2021

### 5.4.3. User Involvement in Software Design and Trusted Output

Users' ideas and requirements are important ingredients in the designing of any software with high usability levels. Cross tabulation off data of generation of trusted output and satisfactory software functions was performed. This was conducted in the pretext that in case users were adequately involved in the designing of the software the resulting software shall have trusted output for both the organization and its users. Field data presented in table 10 demonstrate that the majority of the respondents, i.e. 66/102 agreed that in-house software developed by the NSSF generated *trusted output* and had *satisfactory functions* for the requisite tasks. Therefore the above scenario indicates that software development practice of involving users in the designing phase has an impact on software usability.

Table 10. Generation of Trusted Output and having Satisfactory Functions – Cross tabulation

Generating Trusted Output	Ranking Software for their Satisfactory Functions			Total
	Strongly Agree	Agree	Neutral	
Strongly Agree	19	9	0	28
Agree	9	51	6	66
Neutral	2	4	1	7
Disagree	0	0	1	1
<b>Total</b>	<b>30</b>	<b>64</b>	<b>8</b>	<b>102</b>

Source: Field Data, 2021

## **6. CONCLUSION AND RECOMMENDATIONS**

### **6.1. Conclusion**

There is a close positive relationship between *software development practices* and *software usability* of the in-house software produced. Performance of software usability test tends to ascertain the usability of the software, hence enhancing its actual performance in terms of *easy to use, quick job completion, flexibility and ability to generate correct output*; and having *satisfactory functions* with its *output being trusted* by the users and the corporate management.

### **6.2. Recommendations**

The paper recommends that more user involvement be accorded in the software development process so as to narrow the gap between what the software can do and the user expectations on the software produced and set to be used. The very gap, i.e. between user expectations on the newly introduced in-house software and the actual functions of the software, is what creates lower level of usability, hence turning software development process as ineffective and loss making to the organization. Furthermore training on users is emphasized for reducing user resistance so as to promote both in-house software awareness and usability

### **6.3. Contribution and Limitations of the Study**

#### **6.3.1. Contribution of the Study**

The study posits in-house software development process within the context of software usability, hence demonstrating the need for production of useful and efficient software products, through involvement of both software developers and users.

#### **6.3.2. Limitations of the Study**

This study was located in only one organization that operates in Tanzania, hence possibly raising issues of replication of results in other organizations and or countries. Variations on work environments and management of software development practices in various organizations may pose some questions leading to lack of uniformity regarding software development and usability to their respective targeted users.

## **REFERENCES**

- [1] Chaudhary, N. and Apadhyay, A. 2021. Usability Testing of Android Applications Survey and Case Study. Journal of Information Technology, Review Article. Retrieved from [2] Cole, B. 2014. What is software usability?. Rocket Softw. Blog. Retrieved from <https://blog.rocketsoftware.com/2014/11/software-usability>
- [3] Samwi, R. 2020. The Role of Web Usability Evaluation: An Illustrative Case Study of the User Centred Developed Web-Portal to Facilitate Tanzania's Higher learning Institutions Students Field attachment. Journal of Cooperative and Business Studies. Vol. 5, Issue 2, 2020
- [4] Bevan, N., Carter, J., Earthy, J., Geis, T., & Harker, S. (2016) New ISO Standard for Usability, Usability Reports and Usability Measures. SO-9241-11 Retrieved from <file:///C:/Users/ABCD/Downloads/Bevan2016.NewISOStandardsforUsabilityUsabilityReportsandUsabilityMeasures.pdf>
- [5] Abran, A., Khelifi, A., Suryan, W., Seffah, A., 2003. Usability Meanings and Interpretations in ISO Standards.. Software Quality Journal 11, 325–338. <https://doi.org/10.1023/A:1025869312943>
- [6] Capilla, R., Kazman, R., Romera, C., and Carillo, C. (2020) Usability Implications on Software Architecture: the Case Study of Mobile apps. Wiley Online

- [7] Folmer, E., Bosch, J., 2004. Architecting for Usability: a Survey. *J. Syst. Softw.* 70, 61–78. [https://doi.org/10.1016/S0164-1212\(02\)00159-0](https://doi.org/10.1016/S0164-1212(02)00159-0)
- [8] Iacovou, C.L., Nakatsu, R., 2008. A Risk Profile of Offshore-outsourced Development Projects. *Commun. ACM* 51, 89–94. <https://doi.org/10.1145/1349026.1349044>
- [9] Lan, C.-H., Lan, K.-T., Chen, H.-M., Kuo, T.-S., 2004. Determining in-House/Outsourcing Activities of Hi-tech Project to Seek ROS Maximization. *J. Inf. Optim. Sci.* 25, 491–505. <https://doi.org/10.1080/02522667.2004.10699623>
- [10] Jennings, D., 2002. Strategic Sourcing: Benefits, Problems and a Contextual Model. *Manag. Decis.* 40, 26–34. <https://doi.org/10.1108/00251740210413334>
- [11] Boivie, I., Åborg, C., Persson, J., Löfberg, M., 2003. Why Usability Gets Lost or Usability in In-house Software Development. *Interact. Comput.* 15, 623–639. [https://doi.org/10.1016/S0953-5438\(03\)00055-9](https://doi.org/10.1016/S0953-5438(03)00055-9)
- [12] Nielsen, J., 1994. Usability Inspection Methods, in: *Conference Companion on Human Factors in Computing Systems - CHI '94*. Presented at the Conference Companion, ACM Press, Boston, Massachusetts, United States, pp. 413–414. <https://doi.org/10.1145/259963.260531>
- [13] Bak, J., Nguyen, K., Risgaard, P., Stage, J., 2008. Obstacles to Usability Evaluation in Practice: A Survey of Software Development Organizations, in: *Proceedings of the 5th Nordic Conference on Human-Computer Interaction: Building Bridges, NordiCHI '08*. Association for Computing Machinery, Lund, Sweden, pp. 23–32. <https://doi.org/10.1145/1463160.1463164>
- [14] Landauer, T. 1995. The trouble with Computers: Usefulness, Usability and Productivity. Retrieved from <https://journals.sagepub.com/doi/abs/10.1177/027046769801800324>
- [15] Jeng, J., 2005. Usability Assessment of Academic Digital Libraries: Effectiveness, Efficiency, Satisfaction, and Learnability. *Libri* 55. <https://doi.org/10.1515/LIBR.2005.96>
- [16] Blandford, A., Buchanan, G., Jones, M., 2004. Usability of Digital Libraries. *Int. J. Digit. Libr.* 4, 69–70. <https://doi.org/10.1007/s00799-003-0070-8>
- [17] Affairs, A. 2013. Usability Testing. [WWW Document]. URL [/how-to-and-tools/methods/usability-testing.html](http://how-to-and-tools/methods/usability-testing.html) (accessed 2.23.20).
- [18] Kumar, P. 2017. What is the Difference between Usability Test and User Acceptance Test? H2K Infosys. Retrieved from <https://www.h2kinfosys.com/blog/>
- [19] Mushashu, E. and Mtebe, J. 2019. Investigating Software Development Methodologies and Practices in Software Industry in Tanzania. Retrieved from <https://ieeexplore.ieee.org/abstract/document/8764884>
- [20] Mariage, C., Vanderdonckt, J., Pribeanu, C., n.d. State of the Art of Web Usability Guideline 26. Retrieved from [https://www.researchgate.net/publication/2934937Web\\_Usability\\_Guidelines](https://www.researchgate.net/publication/2934937Web_Usability_Guidelines)