OPTIMIZING THE DESIGN FOR USER EXPERIENCES IN DEVELOPMENT PROCESSES OF DESKTOP-BASED FREE AND OPEN-SOURCE SOFTWARE PROJECTS

PHESTO PETER NAMAYALA

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF DODOMA

OCTOBER, 2023

OPTIMIZING THE DESIGN FOR USER EXPERIENCES IN DEVELOPMENT PROCESSES OF DESKTOP-BASED FREE AND OPEN-SOURCE SOFTWARE PROJECTS

BY
PHESTO PETER NAMAYALA

A THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF DODOMA
OCTOBER, 2023

DECLARATION AND COPYRIGHT

I, **Phesto Peter Namayala**, declare that this thesis is my original work and that it has not been presented and will not be presented to any other university for a similar or any other degree award.

Signature:…………………………………………………….

No part of this thesis may be reproduced, stored in any retrieval system, or transmitted in any form or by any means without prior written permission of the author or the University of Dodoma. If transformed for publication in any other format, it shall be acknowledged that this work has been submitted for the doctoral award at the University of Dodoma.

CERTIFICATION

The undersigned certify that they have read and hereby recommend for acceptance by the University of Dodoma the thesis entitled "**Optimizing the Design for User Experience in Development Processes of Desktop-Based Free and Open-Source Software Projects"** in fulfillment of the requirements for the degree of the Doctor of Philosophy in Computer Science of the University of Dodoma.

Signature:………………………………… Date: 23RD OCTOBER 2023

Dr. Tabu S. Kondo (PhD)

SUPERVISOR

Signature:………………………………… Date: 23RD OCTOBER 2023

Prof Leonard J. Mselle

SUPERVISOR

ACKNOWLEDGEMENT

Many individuals and organizations have significantly contributed to completing this doctorate research. However, listing all supporters' names is unbearable due to space limits. Thus, the study recognizes only a few supporters and institutions as representatives.

Firstly, heartfelt gratitude goes to the supervisors of this study, Dr. Tabu S. Kondo and Prof. Leonard J. Mselle, for their continuous support, motivation, patience, and knowledge necessary to accomplish this study. Their leadership helped a lot in completing research activities, including thesis compilation, and I cannot imagine achieving this without their support.

Moreover, I would like to acknowledge the insightful support and remarks offered by the management of the College of Informatics and Virtual Education (CIVE), including the Acting Principal, Dr. Florence Rashid, the Computer Science and Engineering Head of the Department, Dr. Tabu S. Kondo, the current postgraduate Coordinator Dr. Mustapha H. Mohsin, the formal postgraduate coordinators Dr. Hassan Killavo and Dr. Majuto Manyilizu and other members of the staff mainly the principal reviewers of proposal and thesis. Nevertheless, I also extend my gratitude to other discussants and external reviewers for the tricky questions and leadership, which have helped advance my knowledge.

I want to express my heartfelt gratitude to my family, particularly my dear wife, Grace Peter Salia, daughters Clara Phesto Namayala and Estah Phesto Namayala, and son Brian Phesto Namayala, for daily encouragement, moral support, and motivation to complete the study. My thanks are also extended to my classmates, roommates, and other brothers and sisters for their psychological support throughout the research and in the production of this thesis. Nevertheless, I am highly indebted to the management of Mbeya University of Science and Technology for providing me with this fantastic chance to pursue a Doctor of Philosophy in Computer Science.

Above all, I thank my Lord God for making this long journey possible by always blessing, guiding, and inspiring me. As a Christian, I think all the beautiful and lovely things we obtain or have in life result from God's help, love, and kindness.

DEDICATIONS

This thesis is dedicated to my late parents, Macrina Jakob Manyika and Peter Mayanga Namayala. The dedications are also extended to my late brother Stanislaus J. Namayala. May their souls rest in peace.

ABSTRACT

The popularity of User Experience (UX) is exponentially increasing in every organization, including the Free and Open Source Software (FOSS) community. It is often the predictor of success as it assists in developing products that meet users' requirements. Although stakeholders in the FOSS community are beginning to realize the values of UX and gain a shared understanding, many FOSS projects do not value UX and inconsistently incorporate related activities in their strategic plans, causing them to have undesirable UX, being complex to use and poorly adopted.

Although server-based FOSS applications enjoy a significant market share, desktop-based applications are not well-performing compared to their counterpart proprietary software, mainly due to poor UX. Although examples of desktop-based FOSS projects with desirable UX exist, such as Mozilla Firefox and Emacs, there is still potential to improve many others by enhancing processes for incorporating UX-related tasks.

UX is an organizational endeavor and demands the entire organization's efforts. As a result, its assessment must check the organization's strength to carry out UX-related operations. However, little research exists on the UX assessment of FOSS projects. The fundamental contribution of this thesis is a model entitled the Free and Open Source Software User Experience Maturity Model (FOSS-UXMM) that measures organizational strengths and weaknesses in implementing UX-related activities. The model addresses the need to quantify how much FOSS initiatives support UX and propose initiatives to enhance UX aspects in the development life cycles of desktop-based FOSS projects.

The FOSS-UXMM mainly investigates the association between FOSS project UX maturity and UX maturity impacting factors. Its measuring instrument comprises factors from a completed literature review and experts' options collected using the Fuzzy Delphi Method (FDM). Before establishing the factors influencing the FOSS community's UX maturity that has been adopted to develop FOSS-UXMM, the study examined the applicability of present User Experience Capability Maturity Models (UXCMMs) in the FOSS community and identified ten lenses for measuring the FOSS community's preparation to achieve higher levels of UX maturity. It also

completed two case studies to validate and verify the proposed FOSS-UXMM. Finally, the FOSS-UXMM was appraised by comparing it with the selected UXCMMs. The findings indicate that the FOSS-UXMM is ideal and superior for assessing the UX maturity of the FOSS community because it contains FOSS-exclusive UX maturity influencing factors established by involving the right key stakeholders, it is tested with the actual projects and provides desirable and consistent results.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

ABBREVIATIONS AND ACRONYMS

AR	Augmented Reality
AI	Artificial Intelligence
ASQ	The After Scenario Questionnaire
CLT	Cognitive Load Theory
CHI	Computer-Human Interactions
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CSUQ	Computer Systems Usability Questionnaire
DEEP	Design-oriented Evaluation of Perceived Usability
DSR	Design Science Research
EMO-CheQ	Emotional Communication in Hearing Questionnaire
F/LOSS	Free/Libre and Open Source Software
F/OSS	Free and Open Source Software
FINERMAPS	Feasible, Interesting, Novel, Ethical, Relevant, Manageable, Appropriate, Potential value, Publishability, and Systematic
FOSS	Free and Open Source Software
FSF	Free Software Foundation
HCI	Human-Computer Interaction
IoT	Internet of Things
ISO/IEC	International Organization for Standardisation and the International Electrotechnical Commission
ISQ	Intranet Satisfaction Questionnaire
KPIs	Key Performance Indicators
MCDM	Multi-criteria Decision-making
ML	Machine Learning

OSI	Open Source Software Initiative
OSS	Open Source Software
PANAS	The Positive and Negative Affect Schedule
PSQ	Psychometric Evaluation of a Printer Scenario Questionnaire
PSSUQ	Post-Study System Usability Questionnaire
PUTQ	Purdue Usability Testing Questionnaire
QUIS	Questionnaire for User Interface Satisfaction
RERO	Release Early, Release Often
SDLC	Software Development Life Cycles
SE	Software Engineering
SPA	Software Process Assessments
SPCMMs	Software Process Capability Maturity Models
SPDX	Software Processes Capability Maturity Models
SPI	Software Processes Improvement
SquaRE	Software Quality Requirements and Evaluation
SUMI	The Software Usability Measurement Inventory
SUS	Software Usability Scale
TAM	Technological Acceptance Model
TCO	Total Cost of Ownership
UCD	User-Centred Design
UID	User Interface Design
UMUX	Usability Metric for User Experience
USE	Usefulness, Satisfaction, and Ease of Use
UTAUT	The Unified Theory of Acceptance and Use of Technology
UX	User Experience

UXA	User Experience Assurance
UXCMMs	User Experience Capability Maturity Models
UXD	User Experience Design
VR	Virtual Reality
UXPA	User Experience Professional Association
VisAWI	Visual Aesthetics of Websites Inventory
WAMI	Website Analysis and Measurement Inventory

CHAPTER ONE
INTRODUCTION

1.1 Overview

This chapter highlights the study's most critical issues. It covers several items that build the study's groundwork, including background information, theories, and findings from earlier studies. It also enlightens the research problem statement, objectives, research questions, study's significance, contributions, scope, limitations, and delimitations. The chapter ends with definitions of critical terminologies and a thesis layout.

1.2 Background Information

Free and Open-Source Software (FOSS) projects must be useable by individuals outside the development team or keepers (Berendes et al., 2022). According to AlMheiri et al. (2018), FOSS projects are ideal for startups, government agencies, students, and ordinary users, particularly in developing countries where meeting software acquisition costs has been challenging. By accommodating low Total Cost of Ownership (TCO) (Bwalya, Akakandelwa, & Dobreva-McPherson, 2019; Dhir & Dhir, 2017; Namayala et al., 2022; Patino-Toro, Valencia-Arias, Gomez-Molina, & Bermeo-Giraldo, 2022; Sanchez, Ayuso, Galindo, & Benavides, 2020), adopting FOSS projects was predicted to be high for desktop and server-based applications. However, the situation is far from ideal. Desktop-based FOSS applications face low acceptance globally (Colombo, Piva, & Rossi-Lamastra, 2014), regionally (Masson et al., 2017; Nagy, Yassin, Bhattacherjee, Spinellis, & Giannikas, 2012; Oreku & Mtenzi, 2013) and locally (MWTC, 2016; Oreku & Mtenzi, 2013).

Although several factors are predicted to influence the adoption of FOSS projects, available evidence acknowledges User Experience (UX) as critical in deciding the adoption rate (Berendes et al., 2022; Cheng & Guo, 2018; Dhir & Dhir, 2017; Masson et al., 2017). When interacting with products, especially FOSS projects, users seek a positive UX (Hinderks et al., 2022). The FOSS community user base has grown beyond technically equipped, contradicting its culture and architecture, and its projects are applied in critically based missions. As a result, the popularity of UX has also increased in the community.

UX comes with several advantages to the software developing communities. For example, it often envisages users' engagement by satisfying needs and expectations (Cheng & Guo, 2018; Magin, Maier, & Hess, 2015; Mtebe, 2019). It determines the success of products, augments revenue, satisfies employees, and predicts user errors and mandatory support. Thus, maintaining a robust UX design that can deliver products with desirable UX and acquire competitive advantages over rivals is a requirement for every organization (Hokkanen & Väänänen-Vainio-Mattila, 2015; Unterkalmsteiner et al., 2016).

The robust UX design requires the pledged commitment of the entire organization that may involve dedicating resources to consistently and appropriately integrate UX methods, practices, principles, and tools into the software development lifecycle (SDLC) (Chapman & Plewes, 2014; Cheng & Guo, 2018; Stone, Bentley, White, & Shebanek, 2016). However, the FOSS neither allocates adequate UX resources (Llerena, Rodriguez, Llerena, Castro, & Acuña, 2018) nor equally values UX-related tasks (Wale-Kolade & Nielsen, 2016), raising the question of why. The FOSS community must have relevant initiatives and tools to respond to this question. In addition, UX has not secured a position in traditional SDLC, leaving Human-computer interactions (HCI or CHI), Software Product Quality Requirements and Evaluation (SquaRE), and Software Engineering (SE) discussing efforts to optimize the design for UX in the FOSS community.

Current knowledge supports the argument that correctly implementing user research and adopting User-Centred Design (UCD) practices may create robust UX designs (Saad et al., 2021). UX is also a result of the ideal implementation of Human-Centred Design (HCD) because it has its roots in human factors and ergonomics (Edwards, Huber, & Kramarova, 2020). These fields have focused on the interaction between human users, machines, and contextual environments to design systems that address the user's experience since the late 1940s. However, unlike other software-developing communities, the FOSS community has a different structure and culture (Crowston & Howison, 2005; Nagunwa & Lwoga, 2012), preventing the straight adoption of existing UCD, HCD, and HCI methods. For example, it supports distributed developers and asynchronous internet-based communication and prioritizes functionality and source codes' beauty rather than UX (Butler et al., 2022; Raza et al., 2012a).

Although the FOSS community has started to have some understanding of UX and its benefits (Da Silva, Silveira, Maurer, & Silveira, 2018; Hokkanen & Väänänen-Vainio-Mattila, 2015; Nguyen-Duc, Kemell, & Abrahamsson, 2021), it often adopts ad hoc interventions for optimizing the design for UX which lack formal protocols and they are of unknown quality (Kurosu, 2017; Ovad & Larsen, 2015; Rajanen & Iivari, 2015b). Many of these interventions are developed based on the authors' years of working experience, and they inefficiently manage UX methods, tools, practices, and principles (Namayala et al., 2022). Finally, most of them are generically developed, missing multiple dynamics of the FOSS community (Raza et al., 2012a).

Best design for UX and evaluation practices recommend tools, methods, practices, and principles that account for organizational dynamics and other Multi-Criteria Decision-Making (MCDM) issues (Ali, Li, Khan, Zhao, & Li, 2018). These recommendations have shifted the paradigm in the design for UX, where current initiatives are adopting Artificial Intelligence (AI) and Machine Learning (ML) (Donahole, 2021; Yang, Wei, & Pu, 2020) to support designers in making data-driven design decisions (Yang et al., 2020). However, the shift is not much discussed. Thus, improved initiatives, strategies, and tools that may evaluate and optimize the design for UX are still missing (Lacerda & von Wangenheim, 2018; Namayala et al., 2022; Novák, Masner, Benda, Šimek, & Merunka, 2023; Raza, Capretz, & Ahmed, 2012a; Terry, Kay, & Lafreniere, 2010).

Nevertheless, the FOSS community has very few empirically developed and tested factors, tools, methods, practices, and principles to support systematic design for UX and appraisal (Andreasen, Nielsen, Schrøder, & Stage, 2006; Mashapa, Chelule, Van Greunen, & Veldsman, 2013; Nichols & Twidale, 2006). Moreover, it lacks standardized lenses to assess how its projects are prepared to deliver better UX design (Hasan & Al-Sarayreh, 2015; Nichols & Twidale, 2006). Otchere (2022) argued that without the proper preparation for UX maturity, FOSS projects risk holding sessions that may not provide the necessary insights into UX design, which could waste time and effort.

Generally, the FOSS community does not know what creates a robust design for UX to deliver projects with desirable UX (Buis, Ashby, & Kouwenberg, 2023; Kocaballi, Laranjo, & Coiera, 2019; Namayala et al., 2022). This study, therefore, offers contributions that may improve the design for UX in the FOSS community development chain and guarantee products with desirable UX. Its scope covers the development of a novel Free and Open Source Software User Experience Maturity Model (FOSS-UXMM). While developing FOSS-UXMM, this study investigated the relevance of contemporary User Experience Capability Maturity Models (UXCMMs) in FOSS projects, and it identified lenses that can help analyze FOSS project preparation to achieve higher levels of UX maturity. In addition, it empirically determined factors influencing FOSS community UX maturity, used the identified factors to create FOSS-UXMM, and compared FOSS-UXMM with selected UXCMMs to determine its scalability.

1.2.1 Overviews of FOSS Concepts

FOSS refers to computer programs with licenses allowing programmers to view and alter their source code to fix bugs, improve functionalities, or adapt to fit their needs (Nagy et al., 2012; Pickett, 2019). Ethically and socially, FOSSs are superior to proprietary software (FSF, 2011; Kuwata, Takeda, & Miura, 2014). They certify four mandatory users' freedoms that allow the use of the software for any purpose, modify source codes to fit needs, redistribute the software without authors' consent, and improve the source codes (Free Software Foundation, 2012; Srinivasa, 2017; Stallman, 2002). By citing the Foundation for Open-source Software (FSF) and Open Source Initiatives (OSI), the word 'free' implies the liberty of using the software but does not necessarily mean the software is costless.

There are two major categories of FOSS based on the mode of installation. Those running on servers are server-based applications, and those running on personal devices are desktop-based. For example, MySQL, Linux Server Operating Systems, Apache HTTP Server, Nginx, Apache Tomcat, and Lighttpd are examples of Server-based apps. In contrast, Apache OpenOffice, LibreOffice, NeoOffice, and the Linux desktop operating system are examples of desktop-based applications.

Although this study uses the acronym FOSS to represent Free and Open Source Software, several other acronyms exist, including OSS, F/OSS, and FLOSS. Choosing the correct acronym is another challenge in the literature. However, adopting the FOSS acronym ensures neutrality and reduces other well-known complications and license problems (Colford, 1986; Stallman, 2002; Tsai, 2008). Architecturally, FOSS contains two terms: open-source and free software (see **Table 1.1**) for their explanations. However, literature links two other words to FOSS: freeware and public domain. Ideally, open source, free software, freeware, and public domain undertake assorted meanings (Chełkowski, Jemielniak, & Macikowski, 2021).

Table 1.1: Comparing Terms Linked to FOSS

Concept/item	**free software**	**open-source**	**freeware**	**public-domain**
Descriptions	Discuss the liberty of using the software	It is more than the accessibility of the source code.	Owners restrict prices and freedom of use	The software belongs to the public
Philosophy	Social	Methodology	Marketing	Copyrights
Rules	Four freedoms	Open-Source initiatives	N/a	Creative common organization
Obtained free of charge	Not compulsory	Not compulsory	Yes	Yes
Copyright	Yes	Yes	Yes	No
Instances	Ubuntu	Apache	Skype	SQLite

The reputation of FOSS projects, particularly server-based projects, and the FOSS source codes commonly adopted in Component-Based Development (CBD) is mounting exponentially worldwide (Steinmacher, Treude, & Gerosa, 2019). Recent estimates indicate the FOSS source code makes up 80–90% of all contemporary software (Nagle & Hoffman, 2020). As a result, current patterns show that developers, institutions, contributors, donors, and end-users are turning to FOSS projects as an alternative to proprietary software (Abramova, Pires, & Bernardino, 2016; Bahamdain, 2015; Menéndez-Caravaca, Bueno, & Gallego, 2021; Shekgola, Maluleka, & Rodrigues, 2021).

Unlike the original traditions of the FOSS community, the FOSS community is now witnessing several changes. For example, its user base is now fully-fledged beyond technically equipped users (Llerena et al., 2018). Likewise, several critical-based missions where a minor failure in operating FOSS could have catastrophic effects on people's lives also use FOSS projects (Iivari, Hedberg, & Kirves, 2008; Raza et al., 2012a).

The FOSS community has several obsolete assumptions that jeopardize its future research from operating in updated knowledge (Marois, Marsan, Carillo, Stol, & Fitzgerald, 2022). However, Spinellis (2019) argues that with today's technological advancement in software development, if most of the code in somebody's product is not FOSS, they are likely wasting effort and cash reinventing the wheel. The 10th Annual Future of Open Source Survey showed that 65% of the organizations surveyed had leveraged FOSS to accelerate software development (Taylor & Dantu, 2022). Therefore, FOSS supports establishing domestic software industries (Oreku & Mtenzi, 2013; Sarrab & Rehman, 2014). FOSS accelerates technological innovations (Silva, Coutinho, & Costa, 2023) and offers solutions to software piracy to some extent, as it is not a complete solution. It provides methods for reducing the time and resources required for resource-intensive software development life cycles (Badampudi, Wohlin, & Petersen, 2016; Kazimierczak, Breckwoldt, & Wajsman, 2020; Petersen et al., 2018; Spinellis, 2019).

Several researchers predict that FOSS development methodologies may dominate the future of software development (Begel, Herbsleb, & Storey, 2012; von Krogh & von Hippel, 2006), so longer businesses and individuals continue to demand high-quality software at lower costs. FOSS is forecasted to become an increasingly attractive option for development. These predictions result from the effectiveness of FOSS development methodologies in delivering software that provides cost-effectiveness, collaborative approach, transparency, flexibility, security, and global reach

(Krishnamurthy, O'Connor, & O'Connor, 2013; Napoleao, Petrillo, & Halle, 2020; Pinto, Steinmacher, Dias, & Gerosa, 2018).

Several organizational strategies consider FOSS as something to invest in and work on for businesses to adopt (Butler et al., 2022). FOSS supports multiple subdivisions, which include health, education, and economics (Androutsellis-Theotokis, Spinellis, Kechagia, & Gousios, 2010; Poizot & Méar, 2010; Tsou & Smith, 2011). Optimally, it delivers other technological possibilities such as reliability, stability, and flexibility (Colombo et al., 2014; Nagy et al., 2012; Statista, 2021; Zaffar, Kumar, & Zhao, 2011). Different scholars express the structure of the FOSS community in different ways. For example, Deprez and Alexandre (2008) represent the FOSS organization structure as flat with worldwide distributed stakeholders. Long and Siau (2009) describe it as a layered structure commonly known as the core-periphery structure. However, the core-periphery structure is widely accepted (Amrit & Van Hillegersberg, 2010; Barcellini, Détienne, & Burkhardt, 2014).

The FOSS community does not exist without limitations. For example, it lacks a defined development structure (Napoleao et al., 2020) and economic incentives where stakeholders work voluntarily and individually in isolation (Ferraz & dos Santos, 2022). These limitations account for problems that limit the eventual adoption of UX/HCI practices and several other quality-related initiatives.

1.2.2 Synopsis of UX

UX is a relatively new concept (Law & Van Schaik, 2012; Wallach, Conrad, & Steimle, 2017) and is considered the fifth generation in the HCI domain (Law, 2011; Yong, 2013). Several studies still disagree on UX definition and measurement, and its relation to software development and how to collect user feedback is still uncertain (Law & Abrahão, 2014). UX offers different connotations based on who is asked, how, when, and in what context (Araz, 2018; Forlizzi & Battarbee, 2004; Law & Abrahão, 2014). As a result, UX is often a complicated, confusing, and ambiguous statement virtually unbearable to design (Hellweger & Wang, 2015; Law & Abrahão, 2014).

Several efforts that offer the definitions of UX exist. For example, Roto et al. (2010), Mashapa (2013), Hassenzahl and Tractinsky (2006), Nielsen and Norman (2017), Anthony et al. (2016), and U S Department of Health and Human Services (2015). However, two definitions are widely adopted: the ISO/IEC (2019a) and Hassenzahl and Tractinsky (2006). The ISO/IEC (2019a) describes UX as: "*the person's reactions and responses to using or expecting to use a particular system, product, or service*." Likewise, Hassenzahl and Tractinsky (2006) define UX as "*A consequence of a user's internal state (predispositions, expectations, needs, motivation, mood, etc.), the characteristics of the designed system (e.g., complexity, purpose, usability, functionality, etc.) and the context (or the environment) within which the interaction occurs (e.g., organizational/social setting, the meaningfulness of the activity, voluntariness of use, etc.)*." Unfortunately, both definitions are still silent on the metrics or methods required to comprehensively assess UX maturity (Hinderks, Schrepp, Domínguez Mayo, Escalona, & Thomaschewski, 2019; Law & Abrahão, 2014).

Despite having different definitions, practitioners, researchers, and other stakeholders admit that UX is a broader concept regarding the systems' interactions and often goes beyond standard usability and functionality (Bach, DeLine, & Carroll, 2009; Hassenzahl, Burmester, & Koller, 2021; Law & Abrahão, 2014). It covers all aspects of product-user interactions (Nielsen & Norman, 2017), including usability issues, but only as a sub-element of successful UX (Hassenzahl et al., 2021). It also accommodates users' feelings and emotions (Allam, Che Hussin, & Dahlan, 2013; Cheng & Guo, 2018; Hartson & Pyla, 2012; Magin et al., 2015; Mtebe, 2019).

UX is a highly multidisciplinary subject (Forlizzi & Battarbee, 2004; Roto et al., 2010), with much confusion linked to embraced specialties or design backgrounds (Santoso, Schrepp, Yugo Kartono Isal, Utomo, & Priyogi, 2016). It is also holistic, dynamic, subjective, and situational (Bosley, 2013; Hassenzahl, 2010; Hassenzahl & Tractinsky, 2006; Karapanos, Zimmerman, Forlizzi, & Martens, 2009; Law, van Schaik, & Roto, 2014; Vermeeren et al., 2010). Based on perspective, different fields approach UX differently (Hellweger & Wang, 2015). Nevertheless, UX has not adequately found its way into current HCI practices, Interaction Design, and Usability Engineering (UE) literature (Karapanos et al., 2009).

Nevertheless, the present body of knowledge supports the argument that UX consists of pragmatic (instrumental qualities), hedonic (non-instrumental attributes), and the consequences of user-product interactions (resulting from the user’s emotional reaction) (Hassenzahl, 2018; Hinderks et al., 2019; Magin et al., 2015). Meanwhile, Forlizzi and Ford (2000), Roto (2007), and Arhippainen (2009) argue that UX consists of user, system or product, or artifact and interaction contexts. This argument has widespread acceptability among stakeholders.

Nevertheless, based on the author’s analysis of current UX definitions and other authors' views, Mashapa's (2013) study identified tasks as another UX component to create four UX components. However, from the analysis performed by this study, one more aspect was added due to the existence of different types of UX caused by time differences. Moreover, the task element Mashapa et al. (2013) identified was converted to goals or objectives. Thus, this study acknowledges five components of UX: users, product or artifact, context, goals or objectives, and period (see **Figure 1.1**).

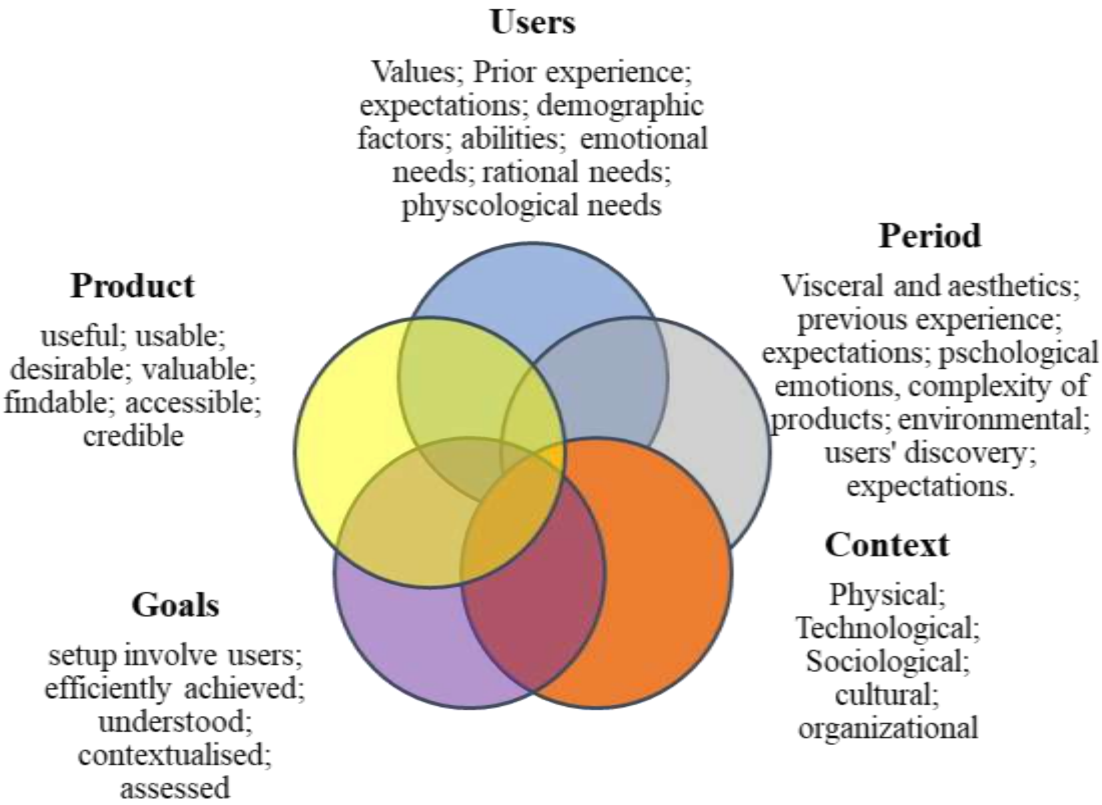

Figure 1.1: Proposed Elements of UX

SOURCE: Researcher's Interpretation

1.2.2.1 Users

Users are the focal point of UX. According to Mashapa (2013), UX would not exist without users. UX design must consider users' emotional and rational needs, values, prior experience, demographics, product-use abilities, goals, and psychological conditions (Hess, 2009). The total UX is a summative reflection of the user's anticipatory, instantaneous, episodic, and long-term UX (Hassenzahl & Tractinsky, 2006). On many occasions, users' expectations are the drivers of UX, as illustrated in the UX life cycle by Hiltunen et al.(2002) (see **Figure .1.2**)

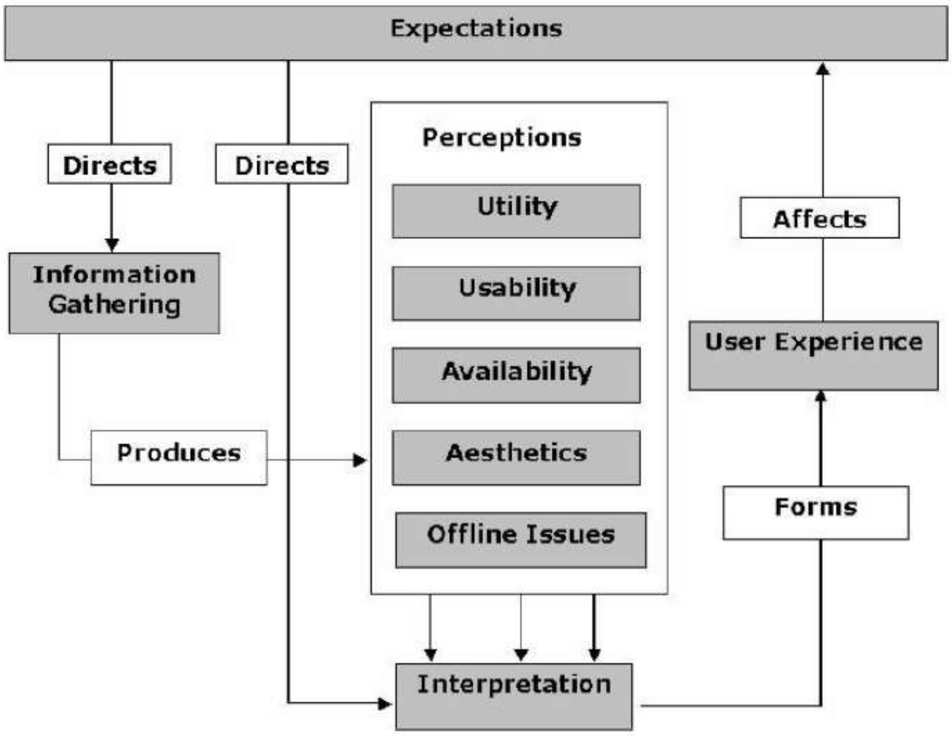

Figure .1.2: **UX Life Cycles**

SOURCE Hiltunen et al.(2002)

1.2.2.2 Product

UX is incomplete without a product/system or artifact because it results from users interacting with it, among other things (Roto, 2007). As a result, several UX-related features from a product perspective affect UX (Mashapa, 2013). Present literature explains product-related UX influencing factors, including Morville's (2004) "The user experience honeycomb," which identifies seven facets: useful, usable, desirable, valuable, findable, accessible, and credible. Products must partake in these facets to provide a positive UX. Each factor from Morville (2004) is associated with a special meaning in UX. Useful, for instance, helps users achieve their goals, and usable enables users to carry out specific tasks. Valuable offers desired benefits; findable makes information easily accessible; desirable is aesthetically pleasing; accessibility helps users with disabilities, and credible inspires trust in the users.

1.2.2.3 Context

The user-product interaction context, which spans widely and may encompass the technological and physical components and the values and culture of an individual or organization, is another essential component of UX (Hess, 2009). According to Arhippainen’s (2009) Usability and User Experience Framework (U2E-Frame), user-product interaction contexts are established based on the user’s psychological state (Emotion, Propensities). Others include the social environment in which the product is used, aspects of organizational culture, physical characteristics, and technical concerns (appearance, scalability, and operation platform).

Finally, the statement that meeting users’ context-specific UX requirements determines UX quality (IDF, 2018) is often valid. However, it is still challenging to adequately identify the context-specific UX ingredients (Hogan, Soutar, McColl-Kennedy, & Sweeney, 2011; Kocaballi et al., 2019; Kremer, Schlimm, & Lindemann, 2017; Law et al., 2014; Zarour & Alharbi, 2017).

1.2.2.4 Goal

This element was also addressed as the task by Mashapa (2013). However, this study has found it appropriate to refer to the element as an objective or goal because users’ purposes are not to complete tasks but rather objectives or goals that they expect to accomplish and may contain one or more tasks. In actuality, users’ expectations can be to efficiently fulfill a purpose while also feeling satisfaction from using an artifact. Users’ goals are crucial to designing a great UX, and if they are misconstrued or taken out of context, they could make the system or product ineffective or inappropriate and fail to give a positive UX.

Many frameworks explain how to analyze users’ goals as they engage with products, such as Norman’s (2016) Execution/Evaluation Action Cycle (EEAC) Framework (see **Figure 1.3**), which outlines the user’s involvement, including setting up their goals, achieving them, and assessing their performance.

Figure 1.3: The Execution/Evaluation Action Cycle (EEAC) Framework
SOURCE: Norman (2016)

Generally, according to the study by Ahmad and Ali (2013), a product should always encourage users' utilization and meet their goals, build trust, and elicit favorable feelings.

1.2.2.5 Period of use

The perception of users toward a particular product may change from positive to negative, and vice versa is true after prolonged use of the product. This argument is supported by the study of Karapanos (2010) and Roto et al. (2010), who commented that individuals, situations, time, and products influence UX.

The types of UX based on timing bring other challenges to the FOSS community that demand identifying which UX is more critical for the community's success. Is it acquired before, during, or after interacting with products? According to the adopted UX Factor Diagram (UXFD) by Mashapa et al. (2013), there are five well-known types of UX based on period: anticipated UX gained before interacting with a product, momentary UX experienced while interacting with a product, episodic UX experienced soon after interacting with a product, and cumulative UX acquired over time (see **Figure 1.4**).

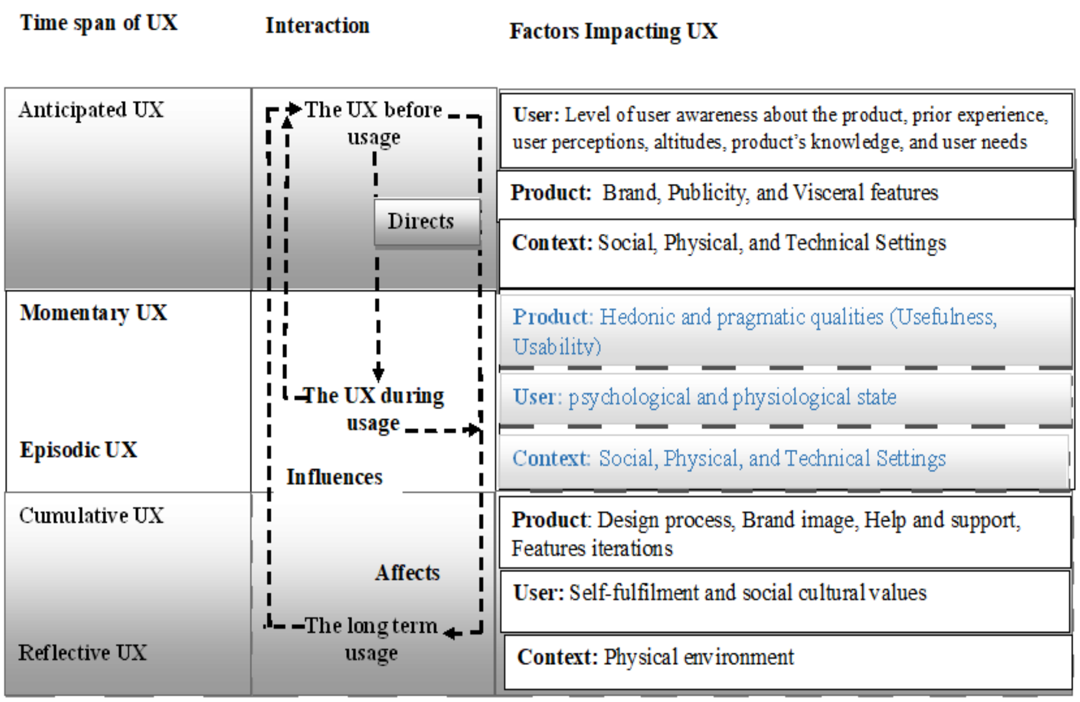

Figure 1.4: UX Factors Diagram

SOURCE: Mashapa et al. (2013)

1.2.3 Overview of User Experience (UX) and User Interface Design (UID)

Unlike UX, User Interface (UI) specifies all of the interactive system's components (software or hardware), which give information and ways to control it and enable users to complete specific tasks with the assistance of an interactive system (International Organisation for Standardisation (ISO), 2018; Johnson, 2014). It is an essential step in the software development process, and it is estimated that developing and implementing the user interface accounts for up to 48% of the project work (Myers & Rosson, 1992).

User Experience (UX) and User Interface (UI) design are linked and sometimes confused and used interchangeably, but they are not identical. However, designing a good application that suits the users' needs requires input from UI and UX designers (Adhitya, Andreswari, & Alam, 2021). UI design is often regarded as part of the UX design process and concerns the development of the actual product or service interface. It finalizes products and their designs for actual user engagements, is primarily concerned with how a product looks and operates, and is detail-oriented. In contrast, UX design manages the user journey as they interact with a product or service to deliver meaningful UX through various product development domains such as branding, usability, function, and design (Bilousova, Gryzun, & Zhytienova, 2021). UX design is a complicated and ambiguous process (Kompaniets, Lyz, & Kazanskaya, 2020), and it is mainly concerned with the overall feeling of the products' experience, prototyping that creates wireframes and personas forming products' basis, and viewing the product from a high level.

1.2.4 Overviews of UX in the FOSS Community

In the FOSS community, it is unclear what makes UX (Hassenzahl, Wiklund-Engblom, Bengs, Hägglund, & Diefenbach, 2015; Law et al., 2014). It is also unclear if the community design UX or design for UX (Karahanoğlu & Bakırlıoğlu, 2020). However, like any other software industry, UX may drive the FOSS business strategies (Bryan, 2012; Goyal, 2017; Robbio, 2019) and help the FOSS community succeed. There are several examples showing how other businesses have grown because of UX. For example, Apple's iPhone defeats smartphone manufacturers, Uber dominates the taxi industry, Facebook dominates social networks, Dyson dominates vacuum cleaners, and Tesla dominates electric vehicles.

UX can potentially affect the success of FOSS Projects in the FOSS community. However, many projects still have depraved UX (Bian & Zhang, 2012; Botha, Herselman, & Van Greunen, 2010; Chapman & Plewes, 2014; Liedtke, Jolanta Welfens, Rohn, & Nordmann, 2012; Mashapa et al., 2013). The FOSS community does not give deserved priority to UX (Cheng & Guo, 2018; Raza et al., 2012a; Raza, Capretz, & Ahmed, 2013). and, in most cases, participants' experiences and opinions influence the deliberations of UX-related tasks (Cheng & Guo, 2018; Lacerda & von Wangenheim, 2018; Raza et al., 2012a; Rukonić, Kervyn de Meerendré, & Kieffer, 2019; Sauro, Johnson, & Meenan, 2017).

Unlike other software-developing communities, the FOSS community cannot quickly adopt available HCI methods (Bornoe & Stage, 2014). Because its structure restricts the holistic engagement of HCI practices (Bach et al., 2009; Benson, Muller-Prove, & Mzourek, 2004; Twidale & Nichols, 2005). Terry et al. (2010) have indicated that existing UX/HCI practices may not be used in the FOSS community unless adapted to meet the community's needs and dynamics.

Within the FOSS community, UX is not a novel concept. It has been a standard term for decades and has been discussed in HCI and SQuaRE (ISO/IEC, 2011). However, adopting their practices has been difficult (Bornoe & Stage, 2014). UX practices always require pledges from all stakeholders and considerations of manifold dynamics: product interactivity, context, personal characteristics, users' knowledge, emotions, and moods (Cheng & Guo, 2018; Kronbauer, Machado, & Santos, 2015). They also require understanding the dynamics of all UX stakeholders, such as how they think, act, and get motivated (Terry et al., 2010). Unfortunately, it is challenging to commit these pledges in the FOSS community.

Chapman and Plewes (2014) quoted Cordell-Ratzlaff's speech in 2010, saying, "*I think to design great products, you need to have the culture in place.*" This quote emphasizes that UX is an organizational effort that demands organizational evaluation to streamline its implementation. Current practices suggest using the Capability Maturity Models for Software Processes (SPCMMs), such as The Capability Maturity Model Integration (CMMI) by the CMMI Product Team (2010) and ISO 15504 by ISO/IEC (2004) to evaluate the design for UX. However, neither ISO 15504 nor CMMI has processes to assess the FOSS community's UX (Lacerda & von Wangenheim, 2018; Petrinja & Succi, 2012; Raza et al., 2012a).

1.2.5 Overviews of FOSS UX Capability/Maturity

The UX assessment community has developed several generic UX Capability Maturity Models (UXCMMs) to incorporate processes of UX assessment and simplify complexities of widely accepted SPCMMs, such as CMMI and ISO 15504. However, no evidence supports their smooth applicability in the FOSS community (Lacerda & von Wangenheim, 2018; Petrinja & Succi, 2012; Raza et al., 2012a). In addition to this, none has been created exclusively for the FOSS community (Namayala et al., 2022).

Having UXCMMs exclusive to the FOSS community is extremely important to address community-specific issues and dynamics. The current literature shows that FOSS projects have grown essential in practically all industries. As a result, the public and private sectors, including many non-IT companies, heavily rely on them (Marois et al., 2022).

Developing the right tools to measure the FOSS community's UX maturity may guide their improvement. Citing Harrington's (2006) quote, "..*Measurement is the first step that leads to control and improvement. If you cannot measure something, you cannot understand it. If you cannot understand it, you cannot control it. If you cannot control it, you cannot improve it."* The study sees the necessity of proposing a methodology to assess the status quo of UX maturity in the FOSS community and recommends means for improving and advancing to the following maturity levels. The study assumes that the proposed method may optimize the design for UX in the FOSS community.

According to the Oxford Dictionary, Optimization means *"Finding the best possible solution to a problem."* As a result, this study views optimizing the design for UX as implementing the best controls within the FOSS community to ensure developed products have consistent and desirable UX. As the path to follow for optimizing the design for UX, the study has investigated the applicability of contemporary UXCMMs in the FOSS community and examined if any available UXCMM is developed elusively for the community. It also investigated analytical lenses that may be applied across FOSS projects to analyze the preparation of FOSS projects for UX maturity. Nevertheless, the study has researched factors affecting UX maturity in the FOSS community and used them to propose a tool that may be adopted to evaluate and improve the UX maturity of FOSS projects and help developers, project owners, and other core users from wasting resources developing projects that may not meet users' expectations and end up rejected.

1.3 A Statement of the Research Problem

Researchers and practitioners in the FOSS community consider desktop-based FOSS projects a natural choice and expect a full-fledged adoption (Kamau & Namuye, 2012; Shekgola et al., 2021). However, the situation is different on the ground. Most desktop computers still use proprietary software (Zhao, Liu, & Zhu, 2016). Data from Statcounter (2019) shows that Windows desktop operating systems enjoy 73.75% of the global market share, while Linux desktop operating systems enjoy only 2.09%. Nevertheless, the continued success of Microsoft Office serves as additional justification for the low uptake of desktop-based FOSS projects (Yeates, 2015). The continued poor adoption of desktop-based FOSS projects insinuates disappointments of other techniques endorsing their adoption (Taha, Abbood, Razzaq, & Al-Bahri, 2018).

According to Coutinho and Costa (2023), low quality and poor UX, incompatibility issues with other commonly used hardware and software, user reluctance, poor performance, community attitudes, and lack of support are among the causes of adoption misfortune. Moreover, Lenarduzzi et al.(2020) indicated a lack of detailed and up-to-date documentation as an additional cause of the desktop-based FOSS projects' adoption misfortune. However, poor UX is a recurring cause of users rejecting desktop-based FOSS projects (Berendes et al., 2022; Cheng & Guo, 2018; Dhir & Dhir, 2017; Masson et al., 2017). As a result, improving the UX of desktop-based FOSS projects may help to solve adoption-related problems and encourage users and other stakeholders to use FOSS projects.

Currently, FOSS projects offer little evidence of attempts to satisfy the persistent requirements of users. They also have limited data on the right ingredients for UX methods, tools, practices, and principles (Bargas-Avila & Hornbæk, 2011; Chapman & Plewes, 2014; Pretorius, Hobbs, & Fenn, 2015; Sauro et al., 2017).

UX is an organizational effort (Chapman & Plewes, 2014), so the FOSS community must understand how to improve, adapt, and integrate organizational UX valuation methods. Contemporary practices recommend adopting UXCMMs that check corporate compliance with UCD design practices. The main objective of UXCMMs is to assess organizational strengths and weaknesses regarding UX-related capabilities and suggest ways for improvement. However, the setup of the FOSS community only esteems stakeholders with technical knowledge who might be unable to implement UCD design practices smoothly. Nevertheless, present UXCMMs are generically developed through owners' years of working experiences and are not thoroughly evaluated across various situations and domains, making their universal applicability dubious (Lacerda & von Wangenheim, 2018; Namayala et al., 2022; Raza et al., 2012a). As a result, they may not be relevant or inadequately account for the FOSS community dynamics.

UX-matured organizations typically have professional UX designers who can offer UX-focused plans (Pernice, Gibbons, Moran, & Whitenton, 2021). To achieve higher levels of UX maturity, the FOSS community must focus on quality, consistent research, well-planned design procedures, resources, and tools. However, the community has a shortage of researchers and resources (Llerena et al., 2018; Raza et al., 2012a) attesting to the scarcity of empirically conducted UX-related studies (Carvajal & Moreno, 2017; Lacerda & von Wangenheim, 2018; Peres & Meira, 2015; Rajanen & Iivari, 2013; Raza et al., 2012a). As a result, the FOSS community lacks excellent and easy-to-use UXCMM, mainly in the early stages of projects (Rukonić et al., 2019). Nevertheless, it is not well-informed about what factors influence its maturity and adopted UXCMMs may accommodate inadequate metrics or KPIs (Carraro, 2014; Mashapa et al., 2013; Pretorius et al., 2015).

As the prerequisite for a proper UX maturity assessment, the FOSS community must make necessary preparations and adjustments. Otchere (2022) specified that without the appropriate preparation, the UX assessment runs the danger of holding a session that does not provide essential insights into UX evaluation, which could waste time and effort. However, many projects do not prepare for assessment except for a few, such as Mozilla Firefox and Emacs. For example, it is still tricky for UX professionals and other practices to participate in many FOSS projects, but not as severe as decades ago (Inal et al., 2020). Finally, detailed and up-to-date documentation and user support are still challenging (Lenarduzzi et al., 2020; Raza et al., 2012a; Sauro et al., 2017).

Despite current initiatives, many projects still do not profoundly accommodate desired UX features (Cheng & Guo, 2018; Raza, Capretz, & Ahmed, 2012b). However, addressing UX issues is critical to uplifting projects' adoptions. Toward the solution lane, the study has provided techniques to the body of knowledge that may aid in understanding and resolving UX-related issues. UX professionals were recruited to validate the suggested strategy, and lastly, the advanced technique was validated in genuine FOSS projects using two separate case studies to justify its applicability.

1.4 Research Objectives

1.4.1 The General Objective

The main objective of this research was to develop a UX maturity model for optimizing the design for UX in the development processes of desktop-based FOSS projects.

1.4.2 Specific Objectives

The study proposed the following specific objectives to achieve the general purpose.

i. To examine the practical applicability of contemporary UXCMMs in desktop-based FOSS projects
ii. To determine analytical lenses for analyzing the preparation of the FOSS projects for UX maturity.
iii. To establish UX maturity influencing factors of the desktop-based FOSS projects.
iv. To develop the UX Maturity Model (FOSS-UXMM) for assessing the UX maturity of desktop-based FOSS projects using established UX maturity influencing factors.

1.5 Research Questions

The study has created six (6) research questions to lessen its complexity. Creating these research questions was guided by procedures proposed by Ratan et al.(2019). According to Cummings et al. (2001), research questions must also be Feasible, Interesting, Novel, Ethical, Relevant, Manageable, and Appropriate, abbreviated as FINERMAPS. Responses consolidated from RQ-1 address the specific objective one, and responses from RQ-2 address the specific objective two. Likewise, answers from RQ-3 address specific objective three. The fourth specific objective is handled by three research questions: RQ-4, RQ-5, and RQ-6.

Theoretically, research questions detail the problem statement, refine the study's issues, focus on the problem statement, draw a roadmap for data collection and analysis, and define the research context (Ratan et al., 2019). The thesis proposed the following research questions:

RQ-1 How applicable are the current UXCMMs in determining the UX maturity of desktop-based FOSS projects?

RQ-2 What analytical lenses may analyze the preparation of the desktop-based FOSS projects for UX maturity?

RQ-3 What factors (metrics) influence the UX maturity of desktop-based FOSS projects?

RQ-4 What tool can optimize the assessment of the UX maturity of desktop-based FOSS projects? How is it developed?

RQ-5 How applicable is the FOSS-UXMM in determining the UX maturity of FOSS projects compared to other generic UXCMMs?

RQ-6 How is the design for UX in desktop-based FOSS projects optimized?

Positive responses to the research questions above gave a rigorous process for developing novel methods for evaluating desktop-based FOSS projects' UX maturity/capability. They also provided a road map for achieving the study's overall purpose and measures to fill identified gaps in evaluating the FOSS community's UX maturity/capability.

1.6 Significance of the Study

Findings from this study have provided four types of insights into the UX maturity assessment technology, particularly in the FOSS community, which include contributions to knowledge, methods, research, and practice.

1.6.1 Contributions to Knowledge

This study's significant contribution is a model for assessing the UX maturity of the FOSS community that may account for its dynamics. It has added five aspects to the body of knowledge regarding the UX maturity assessment, particularly in the FOSS community. First, it has provided ten analytical lenses developed by the fuzzy Delphi Method for assessing how prepared the FOSS projects are to achieve higher levels of UX maturity. Second, it has empirically established factors influencing UX maturity in the FOSS community. Thirdly, it created a new model exclusive to the community and extensively tested it through case studies. Fourthly, the study has proposed additional components for UX, which has transformed UX into five features: user, period, product, goals, and context. The current body of knowledge acknowledges

UX has three parts: user, product, and context where interaction occurs. Finally, the study has submitted five manuscripts for publication (see **Appendix XII**) in highly credible and reputable journals and publishers. They are in different stages, from already published to a first review of the submitted manuscript.

The FOSS community may benefit from the knowledge contributed by getting more insights into the fundamental components of UX maturity, how to create user-friendly assessment procedures experimentally, and how to develop user-friendly tools, especially in the early stages of FOSS projects. Nevertheless, they may also help practitioners and other stakeholders learn how to engage all FOSS stakeholders, assembling UX maturity ingredients and pledging projects with desirable UX.

1.6.2 Contributions to Methods

The study has provided four methodology contributions to the UX assessment community in the FOSS community. Firstly, it proposed the new rating scale "Unknown" to avoid the complexity imposed by the existing scales. In this study, the "Unknown" scale represents the number of unknown statements in the Process Reference Models (PRMs) passing thresholds. Secondly, the study proposed a new PRM exclusive to the FOSS community, derived from the UX maturity influencing factors the survey found significant and agreed upon among UX maturity experts. Thirdly, the study demonstrated the triangulation of multiple research techniques and data collection tools to accomplish studies. The study adopted interview sessions, consultations, case studies, questionnaires, and a literature review to develop strategies for optimizing the design for UX. Finally, in one published article, the study proposed a framework for making the procedure for developing new Software Process Capability Maturity Models (SPCMMs) consistent.

1.6.3 Contributions to Research

The study identified several unexplored areas regarding UX maturity valuation best practices in the FOSS community, warranting additional research. Some of these areas were also mentioned by Ali et al.(2018), Jokela et al. (2006), and Lacerda and von Wangenheim (2018). The topics identified are sevenfold. First, as also proposed by Kuo and Liang (2012), Ali et al.(2018), and Sangaiah and Thangavelu (2013), developers of new UXCMMs must research how to apply fuzzy mathematics while developing new UXCMMs that may handle subjectivity, human biases, vagueness, and uncertainties. Second, future-developed UXCMMs must adequately address MCDM, as Ali et al. (2018) proposed. Third, more study initiatives must perform comparative analyses of all existing UXCMMs to uncover methodological jungles in their developments, characteristics, and behaviors. Fourth, present models lack theoretical backgrounds (Becker, Knackstedt, & Pöppelbuß, 2009; Monteiro & Maciel, 2020; Solli-Sæther & Gottschalk, 2010). Therefore, future researchers must develop new theoretical backgrounds, standards, or frameworks to validate them (Otto, Bley, & Harst, 2020; Pöppelbuß & Röglinger, 2011; Proença et al., 2013). Fifth, future researchers must conduct additional SLRs and analyses on contemporary UXCMMs to disclose detailed information. Sixth, they must test and validate contemporary UXCMMs in natural settings and demonstrate their practical applicability in writing. Finally, future researchers must further research how to develop FOSS-exclusive UXCMMs.

1.6.4 Contributions to Practices

In practical terms, this study was expected to assist the decision-makers involved in UX maturity assessment in the FOSS community in understanding better how to prepare, plan, assess, report, and improve UX maturity. It provided initiatives to enhance UX-related processes that may help develop desktop-based FOSS projects with desirable UX, well-documented, and better adoption. The study's findings drew a picture that would provide a more profound insight into the FOSS community's future and current UX maturity assessment initiatives. The image also highlights potential dangers to the accuracy and legitimacy of the FOSS community's UX maturity assessment by showing the absence of UX maturity models that account for the community's dynamics. It also identified areas needing further improvements and provided insights into promoting better UX practices.

1.7 Scope of the Study

The study only focused on building a model for improving UX design in desktop-based FOSS applications. It sought to investigate how to propose new methods for determining the UX maturity of FOSS projects, the relevance of the available UXCMMs, obstacles encountered, and how the adopted procedures can be improved to meet the FOSS community's dynamics. During the investigation, the study has identified ten analytical lenses for determining the readiness of the FOSS community to assess its UX maturity and the UX maturity influencing factors. It then developed, verified, and validated the developed model and, finally, decided on its effectiveness by comparing it with the selected UXCMMs.

Respondents for this study were purposely selected worldwide with the help of internet technology. The inclusion/exclusions criteria involved stakeholders who contributed to projects chosen or have published articles in UX maturity assessment. Data were collected from sixty-two active FOSS projects hosted in five open source code hosting facilities with activity levels of 90% and above calculated from the user ratings, mailing lists with at least 200 members in discussion forums, and a weekly download count of 100 and above.

The study adopted Google Forms technology to develop and host questionnaires and interview guides. The links to the Google forms were disseminated using discussion forums, mailing lists, personalized e-mails, social media, and other online collaborative software such as ZOOM.

1.8 Limitations and Delimitations of the Study

Although experienced and non-experienced researchers are still confused by the limitations and delimitations terms (Miles, 2017), their explanations are more direct. Limitations denote elements of the research that researchers have no control over, while delimitations are researchers' controlled elements that usually define the study's boundaries without affecting goals and objectives (Miles, 2017; Theofanidis & Fountouki, 2018). In other words, the study's limitations are those features of design or methodology that affect or influence interpretations of the results (Price & Murnan, 2004). Theofanidis and Fountouki (2018) listed several factors that cause limitations and delimitations in their literature review paper. They further argued that researchers must adequately acknowledge their studies' limitations, delimitations, and assumptions to enhance the quality and validity of their findings.

1.8.1 Study's Acknowledged Limitations

This study had several drawbacks, much like many other investigations. Despite several noteworthy and possibly significant findings, this study had sixfold limitations.

First, there was a delay and difficulties in getting respondents to respond to the online surveys. For example, when collecting data for identifying factors influencing UX maturity, the recruitment process occurred between mid-July 2021 and November 2022, with an initial response rate of 32.29% (31/96). Finally, 12 FOSS stakeholders agreed to participate and signed consent forms.

Delays in responding to online surveys were nothing new. Many other authors have reported low response rates, according to Adem et al. (2021), and debates about what researchers may do to persuade potential respondents to boost response rates are still active. However, multiple studies, such as Saleh and Bista (2017), identify rewards, trust, and costs as factors influencing online survey response rates.

Second, although similar findings may be observed in other software development communities, the FOSS community may be the only audience for the study's findings. The sampling technique used in the study lends more credence to this hypothesis. Nevertheless, according to Andrade (2020), online survey data may frequently have two severe methodological flaws: the population to which they are disseminated cannot be described, and respondents may self-select into the sample with biases. Nevertheless, due to variations in procedures, stakeholders, and motivations, it might also be more challenging to generalize and apply the methodology of this study to other software development communities across frontiers.

Third, being a cross-sectional study, its findings would be vulnerable to longitudinal effects. From the explanation provided by Hassenzahl and Tractinsky (2006), the product's aging and prolonged use highly influence UX. According to Bell (2014), the longitudinal impact is a shift within a sample linked to aging. Since this thesis resulted from a PhD program that took 36 months, it was impossible to thoroughly examine every variable over a long period to identify behaviors that take time to manifest.

Fourth, the academic community, interested parties, policy and decision-makers, and the general public benefit from published research findings. Moreover, publishing results is crucial in the research cycle and academic endeavors. However, the study was limited to only publishers acknowledged by the University of Dodoma's College of Informatics and Virtual Education (CIVE), which mostly contained the Science Indexed Journals (SCI), to publish its findings. These famous publishers, Springer, Taylor and Francis, IEEE Access, Elsevier, and Sage, often take longer than anticipated to publish results. Consequently, it took longer to disseminate the study's findings to a large audience, delaying stakeholders' ability to take advantage of the findings.

Although highly qualified, the study has relied solely on experts' opinions, which may lead to unrealistic results. Finally, the author could not locate a similar analysis as the baseline to draw a roadmap for collecting data. Lacking similar studies resulted in difficulties in finding ready-made data collection tools. Therefore, as also specified by Brutus et al. (2013), the study was compelled to use an exploratory rather than an explanatory research strategy.

The identified Limitations result in several consequences falling into four major categories. First, not all members of the software development community may benefit from the conclusions drawn from the data. Second, the study may not have considered how stakeholders viewed the identified UX metrics over time. Third, some of the findings from the survey that would have been important to stakeholders and other interested parties may not have been published. Finally, as other scholars such as Kaye et al. (2020) predicted, the study may lack effective communication and adopt complex ideas.

1.8.2 Delimitations of the Study

The study encountered additional difficulties besides those the author had no control over. The study has derived several delimitations from what Theofanidis and Fountouki (2018) proposed. According to Theofanidis and Fountouki (2018), delimitations primarily concern the study's theoretical underpinnings, objectives, research questions, variables, and study sample. Moreover, it examines the study's methodology and philosophical foundation.

First, because this study did not sample many projects due to the adopted sampling techniques, possible contributions from none of the selected projects' stakeholders may have been overlooked. Second, the study may have missed other significant aspects that may have contributed to the challenges of optimizing the design for UX in the development process of FOSS projects due to the adopted specific objectives and research questions. Third, the survey did not consider the respondents' geographic locations. Finally, the study's analysis did not consider how differences between FOSS projects would affect the information acquired about stakeholders' perspectives.

1.9 Study Assumptions

The study assumes that not all FOSS projects are unprepared to assess their UX maturity. However, conducting studies in UX maturity assessment may help improve the UX of more projects. Several projects have strict rules and procedures to ensure final desktop-based projects have desirable UX, for example, Mozilla Firefox and Emacs. Moreover, it assumes that consulted respondents were willing to participate in the survey and provided open-minded feedback without compensation. Finally, the study thought that the level of UX complexity in the chosen FOSS projects was the same.

1.10 Definitions of Operational Key Terms

This section of the thesis clarifies several vital terminologies that have special meanings in the study.

Optimization: The study adopted the definition provided by Google's English dictionary in the Oxford language, which says optimization is *"the action of making the best or most effective use of a situation or resource."*

UX Metrics: According to Tullis and Albert (2013), metrics signify measurements approach to certain phenomena. Therefore, they often capture qualitative aspects of UX (Wallach et al., 2017).

Software Process Assessments (SPAs) are the initiatives toward Software Processes Improvements (SPI) or The Capability Maturity Models for Software Processes (SPCMMs). They identify and characterize contemporary organizational practices to realize the strengths and weaknesses of an organization under review in controlling the developed product's cost, quality, and schedule (Emam & Goldenson, 2000; Tarhan & Giray, 2017). They also assist organizations in meeting business requirements and objectives (CMMI Product Team, 2010). The ISO 15504 (2004) and the ISO (2015) have further described them as *"the systematic evaluation of an organization's processes against a process reference model (PRM)."*

The Process Reference Model (PRM) Usually defines architecture and processes in the product chain. It provides information on domain-specific standard procedures and specifications to compare to production processes for process evaluation (ISO/IEC, 2004). Mapping organizational processes and PRM is necessary for a proper software process assessment (Tarhan & Giray, 2017).

Software Process Maturity: This is the extent to which defined processes are managed, measured, and continually improved (Dooley, Subra, & Anderson, 2001; Hoover & Stewart, 2020; Paulk, Curtis, Chrissis, & Weber, 1993; Sanchez, Delgado, Ruiz, Garcia, & Piattini, 2009). Software process maturity, among other things, ensures that process implementations are transparent, adequately documented, and highly effective. It usually helps the organization choose the best improvement strategy (Van Looy, De Backer, & Poels, 2011).

Software Process Capability: It predicts expected outcomes due to defined software processes (Paulk et al., 1993) and ensures that software development processes consistently deliver products with desired results. It further helps characterize processes' ability to achieve or contribute to present or forecasted organizational goals (de Bruin, Rosemann, Freeze, & Kulkarni, 2005; ISO/IEC, 2004, 2019a)

UX Maturity: It evaluates an organization's potential to apply UCD principles to its team dynamics, design methodology, and product development. (Pernice et al., 2021). It uses three pillars to evaluate organizations: culture, process, and product. To reach higher levels of UX maturity, the organization must demonstrate a culture of improving deliverables by integrating UCD's philosophies into products' design and development processes.

User-centered Design (UCD) is an iterative design approach in which designers concentrate on users and their needs by including users at all stages of the design process through various research and design methodologies to produce highly usable and accessible products. Five main principles guide UCD. (1) Early involvement of end-users in the design process (2) A clear understanding of the context of use and user needs in an attempt to align them with overall business requirements (3) Regular collection, analysis, and incorporation of user feedback through the product design cycle (4) Employing a user-centric approach to product development and delivery procedures and (5) Utilizing an iterative design process with a continual goal to improve user experience.

The Maturity Scales: They often contain maturity levels that show organizational maturity, where each maturity level is built based on the previous level (Lacerda & von Wangenheim, 2018). Typically, maturity scales use continuous or staged representations or continuous and staged representation.

The Capability Scales: They often contain capability levels and represent an increased order of implementing process capability, ranging from not meeting the purpose to meeting projected goals (Lacerda & von Wangenheim, 2018). Typically, capability scales use continuous or staged or both continuous and staged representations:

The rating scales: Lacerda and von Wangenheim (2018) defined a rating scale as *"an ordinal scale to measure the extent of the process achievement.*

Maturity Models: Paulk et al.(1993) defined the maturity model as "*The framework that describes an evolutionary path from ad hoc, chaotic processes to mature,* *disciplined software processes."* It is mainly a result of integrating several concepts, including process, process maturity, process capability, and process performances of software.

Mental model: This is a cognitive illustration of external reality. Jones et al. (2011) defined it as "the *personal, internal representations of external reality that people use to interact with the world around them."* Ideally, mental models are individually and internally constructed based on knowledge, perceptions, and personal experiences. They help with reasoning, decision-making, and predicting an individual's behavior (Jones et al., 2011).

1.11 The Organization of the Thesis

The third revised version of the University of Dodoma regulations and guidelines for postgraduate students mandates all PhD theses to have five chapters. As a result, this work is organized into five chapters. Details of each chapter are as follows:

Chapter One presents the study by explaining the study's introduction, background information, the research problem statement, objectives, research questions, significance, contributions, scope, limitations, delimitations, and assumptions. It also defines critical terminologies adopted throughout the study and outlines the structure of the thesis.

Chapter Two is allocated for the literature review regarding technology timelines that mirror the knowledge in measuring the UX maturity of the FOSS community. It comprises theoretical reviews, empirical reviews, a conceptual framework, and a summary. It further shows accomplishments, what is missing, and the consequences of completing this study. Generally, this chapter identifies gaps and proposes solutions for them.

Chapter Three presents the research methodology. It states the methods adopted to accomplish the research, instruments used to collect and analyze data, study locations, and steps for collecting data. It also discloses detailed plans for analyzing data and why they were selected. Moreover, the chapter discusses the study's validity, reliability, and ethical issues.

Chapter Four presents and discusses the findings. It tells readers about accomplished things and techniques tied to their accomplishments. It further portrays applied instruments, data collection and analysis methods, and the results. Moreover, the chapter explicitly presents the proposed UX assessment technique, metrics, and hypotheses. It further applies analysis techniques to analyze data established from several data collection tools and techniques.

Chapter Five provides a preamble and summarizes the study's conclusion and recommendations for future studies after discussing and interpreting the findings. It communicates personal insights resulting from interpreting the findings. Moreover, the chapter presented the researcher's reflections on the relationships between the study's questions and interpreted findings.

CHAPTER TWO
LITERATURE REVIEW

2.1 Overview

Chapter One explained the study's background information, relevance, research problem, objectives and questions, and thesis structure. This chapter provides a complete literature analysis that helps locate, synthesize, and assess available research concerning the timeline for optimizing the design for UX in the development process of desktop-based FOSS projects. Its primary objective is to show the contributions of other studies toward achieving goals, and the scope of this study mainly stresses enhancing the adoption of desktop-based FOSS projects. The chapter communicates contemporary challenges that limit the adoption of current HCI practices in the FOSS community.

Substantial literature reviews usually build the foundation for proper research (Boote & Beile, 2005). It also provides a better understanding of existing techniques, concepts, and theories that may help investigate and uncover answers to the study's research questions. However, it does not mean educating readers about the study area's state of the art (Locke, Spirduso, & Silverman, 2013). According to Rifkin (2002), a literature review maps fields' knowledge evolution, accomplished, adopted, emerging, and the current state of thinking regarding assessing the UX of the FOSS community. It further identifies research gaps for further studies and proposes their solutions. It ideally acts as the roadmap for contemporary situations in assessing the UX of the FOSS community. Nevertheless, it offers contributions to the body of knowledge (O'Leary, 2017; Okoli, 2021), and on several occasions, scholars picture literature review as a dinner party (Kamler & Thomson, 2006) to avoid including everything encountered.

This study used The Systematic Literature Review (SLR) to execute and present most of the literature. It adapted procedures proposed by Kitchenham and Charters (2007), Budgen and Brereton (2006), and Kitchenham et al.(2010) to lay down both the theoretical and empirical foundation for accomplishing this research work. The body of knowledge suggests that SLR procedures are ideal for offering systematic and unbiased ways of extracting data from primary studies through distinctive techniques (Ali, Hongqi, Khan, Zhongguo, & Liping, 2017; Budgen & Brereton, 2006; Kitchenham & Charters, 2007; Kitchenham et al., 2010). **Figure 2.1** communicates the algorithm adopted by this study for performing this SLR.

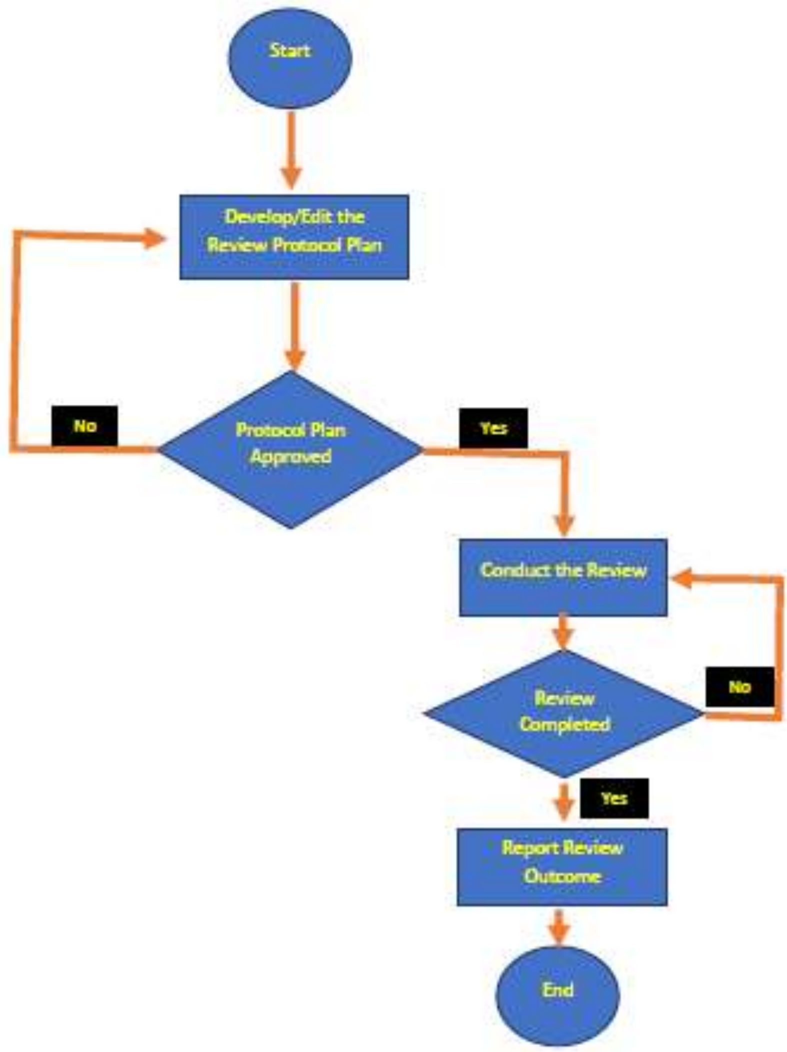

Figure 2.1: The Adopted Algorithm for Accomplishing the SLR
SOURCE: (Namayala et al., 2022)

The author has divided this chapter into five major sections: the second section explains the theoretical literature review and the third section provides the empirical literature review. The fourth section is allocated to the conceptual framework, and related works and research gaps are expressed in section five. Finally, the chapter concludes with a summary to make it easier to read and understand.

2.2 Theoretical Literature Review

Theories are fundamental blocks that build compact bases for studies; without them, studies may lose direction (Imenda, 2014). They are frequently used to direct research (De Vos, Strydom, Fouché, and Delport, 2005). Although there have been efforts to explain the necessity of having theories to guide UX design from Luojus (2010) and Kuutti (2010), the UX community does not have exclusive theories (Obrist et al., 2012; Obrist, Roto, & Väänänen-Vainio-Mattila, 2009). As a result, best practices suggest borrowing theories from cognitive sciences and psychology (Boichuck, 2020; Ishan, 2020) when implementing UCD and the design for UX.

This study has adopted three theories originating in psychology: the original version of the Technology Acceptance Model (TAM), schemas, and gestalt. Adopting these theories makes the study comply with Boichuck's (2020) and Ishan's (2020) recommendations.

2.2.1 The Technological Acceptance Model (TAM)

Davis created TAM in 1986 to anticipate and analyze users' willingness to accept technology and allow stakeholders to identify the hurdles and enablers to adopting new technology. According to Marangunić and Granić (2015), TAM has evolved into a critical paradigm for understanding determinants of human behavior toward future technology acceptance or rejection. TAM is a human behavior theory designed to predict and explain technology-related behavior (Venkatesh & Bala, 2008) and originates from the psychological theory of reasoned action and the theory of planned behavior. TAM proposes that system use is a reaction to users' emotional responses, defined by a system's perceived usefulness (PU), perceived ease of use (PEOU), and attitude toward the system's unique qualities and capabilities, which is an actual usage (AU) (see **Figure 2.2**).

Concerning optimizing the UX design in the FOSS community that consequently results in developing the FOSS-UXMM, TAM may provide a theoretical framework and insights into the factors that influence the acceptance and adoption of FOSS-UXMM. The TAM can guide the study in understanding the user perception (PU and PEOU), identifying key constructs, data collection and analysis, adopting factors influencing UX maturity, model development and validation, and predicting UX adoptions and impacts.

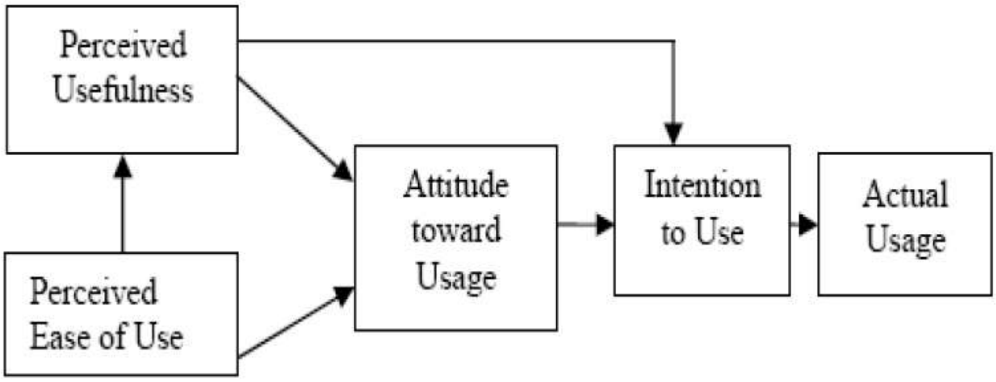

Figure 2.2: Technology Acceptance Model
SOURCE: Davis et al. (1989)

However, few studies in UX have adopted TAM, including Fedorko et al.(2018). One main contributing factor is that TAM was not initially developed for UX maturity studies but for general technology acceptance research. The overlap between TAM work and UX model work is limited: Most TAM work does not cite UX models, while most UX work comprises only one or two pages on TAM (Hornbæk & Hertzum, 2017).

2.2.2 Schema Theory

In 1923, Jean Piaget proposed a schema theory that showed how children learn and group things (Fernberger, 1927). In 1932, Bartlett advanced the theory (Bartlett & Burt, 1933) by specifying that individuals use their mental models (schemata) to perceive confronted information. Moreover, Rumelhart (1984) further advanced the theory by suggesting that schema is the framework that represents concepts and categorizes systems. Pankin (2013) provided its definition for a better understanding, saying schema theory is "*a branch of cognitive science concerned with how the brain structures knowledge.*"

Unsurprisingly, users have mental models (schemata) regarding software products, which helps them possess beautiful ideas of what they guess to encounter in products. They generally consider mental models as guidebooks or maps to aid in understanding concepts, including UX. Moreover, several researchers agree that optimizing UX requires matching actual experiences and personal schemata. Schemata may affect users' memory and influence UX in several ways, such as guiding attention, involving specific stimuli, providing prior knowledge to solve current problems, and integrating the mentioned constructs (Alba & Hasher, 1983).

UX design requires high creativity and is often associated with prior knowledge (Uysal, 2012). UX designers can use schema theory to predict what users already know and expect (Pankin, 2013). Schema theory can also assist researchers in gaining insights into the cognitive structures, beliefs, and perceptions that shape the UX maturity of the FOSS community. As a result, it can inform strategies and interventions to promote and enhance UX practices, processes, and awareness within FOSS projects.

2.2.3 Gestalt Theory

As an introduction to Gestalt theory, several researchers argue that human minds are good at filling in blanks. Seeing faces in tree leaves or cracks is possible, and this explanation is the underlying idea behind the development of Gestalt theory. For example, **Figure 2.3** shows the mango tree with the face of the late Mwalimu Julius Kambarage Nyerere, the first president of The United Republic of Tanzania.

Figure 2.3: Unusual Human Face that Depicts Gestalt Theory

The Gestalt theory believes that looking at visual elements as a whole provides more sense to users than looking at individual components (Jäkel, Singh, Wichmann, & Herzog, 2016). It shows how humans perceive things, for example, by grouping similar elements, recognizing patterns, and simplifying complex representations (Todorovic, 2008).

Therefore, designers, including UX designers, must organize the contents and interfaces to ensure they are aesthetically pleasing, meet functionality, and are easily understood (Bufe, 2021; Chapman, 2020; Todorovic, 2008), which is ideally supported by Gestalt principles. Moreover, the Gestalt theory often helps identify and select the most relevant design constructs that meet users' needs and influence their behaviors toward products (Bufe, 2021), a critical UX design issue.

The Gestalt theory has a German origin, which implies configurations and originated from contributions made by Max Wertheimer (Mitchell, 2006) in response to Wilhelm Wundt's structuralism (Wagemans et al., 2012). However, several other researchers, such as Ernst Mach, Immanuel Kant, and Johann Wolfgang von Goethe, have influenced its development (Bonacchi & Czerniak, 2015). The theory has several principles (Wagemans et al., 2012), shown in **Figure 2.4.** These principles have created several laws, including proximity, similarity, symmetry, and closure.

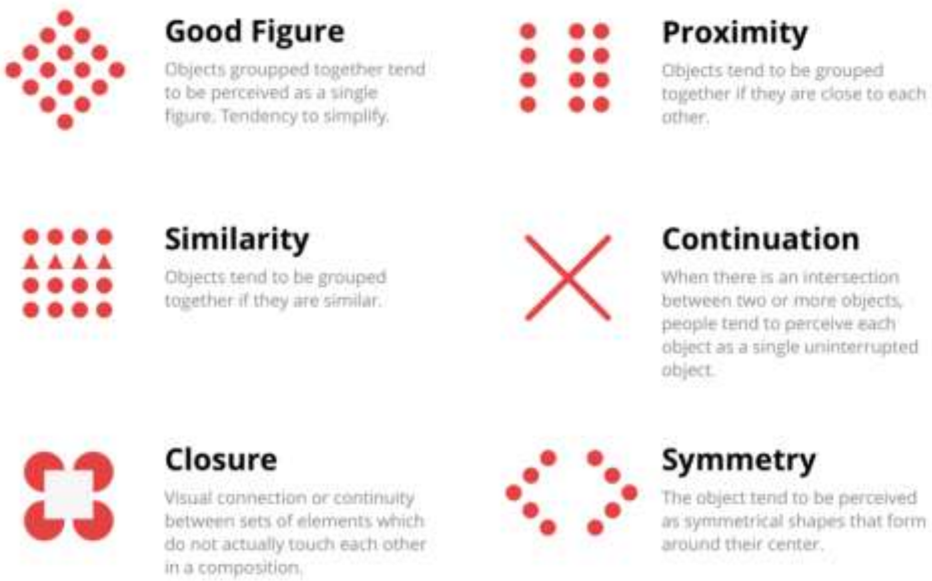

Figure 2.4: Gestalt Principles

SOURCE: Capan (2020)

Through its principles, the Gestalt theory provides meaningful contributions toward visual design for optimizing the design for UX. It extensively helps create more eye-catching designs (Bufe, 2021). However, the Gestalt principle resides in minds, not eyes, implying users must interpret what they see and find what they want. This idea aims to avoid confusing users from getting what they wish while guiding them to their selections.

2.3 Empirical Literature Review

This thesis section critically examines the empirical literature review on improving UX in desktop-based FOSS projects. As Nakano and Muniz Jr. (2018) suggested, the empirical literature review often provides a technological timeline for particular issues under discussion. This section contains UX Evaluations in FOSS Projects: Timeline, FOSS Linked Concepts and Licensing Issues, Quality Management in FOSS Projects, Quality Management in FOSS Projects, and The Capability Maturity Models for Software Processes. It also contains The Software Process Capability/Maturity Models in FOSS Community, The UX Assurance in the FOSS Community, UX Design Issues in FOSS Community, UX Capability Maturity Models and Imposed Challenges, and Usability, UX Metrics, and Measurement Methodologies Timeline.

2.3.1 UX Evaluations in FOSS Projects: Timeline

The reputation of FOSS projects, particularly server-based applications, is exponentially increasing (Bahamdain, 2015; Kamau & Namuye, 2012; Raza et al., 2012a; Steinmacher et al., 2019). As a result, governments, startups, and individual users are highly adopting server-based applications as an alternative to expensive proprietary software (Abramova et al., 2016; Bahamdain, 2015; Bouras, Kokkinos, & Tseliou, 2013; Bwalya et al., 2019; Petrinja & Succi, 2012). Some governments, such as South Africa, have even gone the extra mile and formed policies that direct their organs to adopt FOSS projects as production software (Bwalya et al., 2019). These policies are divided into three categories: mandatory (where the use of FOSS is mandatory), preferential (where the use of FOSS is favored), and advisory (where the use of FOSS is tolerable). Although Tanzania has had a national ICT policy since

2003, reviewed twice in 2016 and 2023, the policy has mainly been advisory regarding adopting FOSS applications.

According to Laszlo (2009), governments and agencies are the most significant computer software users and may significantly influence the FOSS software industry. However, desktop-based FOSS project adoption in many governments, particularly in African states, is unfavorable (Cenatic, 2010; Kamau & Namuye, 2012; Oreku & Mtenzi, 2013). Nonetheless, desktop-based FOSS projects are not well-performing in the private sector; several proprietary counterparts beat them in market share (Nagy et al., 2012; Namayala et al., 2022). Although additional factors contribute to the low acceptance of FOSS projects, the incapacity of ordinary people to interact with them competently is the most significant. Only people with technical backgrounds can use desktop-based FOSS projects smoothly (Lerner & Tirole, 2003; Nichols & Twidale, 2003; Raza et al., 2012a).

Researchers often link the inability to interact with desktop-based FOSS projects competently with undesirable UX. Therefore, poor UX has been a significant cause of poor adoption of several desktop-based FOSS projects (Cheng & Guo, 2018; Dhir & Dhir, 2017; Masson et al., 2017; Rajanen & Iivari, 2015a; Taibi, 2015). Since the cognitive psychologist and designer Don Norman coined UX in 1995, UX has not been the priority in the FOSS community's traditional practices. As a result, the FOSS community poorly plans UX-related tasks and inadequately allocates and dedicates UX resources.

The FOSS community neither expected to sell nor release its projects to a large audience (Duffy, 2018). Engineers developed FOSS projects for engineers, particularly in the academic and research communities. Many early contributors to the community were engineers or computer scientists, and they created the software primarily for their use and that of their colleagues. However, as the use of FOSS grew and became more widespread, it is now adopted by people with a wide range of backgrounds and skill levels (Llerena et al., 2018). The recent pattern witnesses the adoption of FOSS projects into critically-based missions such as missile launching and health systems where usage precision is critical (Bygstad, Ghinea, & Brevik, 2008; Grindrod, Li, & Gates, 2014; Hornbaek, 2010; Lizano, Sandoval, Bruun, & Stage, 2013; Marcilly et al., 2015).

Despite the growing recognition and importance of UX/HCI practices in the FOSS community for creating FOSS projects that are accessible, usable, and appealing to a wide range of users, it has been challenging to adopt their practices and techniques (Bornoe & Stage, 2014; Terry et al., 2010). The study acknowledges the presence of a few FOSS projects that have embraced UX/HCI practices, for example, Mozilla Firefox, the GNOME desktop environment, and the LibreOffice project. However, they have not extensively changed users' mindset from using proprietary counterparts such as Microsoft Office. Nevertheless, embracing UX/HCI practices have been ad hoc (Kurosu, 2017; Ovad & Larsen, 2015), often applied in isolation (Abrahão, Juristo, Law, & Stage, 2010; Ferreira, Sharp, & Robinson, 2012; Ovad & Larsen, 2015), and generically developed (Raza et al., 2012a).

Many works in the literature still support the need to explore and redesign existing HCI techniques to match the community's dynamics (Namayala et al., 2022; Terry et al., 2010). They ascertain that it might be more challenging to coordinate design efforts and ensure consistency across the FOSS projects with current UX/HCI practices. New methods that demand more advanced techniques are still needed (Kashfi, Nilsson, & Feldt, 2017). These techniques may make UX and HCI designers more open to feedback and collaboration from community members and more flexible in adapting their designs to meet the needs and preferences of a diverse user base.

From the review, this study has identified several challenges related to assessing and improving UX in the FOSS community that need immediate research attention. These challenges are threefold. First, experimentation is necessary to exhaust an understanding of users' UX needs. Second, the FOSS community must adapt HCI approaches to match its UX assessment dynamics. Third, the community must propose novel empirical methods for evaluating its UX maturity.

2.3.2 FOSS Linked Concepts and Licensing Issues

The Open Source Initiative was founded in 1986 as a steward of the FOSS community and is a not-for-profit organization. The body of knowledge formulated this entity to provide education and demonstrate the FOSS's embedded benefits. It further lays the foundations regarding smooth applications of FOSS projects and their compliance with legal requirements because it is the license approval.

FOSS projects require no prior agreement with the author and are free to use and modify. However, the right to distribute FOSS projects requires users to accept terms and conditions that serve as agreements. These terms and conditions stipulate requirements for exploiting FOSS projects as commercial applications. According to the Open Source Initiative (2010), the distribution terms of the FOSS projects must satisfy the criteria shown in **Table 2.1**

Table 2.1: FOSS Distribution Terms

Sn.	Criteria	Description
1	Free Redistribution	The license shall not demand royalties or additional fees; therefore, it does not restrict selling or giving away the software or its components.
2	Source Code	Source codes must include the rights to distribute and compile them. Unambiguous guidelines must be provided without additional expenditures when they are not available. Finally, international obfuscation of Source code, as well as intermediate forms, is not permitted.
3	Derived Works	Distribution of licenses must permit modifications and the creation of derivative works distributed under the same terms as the original software license.
4	The integrity of the Author's Source Code	The license must explicitly allow the sharing of the works derived and provide the potential to limit the circulation of these modified source

Sn.	Criteria	Description
		codes. On the other hand, Derived works may have alternative names or version numbers.
5	No Discrimination Against Persons or Groups	Individuals or groups shall not be discriminated against in the distribution license.
6	No Discrimination Against Fields of Endeavor	Anybody must be able to use the program under the license terms. These terms might not prohibit the program from being utilized in education or other broad domains like research.
7	Distribution of Licenses	The software's rights must apply to all users, even those in redistribution.
8	The license Must not be Specific to a Product.	The program's rights are unrelated to the program's inclusion/exclusion in a specific product's distribution agreement. As a result, all redistributing program parties should have the same rights as those granted in the original license when the program is a derivative of the software released, utilized, or distributed with the ancestor's permission.
9	The license must not restrict Other Software.	The license must not restrict other software delivered alongside the licensed software.
10	The License Must be Technology-neutral	The provisional license forecasts do not apply to any specific interface design technology.

SOURCE Open Source Initiative (2010)

Because all internet content, including FOSS, is copyrighted, two major categories of FOSS licenses named copyleft or "Share-Alike" and permissive or "Anything Goes" exist (Open Source Initiative, 2010). Permissive licenses allow users to copy, alter, combine, and redistribute the work with few restrictions. It usually requires attribution on several occasions. The most differentiator differentiating it from a copyleft license is allowing proprietary derivatives and expecting almost nothing in return regarding future commitments. While a copyleft grants the same rights as a permissive license, it mandates that any derivative work created is released under the same licensing terms. The copyleft license is based on the principle that anything made should be free if the contents used are free. Its objective is to stop commercial businesses from benefitting from open-source content without contributing to the community.

As the standard routine in the FOSS community, all licenses must be approved by the Open Source Initiative before becoming effective (Open Source Initiative, 2010). However, no perfect or poor license exists, and practitioners can develop what suits their needs. In this regard, over two hundred FOSS licenses have been approved and exist. Practitioners often group them as popular, special, and superseded or retired licenses.

This study only discusses limited examples of the commonly used FOSS licenses. However, the Open Source Initiative (2010) confirms the availability of over eighty FOSS licenses in this category. **Table 2.2** provides examples of copyleft licenses, and **Table 2.3** lists permissive licenses.

Table 2.2: Examples of Copyleft Licenses

Sn	License	Description
1	General Public License (GPL) or (GNU GPL)	GPL is the firm copyleft license and the most widely used open-source license (Lahti & Peterson, 2007; Negus, 2015). It is a specific application of Richard Stallman's "copyleft" philosophy, which he developed to safeguard GNU software from becoming proprietary. The FSF is now a custodian of this license, and on 29th June 2007, the GPL license was modified to GPLv3 to add more clarity and broader application (Negus, 2015).
2	Mozilla Public License (MPL)	The current version of MPL is MPLv2. The Mozilla Foundation currently hosts and handles maintenance activities related to this license. The movement to create the MPL license dates back to 1998 when Netscape Communications wanted to release binary and source code (Laurent, 2004). This license has a similar history to the Netscape Public License (NPL). This license is a hybrid of BSD and GPL licenses.
3	The Creative Commons "Share-Alike" license (CC-BY-SA)	There are several Common Creative (CC) licenses out there. However, they share some characteristics, and the Common Creative "Share Alike" (CC-BY-SA) is categorized as a copyleft. It allows other developers to remix, modify, and expand upon existing work, even for profit, as long as they acknowledge and license their new products under the same terms.
4	The Common Development and Distribution License (CDDL)	It was produced by Sun Microsystems while imitating the principles of MPL and Eclipse Public License (EPL). The literature considers it a weak copyleft license. According to the Open Source Initiative (2010), it has a strong community behind it and may enable the distribution of compiled object code under

Sn	License	Description
		any license, including the source code. Still, derivatives must be made available under the CDDL.
5	The Eclipse Public License (EPL)	The Eclipse Foundation's open-source license is most notably used for the Eclipse IDE and other projects. It is a weak copyleft license with two versions: the original EPL-1.0, published in 2004, and the EPL-2.0, issued in 2017. Like many other weak copyleft licenses, the EPL imposes more restrictions on using licensed code than permissive licenses while providing more freedom than solid ones.

Table 2.3: Examples of Permissive Licenses

Sn	License	Description
1	MIT License	It belongs to the permissive family of free software licenses, and its development started in the late 1980s at the Massachusetts Institute of Technology. It is much simpler and easier to understand than many other licenses, and it has high license compatibility since it only places minor restrictions on reuse. It also does not compel anyone who changes the original code to share their changes under the same license. Even if derivatives extensively change the code, there is no compulsion for reciprocity or "pay it forward." The updated version can still be kept private.
2	Berkeley Source Distribution (BSD) Licenses	These licenses have a low restriction type that does not impose redistribution obligations. UC Berkeley created its original version, the 4-Clause BSD license, in 1980; however, it is no longer widely used. The BSD 3-Clause and BSD 2-Clause are the most commonly encountered versions, and the most widely used license is the BSD 3-Clause. They support the distribution of various freeware, shareware, and open-source software.
3	Apache License 2.0	The Apache License, written by the Apache Software Foundation, is a liberal free software license (ASF). It permits users to use the software for any purpose, distribute it, modify it, and distribute modified copies under the license terms without worrying about royalties.
4	CC-BY 4.0	This license allows sharing: users can copy and redistribute the material in any media or format, and adaptation: users can remix, change, and build on the material for any purpose, including commercial ones. These liberties are under two terms. First, users must

Sn	License	Description
		offer proper credit, including a link to the license, and identify whether they changed. Second, there are no further limitations: for example, users may not impose legal requirements or technological barriers that prevent others from performing what the license allows. Moreover, the license cannot cancel granted liberties as long as users abide by the terms.

The review in this part has revealed several concerns that demand immediate action regarding FOSS-linked concepts and licensing issues. Although the community has established mandatory FOSS distribution criteria for FOSS products and derivatives and developed various open-source licenses, some ambiguities still necessitate more investigations. For example, there are a plethora of open-source licenses to choose from, and the sheer quantity of them is frightening since it causes users even more uncertainty when comprehending their tiny distinctions and making the best decision. Selecting an appropriate license with this immersed number of licenses is challenging for several practitioners, specifically when they are not versed in law and have never dwelled in FOSS licenses.

Moreover, licensing compliance among community members is an issue, especially in large software packages that contain proprietary and FOSS components (Fendt & Jaeger, 2019; Harutyunyan, Bauer, & Riehle, 2019; Riehle & Harutyunyan, 2019).

Although several initiatives exist to create standards for FOSS licensing and automate checking their compliance, such as The Software Package Data Exchange (SPDX) by Stewart et al. (2010), they still have several limitations that require additional efforts. For example, the interplay between FOSS licenses is complicated (German & Hassan, 2009; van der Burg et al., 2014), and it is easy to get mixed up regarding open-source licenses. Some licenses are much lengthier and complex to recall, making them difficult to follow. Worse, some go against the families' values to which they belong. Moreover, keeping track of current open-source licenses and embedded requirements is challenging for seasoned developers and compliance professionals, making some users wary.

As a result, the FOSS community should research how to combine some of the licenses to reduce their numbers substantially, undertake a comparison analysis to emphasize their observed differences, and ultimately simplify them by using intelligible terminologies to make them shorter.

2.3.3 Quality Management in FOSS Projects

Managing the quality of FOSS projects is critically important, as it has been in other software projects. For decades, the FOSS community's fundamental goal has been reaching a point where they can consistently deliver high-quality products. However, this mission is far from reality; managing project quality has been challenging for years. FOSS projects' quality varies widely depending on factors such as the size of the development community, the level of funding and resources available, and the level of expertise of the developers involved. Every FOSS project is developed uniquely (Haddad, Warner, The Linux Foundation, & Haddad, Ibrahim, 2011; Sarrab & Rehman, 2014), and existing development methodologies are not used consistently across projects.

Moreover, current quality measurement initiatives may not be the perfect match for the FOSS community. There is ongoing debate and discussion within the FOSS community about the most effective and appropriate quality-measuring initiatives. Likely, the current initiatives cannot measure unique aspects of the community, prompting the need to develop the community's specific tools (Adewumi, Misra, & Omoregbe, 2013). However, projects in the FOSS community are assumed to have higher quality than many of the proprietary software (Adewumi et al., 2013; Gwebu & Wang, 2011; Hedberg, Iivari, Rajanen, & Harjumaa, 2007; Kuwata et al., 2014; Laplante, Gold, & Costello, 2007; Mutula & Kalaote, 2010; Nagy et al., 2012; Petrinja & Succi, 2012; Raja & Barry, 2005; Raymond, 2005; Senyard & Michlmayr, 2004; Umm-E-Laila, Zahoor, Mehboob, & Natha, 2017; Zaffar et al., 2011). This assumption results from developers reviewing each other's code to identify and address potential issues or bugs, using automated and manual testing, and establishing standards and guidelines for code quality and best practices.

Many studies and quality standardization efforts exist, such as the ISO 9000 family, that identify customer focus, leadership, and employee involvement as quality management principles that a good software quality management initiative must adhere to. However, defining and measuring software quality is still difficult and complicated (Lysne, 2018; Sung, Kim, & Rhew, 2007). Moreover, there is no universally agreed-upon definition of quality (Agu & Bakpo, 2011), making management of quality ambiguous because it means different things to different people in various roles, situations, and times (Kobyliński, 2013; Safder & Yousaf, 2018; Soomro & Ahmad, 2012).

The software development process is a complex endeavor, which encourages the adoption of SDLC methodologies, such as waterfall, V-shaped, prototype, spiral, iterative incremental, and agile, to deliver high-quality software that complies with users' requirements. SDLC thoroughly defines each distinctive stage of development and provides a better understanding of processes (Rastogi, 2015). However, it only prioritizes functionalities rather than human requirements, such as physical and cognitive abilities, emotional needs, personality qualities, and situational considerations (Majid, Noor, Adnan, & Mansor, 2010; Zhang, Carey, Te'eni, & Tremaine, 2005). It only applies in proprietary software development environments (Mandal, Kandar, & Ray, 2011; Pressman & Maxim, 2014).

Several FOSS SDLCs are discussed in the literature, such as the "release early, release often" (RERO) model by Raymond (2005) and the "fork and merge" model. However, there is no one-size-fits-all development life cycle model for FOSS projects. FOSS lacks a defined and consistent SDLC; different SDLCs exist based on developers' prior expertise and experience (Saini & Kaur, 2014). **Figure 2.5** displays a schematic illustration of Roets et al.'s (2007) FOSS development model. The FOSS SDLC by Roets et al. (2007) incorporated the stages of the conventional SDLC, built on Jørgensen's (2001) life cycle model. and contained additional elements from earlier models, especially Schweik and Semenov's (2003).

However, according to Haddad et al.(2011), the typical FOSS SDLC is iterative, beginning with a new project or feature idea followed by design and, finally, implementation of running software that the development team delivers promptly to users, regardless of quality. The community then tests the software and provides feedback, bug reports, and required fixes via the mailing list and discussion forums. Project stakeholders and maintainers use the provided to improve implementation and release new developments. This process is repeated as often as necessary until the project members believe the commission is robust enough to release as the stable version. Following the release of the regular version, the cycle repeats to obtain and release multiple newer stable versions. FOSS SDLC often mimics the recent software development models, such as Agile, and its derivatives, such as Extreme programming (Saini & Kaur, 2014).

Neither traditional SDLC for proprietary software nor FOSS SDLC adequately accounts for how HCI/UX issues are addressed. Very few studies on SDLC and FOSS SDLC pay attention to UX/HCI issues (Boivie, Åborg, Persson, & Löfberg, 2003; Zhang & Li, 2004), and neither HCI nor other design for UX principles has secured a firm foothold in both lifecycles (Anitha & Prabhu, 2012; Ferre, Juriste, & Moreno, 2005; Velmourougan, Dhavachelvan, Baskaran, & Ravikumar, 2014; Zhang et al., 2005).

Unlike the typical software engineering cascade models, the FOSS development models offer unique qualities that make them better paradigms for developing software (Haddad et al., 2011). They promote a more fluid development process with enhanced intra-team communication, continuous integration and testing, and higher end-user participation. Further, they follow the ground-up development, support early and frequent releases and peer reviews, and accommodate minor and incremental modifications. Other qualities include continuous quality improvement, project testing, and flagging features that neglect security concerns and may involve end-users throughout the process.

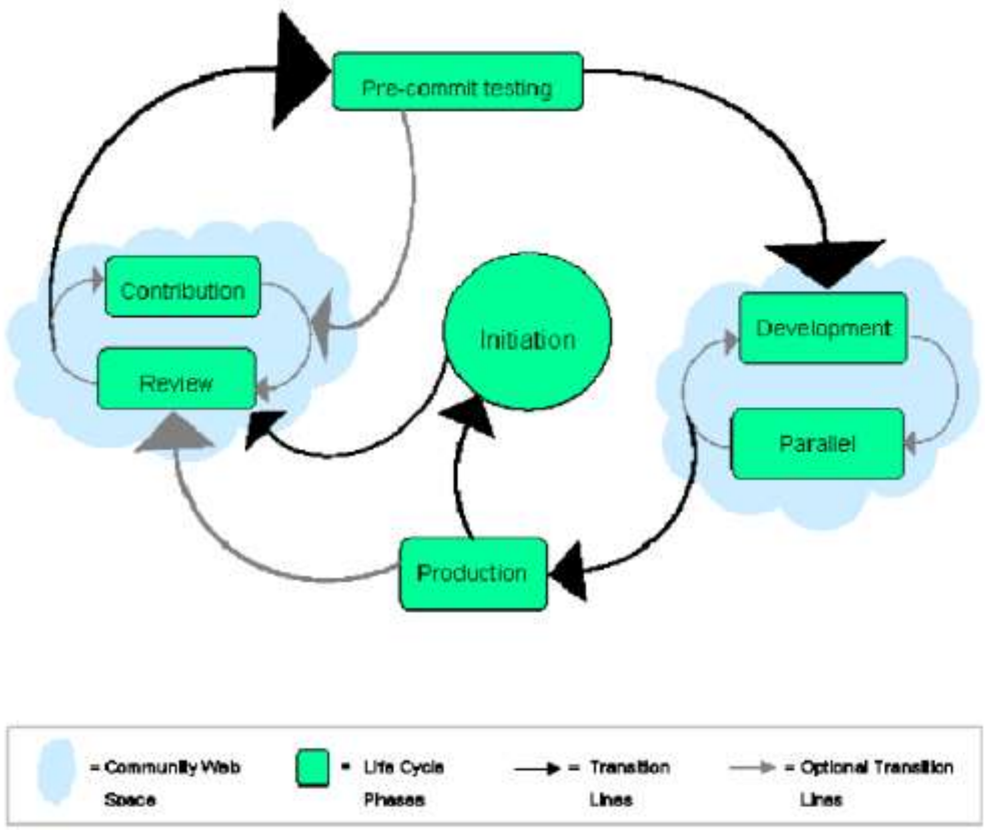

Figure 2.5: FOSS Development Life-Cycles

SOURCE: Roets et al.(Roets et al., 2007)

Taking a different turn, several organizations and users have had misconceptions about the quality of FOSS projects, which have caused them to lose faith (Laplante et al., 2007). Practices confirm that locating, assessing, and selecting the best viable FOSS product or FOSS components for a given need is challenging (Umm-E-Laila et al., 2017), necessitating alternative approaches. Many studies, including Raza et al. (2012a) and Sung et al.(2007), have indicated the necessity of developing quality assessment and selecting models exclusively for the community. Nevertheless, Adewumi et al. (2016) suggested that future FOSS quality models must account for MCDM. The MCDM can more effectively handle ambiguity, subjectivity, human biases, and uncertainties (Ali et al., 2018).

Present knowledge confirms that practitioners have created numerous FOSS quality models to aid in quality assessment and selecting FOSS projects or components to integrate into other projects; however, they are not commonly used (Li et al., 2009; Petrinja & Succi, 2012; Stol & Ali Babar, 2010). The study of Adewumi et al. (2016)

provides more enlightenment on FOSS quality assessment models that frequently provide specific parameters to help select FOSS projects or components. Quality models are found in three categories: definition, assessment, and predictions (Ouhbi, Idri, Aleman, & Toval, 2014; Ouhbi, Idri, Ferńandez-Alemán, & Toval, 2015), and many are derivative of ISO 9126 (Adewumi et al., 2013; Miguel, Mauricio, & Rodríguez, 2014).

Moreover, the comparative study of Umm-E-Laila et al.(2017) lists examples of models responsible for ensuring the quality of selected FOSS projects' components. The list comprises the Methodology of Qualification and Selection of Open Source Software (QSOS) by Atos Origin (2008), A Quality Model for Open Source Software Selection by Sung et al.(2007), Open Business Readiness Rating (Open BRR) by OpenBRR.org (2005), and Quality Platform for Open Source Software (QualiPSo) by QualiPSo (2007). Others include Evaluating OSS through Prototyping by Carbon et al.(2007), Software Quality Observatory for Open Source Software (SQO-OSS) by Samoladas et al.(2008), Evaluation Framework for Free/Open source projecTs (EFFORT) by Aversano and Tortorella (2013), the Quality model for the selection of FLOSS-based issue tracking system by Raffoul et al.(2008), and Quality Measurement Model for Analysis and Design Tools Based on FLOSS by Alfonzo et al.(2008).

Despite the efficacy of current FOSS quality models in supporting the quality selection of FOSS projects and components to integrate into other applications, they do not address all essential dimensions of quality characteristics. Nevertheless, they do not consider previous research suggestions and mostly look at a single quality feature.

The author identified sixfold areas that may require immediate attention from researchers. First, it is necessary to empirically determine the crucial components (metrics or KPIs) of FOSS quality models and offer roadmaps for integrating them into existing models. Second, a thorough comparative analysis of current FOSS quality models is necessary to grasp better their key features, development methods, nature, similarities, and differences. Third, more research is required to understand further how to integrate UX procedures into FOSS SDLC. Fourth, addressing academics' erroneous beliefs about capability maturity models and FOSS quality evaluation methods is necessary. Exhausted review confirms that several studies have erroneously combined capability maturity and quality assessment models when reviewing these concepts. Some of these studies include Adewumi et al.(2016), Adewumi et al. (2013), and Tassone et al. (2018). Fifth, there are few tools and methods to predict the quality of projects in the FOSS community since the third category of quality assessment models, prediction, is uncommon in this community. The lack of tools and methodologies for predicting quality necessitates planning and development to improve quality successfully. Finally, it is necessary to use fuzzy mathematics techniques to represent new FOSS quality evaluation frameworks and ensure they accommodate MCDM.

2.3.4 The Capability Maturity Models Timelines

The Capability Maturity Model (CMM) was established in 1991 by the Software Engineering Institute (SEI) at Carnegie Mellon University (CMU). However, evidence suggests that the movement leading to its establishment started in August 1986. CMM is an endeavor that considers the thoughts of numerous stakeholders from different disciplines (Paulk, Weber, Curtis, & Chrissis, 1996; Paulk, Curtis, Chrissis, & Weber, 2011). CMM has retired and discontinued further revisions since 1997 in favor of Capability Maturity Models Integration (CMMI). However, it was helpful in its heyday in ensuring that software development procedures were formal and mature (Paulk et al., 1996; Paulk et al., 2011).

The ability to produce real organizational process improvements (Titov, Bubnov, Guseva, Lyalin, & Brikoshina, 2016) and reduce the required time and expenses (Aubakirov & Nikulchev, 2016) are the reasons why CMMs were so popular. SEI used a tiered scale to depict CMM and built the organization's maturity level from a single figure obtained from the performance of essential Process Areas (PAs) (Jokela et al., 2006). Although SEI planned to use CMM primarily for establishing new software projects and programs, other disciplines adopted its practices (Titov et al., 2016).

The original version of the CMM, also known as the software CMM version one (SW-CMM v1.0), was a derivative of Total Quality Management (TQM) by Lakhe and Mohanty (1994), Crosby's Quality Management Maturity Grid (QMMG) by Crosby (1980, 1995), and Humphrey and Sweet's knowledge (1987). See **Table 2.7** for details of available stages and accompanying descriptions of CMM, **Table 2.4** for more information on QMMG, and **Table 2.6** for additional details of TQM. The study has offered this information to equip leaders with a complete understanding of their structures, similarities, and differences.

Table 2.4: Maturity Levels of Crosby's Quality Management Maturity Grid

Stage	**Description (s)**
1: Uncertainty	We do not know why we have quality problems.
2: Awakening	Must always have quality problems
3: Enlightenment	Because of Management commitment and quality improvement programs, we are identifying and resolving our quality problems.
4: Wisdom	We routinely prevent defects from occurring.
5: Certainty	We know why we do not have quality problems.

Due to stakeholder and industry feedback, the CMM went through several versions, and each release improved the previous one. As a result of CMM's success stories, it inspired other professionals to create CMM specific to their domains (Titov et al., 2016). For example, the human resource management field developed the People Capability Maturity Model (P-CMM) (Curtis, Hefley, & Miller, 1995; Cutis, Hefley, William, & Miller, Sally, 2001), the project management field developed the Project Management Process Maturity (PM)2 Model (Kwak & Ibbs, 2002), and the strategic planning for project management created Project Management Maturity Model (Kerzner, 2001).

Nevertheless, other fields made multiple CMMs. Examples of these CMMs include the Portfolio, Programme, and Project Management Maturity Model (P3M3) (Office of Government Commerce - OGC, 2010), the PRINCE2 Maturity Model (P2MM) (Williams, 2010), and the Organizational Project Management Maturity Model (OPM3) (Project Management Institute., 2008).

There are several other CMMs for particular applications. For example, Information Quality Management (Baškarada, 2009), Digital Investigations (Kerrigan, 2013), Training and Education (Wagenstein, 2006), Cybersecurity (Curtis & Mehravari, 2015; Kour, Karim, & Thaduri, 2020; Miron & Muita, 2014; Mohammed & Bade, 2019), and Open Source Usability Maturity Model (OS-UMM) by Raza et al. (2012a). However, unlike CMM, many specialized and application-based CMMs have waned over time because they cover a fraction of the system's components and lack sufficient documentation (Titov et al., 2016). They also claimed to duplicate effort and reinvent the wheel.

CMM has also served as the benchmark for other models, including Bootstrap (Kuvaja & Bicego, 1994), a European evaluation technique, and ISO/IEC 15504, formally known as Software Process Improvement and Capability Determination (SPICE) (ISO/IEC, 2004).

Unlike CMM, CMMI assesses software, hardware, and services development in practically every industry (Margarido, Fari, Vieira, & Vidal, 2013; Paulk et al., 2011), and it roughly contains five stages, 18 process zones, 52 priorities, and over 300 primary activities (Hou et al., 2021). It benchmarks the software processes' capability (Hou et al., 2021) and covers two dimensions of representation in enhancing methods: staged and continuous (CMMI Institute, 2018; CMMI Product Team, 2010; Jokela et al., 2006; SEI, 2002). The staged representation shows a path for improvement and must address PAs. On the other hand, continuous representation allows businesses to choose which PAs they want to enhance and in what order.

Compared to CMM, CMMI offers businesses more latitude in approaching process improvement using these dual representations while still receiving its support. Today, there are multiple CMMI versions due to feedback and suggestions from stakeholders and the need to accommodate emerging industry changes (Henriquez, Moreno, Calvo-Manzano, & Feliu, 2021). These versions are CMMI v1.02 (SEI, 2000), CMMI v1.1 (SEI, 2002), CMMI v1.2 (SEI - Software Engineering Institute, 2006), CMMI v1.3 (CMMI Product Team, 2010), and CMMI v2.0 (CMMI Institute, 2018). The current version, CMMI v2.0, is a popular and extensively adopted paradigm among practitioners and stakeholders.

Among other reasons, the popularity of CMMI v2.0 results from its ability to enable various businesses to acquire competitive advantages in tackling various business-related difficulties such as managing suppliers, workforce, work, and quality to improve implementations and sustain the business. Nevertheless, CMMI v2.0 is compatible with other evaluation programs, such as agile methodologies (Henriquez et al., 2021).

Moreover, in December 2018, SEI expanded CMMI v2.0 to take care of services and suppliers' management. **Table *2.5*** provides more details regarding available stages, descriptions, and KPAs of CMMI v2.0.

Table 2.5: Levels of the CMMI v2.0

Levels	Description	Rated PAs
1: Initial	It does not fully improve processes; when it does, it is volatile.	None
2: Managed	Implementers reactively implement process improvement.	Performance and management, management of Supplier Agreement, Process Quality Assurance, Configuration Management, Monitoring and Control, Planning, Estimating, Requirements Development and Maintenance, Governance, and Implementation Infrastructure

Levels	Description	Rated PAs
3: Defined	The organizations use assets, standard practices, and other tailored issues to achieve desired objectives.	Managing Performance and Management, Supplier Agreement Management, Process Quality Assurance, Configuration Management, Monitoring and Control, Planning, Estimating, Requirements Development and Maintenance, Governance, Implementation Infrastructure, Causal Analysis and Resolution, Decision Analysis and Resolution, Organizational Training, Risk Management, Process Asset Development, Peer Reviews, Process Management, Verification and Validation, Technical Solution, and Product Integration
4: Quantitatively Managed	Processes statistically control and quantitatively measure objectives, desired quality, and performance.	Managing Performance and Management, Planning, Governance, Causal Analysis and Resolution, and Process Management
5: Optimizing	Optimizes practices for process improvements using quantitative and statistical techniques	Managing Performance and Management and Causal Analysis and Resolution

Table 2.6: Eight Core Elements of Total Quality Management

Element	Description
1: Customer-focused	Customers are the focal point for improving quality.
2:Total employee involvement	To achieve organizational goals adequately, all employees must work together toward common goals.
3: Process cantered	Process thinking is the core part of the TQM.
4: Integrated system	Interconnecting organizational horizontal and vertical functions is the core focus of TQM.
5: Strategic &systematic approach	TQM must include formulations of strategic plans integrating quality management
6: Continual improvement	TQM must facilitate continually improving processes to meet stakeholders' expectations and stay competitive.
7: Fact-based decision-making	TQM insists on continually collecting and analyzing the organizational data
8: Communications	Effective communication helps motivate employees and maintain their work-related morale.

Table 2.7: The levels of the Capability Maturity Model (CMM)

Levels	Description
1: Initial level	There is no organization of processes at this level, and success depends on the individual's knowledge and experiences regarding their implementations.
2: Repeatable level	Processes are likely to be documented and defined. There is also the exercising of basic techniques in project management.
3: Defined level	Developed Software development processes pay much attention to documentation, standards, and integration.
4: Managed level	Corporate data are collected and analyzed to monitor, evaluate, and control software processes.
5: Optimizing level	Organizations found better strategies for serving needs and constantly improving software processes.

Despite the widespread use of all versions of CMMI and alternative CMMs such as ISO/IEC 15504 (ISO/IEC, 2004), none have processes dedicated solely to evaluating UX (Lacerda & von Wangenheim, 2018; Raza et al., 2012a). They also use insufficient measurements, are overburdened with management challenges, and cannot address constantly changing surroundings (Yassien, 2020). Several studies, such as Garzás et al.(2013) and Sanders (2011), confirm that they are expensive and complex. Nevertheless, Ali et al.(2018) and Wiegers and Sturzenberger (2000) indicate that CMMI and other CMMs' limitations need alternative assessment techniques.

Nevertheless, the CMMI framework is relatively large and complicated for several projects, making the assessment more bureaucratic and likely to terminate users' willingness to learn (Sanders, 2011). It may also be insignificant to software processes where modern software improvement technologies may not immediately boost productivity and competitiveness. Finally, CMMI requires massive document preparations and training assessors before adoption, which is expensive and inaccessible for many firms (Sanders, 2011).

The author has identified many areas in CMM that need additional research. For example, supplementary studies are still invited to develop simplified, cost-effective CMMs that encourage user learning, contain the best metrics, and respond quickly to changing conditions and environments.

2.3.5 The Capability Maturity Models for Software Processes

Salviano and Figueiredo (2008) describe Software Processes Capability/Maturity Models (SPCMMs), also referred to as Software Process Improvements (SPIs), as frameworks that represent best practices for SDLC based on sound engineering, process-management principles, and process-attribute sets for capability and maturity design aspects. Generally, SPCMMs establish basic standards for evaluating strategies and ensure that the outcomes are objective, unbiased, repeatable, consistent, and representative of the evaluated processes (ISO/IEC, 2004; Von-Wangenheim, Hauck, Salviano, & Von-Wangenheim, 2010). Moreover, they provide many advantages that help improve organizational processes and costs (Namayala et al., 2022; Rout, O’Connor, & Dorling, 2015).

SPCMMs often compare and evaluate process improvements or assessments, assuming that higher process capability or organizational maturity means better performance. The lower process capability may suggest potential risk sources (ISO/IEC, 2004; Von Wangenheim et al., 2010). Therefore, if a business wants to be competitive, it must maintain a higher level of maturity (Okręglicka, Mynarzová, & Kaňa, 2015; Păunescu & Acatrinei, 2012). Software developers have had to worry about compliance with seemingly endless SPCMMs at one point or another (Von-Wangenheim et al., 2010). The ISO/IEC 15504 Software Process Improvement and Capability dEtermination (SPICE) standard by ISO/IEC (2004) and the Capability Maturity Model Integration (CMMI) by SEI (2000) are two examples of the SPCMMs.

This study has offered definitions of normative parts of standard SPCMMs as specified by ISO/IEC 15504 (ISO/IEC, 2004) to ensure a better understanding of their architectures: capability, maturity, process reference model, process assessment model, measurement framework, and assessment procedure are some of these elements (Lacerda & von Wangenheim, 2018). However, the process assessment and reference models have the same meaning in CMMI (CMMI Product Team, 2010). As a result, including examples for each is a little unclear.

The study defines process capability as achieving the desired goals (de Bruin et al., 2005; Grabis, Zdravkovic, & Stirna, 2018; Lacerda & von Wangenheim, 2018). On the other hand, a software development process can be capable if its outcomes are consistently predictable and characterize the processes' ability to achieve or contribute to current or predicted organizational goals. (de Bruin et al., 2005; ISO/IEC, 2004, 2019; Paulk et al., 2011). The process capabilities evaluation usually targets specific processes to examine current practices and realize improvement initiatives. In contrast, the maturity of the process ensures that procedures are followed consistently (de Bruin et al., 2005; ISO/IEC, 2004; Lacerda & von Wangenheim, 2018). Software practitioners commonly analyze organizational processes' capabilities throughout the SDLC to ensure that development processes consistently accomplish anticipated outputs. On the other hand, organizational maturity aids specific organizations in selecting viable improvement initiatives (Van Looy et al., 2011), grouping many processes, and creating a single rating identity.

Furthermore, organizational maturities are frequently measured using identified capabilities of each process, implying that process capabilities are inputs to organizational maturities (Galvis-Lista & Sánchez-Torres, 2013; Jokela et al., 2006; Lacerda & von Wangenheim, 2018; Yeh et al., 2017). As a result, understanding organizational maturities requires an in-depth knowledge of the capabilities of each process.

Process Reference Models (PRMs) are the foundations for process assessments (Pöppelbuß & Röglinger, 2011; SEI, 2010). Present literature categorizes processes into process areas (PAs), which are collections of connected practices that, when adopted, achieve essential goals for improving a specific site (Lacerda & von Wangenheim, 2018). Along with capability dimensions (measurement framework), process dimensions (PRM) lay the groundwork for one or more Process Assessment Modes (PAM) (ISO/IEC, 2004). PRM often provides details regarding the reference model's domain and scope, the process's purpose, desired outcomes, the objectives of process cycles, and their current links (ISO/IEC, 2004, 2010). Most results from ISO 15504 (ISO/IEC, 2004) are prerequisites for realizing reached processes capability and organizational maturity levels (Lacerda & von Wangenheim, 2018). Thus, understanding Maturity Levels (MLs) and Capability Levels (CLs) requires a regular comparison of PRM's activities or best practices to organizational processes (Lacerda & von Wangenheim, 2018; Pigosso, Rozenfeld, & McAloone, 2013).

Best practices in processes and organizational evaluations frequently use process attributes to identify and appreciate the capabilities of reached processes and corporate performances (Ivanyos, 2009). Process attributes always measure elements that assure process success and help the firm achieve its objectives (ISO/IEC, 2004). Moreover, MLs and CLs meticulously describe organizational performance and capabilities.

Based on the achievement of established process attributes connected with a defined grouping of process attributes, the measurement framework offers a foundation for rating the capability of processes and the organization's maturity (ISO/IEC, 2004; Lacerda & von Wangenheim, 2018). The measurement framework consists of three normative elements: capability levels (CLs) (incomplete, performed, managed, established, predictable, and optimizing) that show the capability of the specified process in ascending order, from failure to satisfy the process's objective to meet current and projected business goals, process attributes that define a particular aspect of capability, and rating scale that characterizes the extent of process attributes achievement (ISO/IEC, 2004).

Moreover, according to Lacerda and von Wangenheim (2018), A rating scale is "an ordinal scale to quantify the extent of the process achievement." The assessment community expresses achievements of process attributes using ordinal rating scales that assume four different values: N for Not achieved, P for Partially completed, L for Largely achieved, and F for Fully achieved. **Table *2.8*** highlights further details of rating values.

Table 2.8: Rating Scales

Rating	**Meaning**	**Level of achievement**
N	Not achieved	0% to <=15%
P	Partially achieved	>15% to <= 50%
L	Largely achieved	>50% to <= 85%
F	Fully achieved	>85% to 100 %
P-	Partially achieved -	>15% to <= 32.5%
P+	Partially achieved +	>32.5% to <= 50%
L-	Largely achieved -	>50% to <= 67.5%
L+	Largely achieved +	>67.5% to <= 85%

A staged or continuous scaling system is commonly used to indicate capability and maturity levels (ISO/IEC, 2004, 2010, 2019a; Lacerda & von Wangenheim, 2018). They have the same content, but their organization has varied improvement goals. For example, on the other hand, the staged representation uses MLs to measure process improvement, whereas continuous presentation uses CLs to assess process improvement (ISO/IEC, 2004; Lacerda & von Wangenheim, 2018). Several PAs make up each ML, and a set must meet a given CL to reach a particular ML (SEI, 2010). **Figure 2.6** provides examples of staged and continuous scale representations.

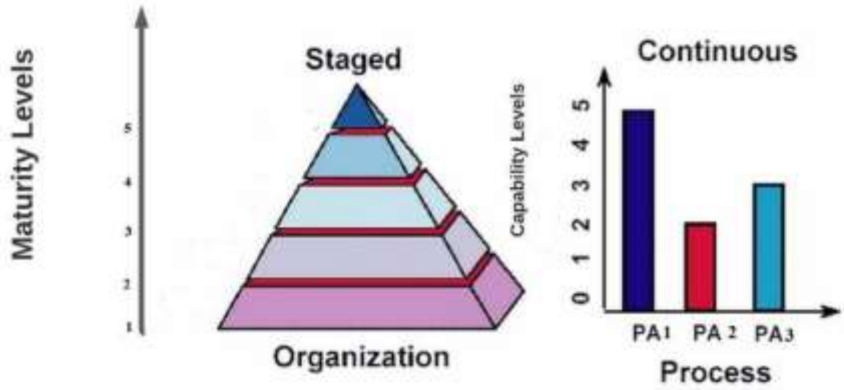

Figure 2.6: Samples of Continuous and Staged Scale

In addition, the Process Assessment Model (PAM) is always linked to PRMs and collects evidence and rates process capabilities (ISO/IEC, 2004). It must have a specification of its goal, scope, elements, mapping to the measurement framework and stated PRMs, and a system for consistently expressing results (Earthy, Bowler, Forster, & Taylor, 1999; ISO/IEC, 2004; Lacerda & von Wangenheim, 2018). Moreover, it includes two-dimensional views, including process entities that define the domain, scope, purpose, processes' results, and capacity scale.

PAM must meet several requirements articulated by ISO/IEC (2004). For example, it must declare its scope, including the selected PRMs, processes from PRMs, and the CLs set from the measurement framework. Moreover, it must be related to at least one function from those listed in PRMs. In addition, PAM must build its foundation based on sets of indicators (e.g., work products, resources, standard operating procedures, and the list continues.) that explicitly address aims and outcomes conformance with selected PRMs of processes scoped. Finally, for any function within its scope for a given process, PAM must handle all or a continuous subset of the levels (beginning at level 1) of the measuring framework for process capability.

The literature supports two approaches to designing SPCMMs: life cycle and potential performance (McBride, 2010; Wendler, 2012). Several previous SPCMMs result from life cycle viewpoints; however, most new SPCMMs are derivative of potential performance perspectives (McBride, 2010; Wendler, 2012). The community of software process improvement is still working on new versions of generic SPCMMs and altering others to upgrade (Von Wangenheim et al., 2010); however, it has never been a simple task (Magee & Thiele, 2004). Multiple efforts are underway to accomplish these tasks. However, most of the newly formed SPCMMs are derivatives of existing models (Monteiro & Maciel, 2020). They continually depict maturity and capability levels linearly and unidirectionally, shadowing staged growth techniques (Ali et al., 2018; de Bruin et al., 2005; Duane & O'Reilly, 2017). However, de Bruin et al. (2005) say that linearly and unidirectional stage representation of models makes it difficult to explain the maturities of complex domains and provides no guidance on how to remedy the problem. They further proposed an alternative presentation entitled "The stage-gate."

Despite available efforts, limited information is available for creating theoretically sound, rigorously proven, and widely approved SPCMMs (Becker et al., 2009; Matook & Indulska, 2009; Mettler, 2009; Monteiro & Maciel, 2020; Solli-Sæther & Gottschalk, 2010). However, several studies, such as Von-Wangenheim et al. (2010), recommend ISO standard processes to guide the customization process and ensure the desired level of modification. Others advise model authors to follow specific design procedure guidelines (Monteiro & Maciel, 2020), such as de Bruin et al. (2005), Becker et al.(2009), Otto et al.(2020), Van Steenbergen et al. (2010), Solli-Sæther and Gottschalk (2010), Lasrado et al. (2015), Mettler (2010) and Pöppelbuß and Röglinger (2011). However, the suggested design procedures and frameworks are insufficiently detailed to tell all the SPCMM design process stories (Otto et al., 2020; Pöppelbuß & Röglinger, 2011; Proença et al., 2013).

Most contemporary models for UX maturity result from the owners' years of knowledge and characteristics from projects that have produced positive results inside a company or industry (Matook & Indulska, 2009). They do not provide important information about their development methodologies (Von Wangenheim et al., 2010). Several SPCMMs are still confusing in practice, and further research is required since practitioners have examined only a few for validity, reliability, and generalization (de Bruin et al., 2005). Finally, contextualizing present SPCMMs to suit particular application contexts, such as the FOSS community, has been challenging for decades. Lastly, SPCMMs exclusive for assessing the FOSS community's UX maturity/capability are unavailable (Namayala et al., 2022; Raza et al., 2012a). As a result, initiatives to create them must be made.

2.3.6 The Software Process Capability/Maturity Models in FOSS Community

Compared to traditional software development approaches, assessing the FOSS community's UX capability and organizational maturity may necessitate different techniques. This requirement results from the FOSS community captivating a different approach to development cycles than traditional software development communities (Glassman & Kang, 2016; Kuwata et al., 2014). For example, it uses entirely different techniques when maintaining, testing, and developing FOSS projects (Glassman & Kang, 2016). Members of the FOSS community come from diverse backgrounds (Bach & Carroll, 2009), are distributed worldwide, and mainly adopt asynchronous communication channels (Baldwin & Clark, 2006).

Kuwata et al.(2014) confirm the absence of SPCMMs for assessing projects in the FOSS community and recommend developing new ones. Moreover, Petrinja and Succi's (2012) study argued that generically created SPCMMs might not adequately solve FOSS-specific difficulties, such as a lack of a dedicated budget, a geographically dispersed development team, and unknown end-users, and the presence of volunteered developers. However, from the developers' standpoint, FOSS development approaches are more practical and convenient for producing software (Glassman & Kang, 2016; Krishnamurthy et al., 2013; Pinto et al., 2018). They enable prompt identification and correction of flaws, boost productivity, motivate stakeholders, and deliver excellent results (Khanjani & Sulaiman, 2011; Kuwata et al., 2014).

Although the FOSS community lacks enough SPCMMs, the concept of SPCMMs is not alien to this community (Namayala et al., 2022; Raza et al., 2012a). Several researchers have developed SPCMMs for evaluating the quality of FOSS or components to integrate into other applications (Butler et al., 2022) that have evolved in reaction to changes in software development methods (Fendt & Jaeger, 2019; Harutyunyan et al., 2019).

Movements to create quality assessment models started in 2003 (Haaland, Groven, Regnesentral, Glott, & Tannenberg, 2010; Stol & Ali Babar, 2010). However, they have not been widely adopted (Butler et al., 2022; Li et al., 2009; Petrinja & Succi, 2012) and are less formalized, making practitioners frequently rely on their knowledge or referrals from their colleagues (Ayala, Hauge, Conradi, Franch, & Li, 2011; Franch et al., 2015; Lenarduzzi, Tosi, Lavazza, & Morasca, 2019; López et al., 2016). For example, choosing an acceptable model from a broad list has been a significant adoption barrier for current models (Maki-Asiala & Matinlassi, 2006). A lack of practical relevance is another issue that contributes to users rejecting them (Stol & Ali Babar, 2010).

As the path toward removing some barriers to adopting quality models in the FOSS community, Stol and Ali Babar (2010) and Umm-E-Laila et al. (2017) developed comparative frameworks that may assist practitioners in intuitively selecting FOSS quality assessment methods. However, little evidence exists regarding testing proposed frameworks in natural contexts.

An empirical literature review on this section has revealed fourfold areas that demand additional research efforts from the FOSS community. First, future academics must figure out how to create powerful SPCMMs specifically for evaluating the UX of this community. Second, to validate existing and generically developed SPCMMs, the community must use these models extensively in natural settings. Third, it strongly advises research to enhance the use of SPCMMs to assess the quality of FOSS projects and select the appropriate components. Finally, the frameworks used need further tests to choose an element to incorporate into other projects or evaluate the quality of FOSS projects.

2.3.7 The UX Assurance in the FOSS Community

Lakhani (2017) expresses UX assurance (UXA) as a practice primarily dealing with UX quality by accounting for usability, design affordances, findability, information clarity, interactions, and positioning of interface elements. In layperson's terms, UXA combines technical quality tests to ensure that the product under consideration is bug-free and meets technical requirements and specifications.

The FOSS community's history reveals that developers in the FOSS community have been creating FOSS projects for other developers (Hedberg et al., 2007; Namayala et al., 2022; Nichols & Twidale, 2003; Raza et al., 2012a). As a result, guaranteeing projects’ quality and desirable UX was not a core function of the FOSS project developers because users could fix projects’ flaws by themselves. However, in the current situation, where products in the FOSS community reach a broad user audience of naive and non-computer professional consumers, ensuring desirable quality and UX has gained prominence (Bahamdain, 2015; Feller & Fitzgerald, 2000; Hedberg et al., 2007; Nichols & Twidale, 2006; Raza et al., 2012a; Zhao & Deek, 2005).

As a step toward quality assurance, the FOSS developers typically rely on a broad user base and the availability of volunteer co-developers to report bugs (Hedberg et al., 2007). However, this poses a problem for software implementations that require quality assurance checks on products before delivery to end-users, mainly in the early phases of development (Bahamdain, 2015; Hedberg et al., 2007; SenthilMurugan & Prakasam, 2013). Delivering FOSS projects not scrutinized for quality assurance makes developers fail to adequately include most improvements that account for users' desires (Hedberg et al., 2007).

FOSS development paradigms still challenge traditional methods for achieving UXA (Hedberg et al., 2007) due to the ad hoc nature of the FOSS development process (Paas & Ayres, 2014), in which the developers determine the outputs. In addition, the FOSS community primarily ignores UXA practices because it lacks explicit norms to follow and a distinctive design and planning framework (Bahamdain, 2015; Van Angeren, Kabbedijk, Jansen, & Popp, 2011).

The FOSS community's literature analysis has found fourfold research gaps on the UXA, requiring immediate research initiatives. First, the onion-like core-periphery structure of the community precludes the simple integration of users with the necessary UX expertise into FOSS development processes. At the same time, the participating developers frequently lack essential UX skills. Nevertheless, it does not support initiatives for integrating UXA-related initiatives, which calls for quick research into how to make the FOSS community's structure more adaptable to users who are not programmers. Second, the FOSS community requires techniques for evaluating UXA because better-organized and manageable processes yield desirable UXA. However, the community is committed to utilizing ISO 15504 and CMMI, which do not entirely suit because they lack relevant processes. As a result, future researchers should focus on establishing FOSS-specific assessment methodologies. Third, traditional UCD approaches, the main ingredient of UXA, do not correctly fit the dispersed development philosophy of the FOSS community; hence, future research must look into how to adapt them to provide the FOSS philosophy. Finally, using user feedback as a significant source of information for UXA is not always accurate because many ordinary users do not know how to report practical issues appropriately. As a result, this community will need to put in more effort to make UXA the norm.

2.3.8 Issues in the FOSS Community's Design for UX

Although several desktop-based FOSS projects exist, they are not adequately adopted (Colombo et al., 2014; Namayala et al., 2022). They are considered burdensome for average users (Crahmaliuc, 2021; Despalatović, 2013; Nichols & Twidale, 2006), mainly caused by problems related to design for UX (UXD). The FOSS projects must ensure superb UX and make UXD the core functionality to attain their full potential (Duffy, 2018). UXD is sometimes mistaken and used interchangeably with usability and user interface design (UID), a big misconception in the UX design field. Usability and UID are just parts of UXD (IDF, 2018). Following SE's suggested procedures, UXD should look at the what (functionality features), why (user motivations, values, and views), and how (accessibility and aesthetics) of products in use.

Several UXD challenges exist in the FOSS community, and they fall under several categories depending on the study's perspectives. For example, Duffy (2018) uncovered three main challenges. (i) FOSS UX efforts concentrate on specific tools rather than higher-level processes, whereas UXD is inconsistent and lacks integration. (ii) FOSS projects require extensive setups compared to similar proprietary software and necessitate users with specific knowledge to configure and operate them, resulting in UX debt. (iii) The FOSS community concentrates on creating software from “itch and scratch” rather than meeting user requirements (Raymond, 2005), as FOSS developers neither market their projects to a broad audience nor value user requirements.

Moreover, the culture of the FOSS community does not support UXD initiatives (Raza et al., 2012a). As a result, UXD has never secured a proper place in the development life cycles of FOSS (Anitha & Prabhu, 2012). Nichols and Twidale's (2006) study supports these arguments and confirms that developers’ perceptions, community integration, and process constraints are the only influencers of UXD interventions in the community.

The current UXD difficulties and their effects are well known to the FOSS community. As a result, significant ideas and contributions from studies, such as Duffy (2018) and Nichols and Twidale (2006), emerged to address the problems, and their suggestions have been adopted and put into practice by Microsoft, Novel, Sun Microsystems, and IBM.

For example, Duffy (2018) provided fourfold recommendations where organizations must first avoid echo members, whereas those users who are not community members will be allowed to submit feedback. Second, boundaries must be established by building a transparent decision-making process or clear accountability. Third, avoid conflict with transparency by illustrating design phases such as the problem, required research, prototyping, and engagement with community members. Finally, add functionality with cautions that avoid detrimental influences on UX by accepting more functionality than a project can commit.

In contrast, Nichols and Twidale's (2006) study recommended five ideas. First, it convinced the FOSS community to adopt commercial software strategies, something Mozilla had previously tried. Second, check interface compliance using automated methods and report current conditions. In the third recommendation, the study encouraged utilizing students enrolled in HCI classes. Fourth, engage non-technical users. Finally, create a social and technological environment that honors and welcomes UX professionals.

Given the nature of the FOSS community, embracing all of Duffy's (2018) and Nichols and Twidale's (2006) recommendations might be unbearable. Moreover, several academics, such as Buxton (2007), suggest that designing better and UX-desirable software requires combining design and development phases. This recommendation highlights the importance of integrating UXD and Requirement Engineering (RE) into the life cycles of FOSS development processes. However, this integration is impossible in the FOSS community before projects are delivered (Llerena et al. 2018).

The status quo shows that the FOSS community gradually adopts UXD practices such as "design-by-blog," where development teams put new designs on a blog for stakeholders' feedback (Terry et al., 2010). However, as reported by Llerena et al. (2019), Rajanen and Iivari (2015a), and Terry et al.(2010), this study still sees the FOSS community must innovate other strategies to implement UXD as a means of improving UX of its projects.

2.3.9 UX Capability Maturity Models and Imposed Challenges

UXCMMs show how organizations have adopted HCD into their team culture, design process, and digital product development. They generally aid in describing organizations' actual implementation of UX issues in their development processes. The completed literature review identified several UXCMMs with different degrees and emphases. However, most UXCMMs are theoretically identical and use different classifications and phrasing to express various concepts (Sauro et al., 2017). They all follow uniform scales that start with obliviousness to UX and culminate with UX Design integration (Jokela et al., 2006; Nielsen, 2006; Sauro et al., 2017). There are several degrees of maturity between these two extremes, often separated and labeled based on the creator's research, angle, and target audience. However, some practitioners have begun to doubt their legitimacy or regard them as nothing more than sales tricks (Sauro et al., 2017).

Rohles (2021) and Drahun (2017) have compiled detailed lists of UXCMMs and attempted to explain each UXCMM. For example, Rohles (2021) has illustrated four UXCMMs, and Drahun (2017) has provided a collection of over twenty UXCMMs. Moreover, Namayala et al.(2022) investigated how current UXCMMs are applicable in appraising the FOSS community's UX and found that many are not the perfect fit for the community. The findings from Namayala et al. (2022) confirm the results of the studies by Bornoe and Stage (2014), Lacerda and von Wangenheim (2018), Mashapa et al.(2013), and Raza et al.(2012a).

Sauro et al.(2017), Nielsen (2006), Carraro (2014), Anchahua et al. (2018), and Pernice et al.(2021) have also thoroughly explored why existing UXCMMs may not be suitable in the FOSS community and support the necessity of creating new UXCMMs. The studies found that contemporary UXCMMs may not be ideal for the FOSS community because they may not accommodate enough or the right UX metrics (Carraro, 2014; Mashapa et al., 2013; Pretorius et al., 2015) and are still indeterminate what makes a system with optimal UX (Bargas-Avila & Hornbæk, 2011; Chapman & Plewes, 2014; Pretorius et al., 2015; Sauro et al., 2017).

Moreover, many encountered UXCMMs are not precise on how users may self-evaluate their projects and emphasize looking at provided indicators to predict the maturity level (Ten & Paz, 2017; Terry et al., 2010). These UXCMMs are also specific to a particular UX feature and often take longer to reach the highest maturity level (Sauro et al., 2017). Finally, current UXCMMs are unclear whether they consider contemporary software development movements (Salah, Paige, & Cairns, 2014).

Nevertheless, there are other critical issues surrounding UXCMMs that need further investigation. For example, why do UXCMMs sell their idea by implicating UX to cost only? Is this view always valid, particularly for start-ups and FOSS communities where resources are always the issue? Are there any other ways of enhancing organizational UX rather than engaging UXCMMs? Why are almost all UXCMMs developed from owners' experience? Finally, is it possible to establish UXCMMs experimentally or through lab testing?

Although several initiatives to assess UX in the FOSS community exist, this thesis still backs up Sauro et al. (2017) claim that UX varies by organization. Sauro et al. 's (2017) assertions necessitate using unique methods and instruments to evaluate UX maturity and show how it correlates with organizational success. Moreover, in natural settings, organizations have different missions associated with adopting UXCMMs. For example, some strive to be more successful than reaching a higher maturity level (Sauro et al., 2017), and others strive for the opposite.

Few studies, such as Buley's (2019), explain the efforts to express the link between UX maturity and organizational success. Buley's (2019) study, the most extensive and recent study, found that a company with high design maturity enjoys cost savings with augmented revenue, brand, and market. It implies that UX assessment practitioners might use the methods proposed by Buley's (2019) study to justify the link between UX maturity and organizational success.

As a result, practitioners and researchers have created UXCMMs in response to the necessity of developing unique methodologies and tools for the UX maturity assessment of multiple organizations. For example, the UX Maturity Model by Feijó (2010), the UX Maturity Model for e-commerce websites by Anchahua et al. (2018), the Keikendo UX Maturity Model by Carraro (2014), the Corporate UX Maturity by Nielsen (Nielsen, 2006), the UX Maturity Model: A Tool To Improve Collaboration with the UX Team by Damsa (2017), The 6 Levels of UX Maturity by Pernice et al.(2021), the UX Maturity Model: From Usable to Delightful by Sakhardande & Thanawala (2014), the Corporate User-Experience Maturity Model by Van Tyne (2009), and A UX Maturity Model: Effective Introduction of UX into an organization by Chapman and Plewes (2014) to mention a few. However, it is difficult to grasp how current UXCMMs are related and distinguished. No studies compare their features, operations, and methodologies (Namayala et al., 2022). Their differences with older Usability Capability Maturity Models (UCMMs) are likewise hazy, sparking controversy among practitioners and academics.

Other academics and practitioners look at UX assessment from a very different perspective. They assume that since current UXCMMs cannot adequately and effectively address UX concerns in the FOSS community (Adewumi et al., 2013; Kuwata et al., 2014; Petrinja & Succi, 2012; Rukonić et al., 2019; Umm-E-Laila et al., 2017), Adopting or adapting old Usability Capability Maturity Models (UCMMs) may be more effective in assessing UX maturity and hypothesize that if UCMMs were invented today, they might have been dubbed UXCMMs. As a result, future researchers are encouraged to look at the effectiveness of UCMMs in assessing UX.

2.3.10 Usability, UX Metrics, and Measurement Methodologies Timeline

With the support of different techniques, the notion of using metrics while analyzing UX and usability is expanding rapidly (Marques et al., 2021). However, UX has not yet established its metrics effectively. Current trends indicate that practitioners from academic institutions and industries frequently use usability metrics and tools when evaluating UX (Osinusi, 2020). For example, the review study of Apraiz et al. (2020) has listed thirty-two standardized questionnaires commonly used to assess usability and UX while looking at mainly usability metrics.

These metrics include task success rate and completion (time, average completion time, meantime failure, and average time on tasks). Other usability metrics include retention rate, conversion rate, error rate, satisfaction (using satisfaction survey, Customer Satisfaction Score, Net Promoter Score, Customer Effort Score, and social media monitoring), and heuristic evaluation.

The UX assessment community is not left behind. It has developed several standardized questionnaires for assessing UX, for example, AttrakDiff by Hassenzahl et al.(2003), UEQ by Hinderks (2015), meCUE by Minge et al.(2017), SASSI by Hone and Graham (2000), SUS by Brooke (1996), SUISQ by Polkosky (2008), MOS-X by Schmidt-Nielsen (1992), PARADISE by Walker et al. (1997), and narrative and contextual analysis. Among them, Perceived Usefulness and Ease of Use by Davis (1989), PANAS by Watson et al.(1988), SUS by Brooke (1996), PANAS-X by Watson and Clark (1994), and QUIS by Wallace et al.(1988) are highly cited. However, it is still challenging to link usability metrics with the design for UX-related decisions in several organizations (Feng & Wei, 2019; Pavliscak, 2014) due to a lack of cohesion (Gil Urrutia, Brangier, Senderowicz, & Cessat, 2018).

UX metrics always signal the effectiveness of UX strategy (Feng & Wei, 2019; Pavliscak, 2014), keep track of changes over time, compare iterations to competitors, and set goals. However, exhaustive literature reviews show that they are either at a superficial level or indirect, making it difficult to assess the impact of user interface changes, and they may also have ambiguous interpretations. Efforts to quantify UX that engage the selection of standardized frameworks for selecting UX metrics are ongoing; however, stakeholders are still divided. According to Rodden et al.(2010), practitioners commonly use PULSE metrics: Pageviews, Uptime, Latency, Seven-day unique active users, and Earnings. However, using PULSE is still associated with several limitations. For example, it may provide ambiguous indicators and perceptions of UX.

Due to PULSE's deficiencies, many initiatives have emerged to aid the accurate identification and selection of UX metrics. These efforts have matured and created frameworks whose names and how they group UX metrics are as follows: The AARRR, which is a set of five user-behavior metrics that product-driven growth companies should monitor: Acquisition, Activation, Retention, Referral, and Revenue by McClure (2007), The Customer Experience Index: descriptive, perception, and outcome by Forrester (2019), and The Google HEART: Happiness, Engagement, Adoption, Retention, and Task Success by (Rodden et al., 2010). However, designers may only select UX metrics that fit their projects and must be cautious, as current techniques stress combining qualitative and quantitative metrics. Moreover, Pavliscak's (2014) study on the big picture of UX metrics has provided significant categories of UX metrics that every framework must adapt, including usability, engagement, and conversion.

Proper tracking of UX metrics requires measurements of the selected metrics. Moreover, the researcher must appropriately choose dimensions (goal, approach, data, granularity, and setup), constructs (utility, usability, aesthetics, identification stimulation, and value), methods (implicit, explicit, and creative), and focus of the research on a specific area of interest (Marques et al., 2021). Doing so increases the likelihood that end products will meet functional requirements and, at the same time, be enjoyable to end users (Ganglbauer, Schrammel, Deutsch, & Tscheligi, 2011).

In addition to questionnaires, there are also several other approaches for automating the recognition of users' expressions in speech, visage, or gestures (Gomez, Stahel, & Danuser, 2004). They include the layered emotions assessment tool (Huisman & Van

Hout, 2008) and psychological approaches for assessing valences to discover social masking and avoiding user interruptions, for example, the valance method by Burmester, Mast, Jäger, and Homans (2010). This method has resulted in detecting a more comprehensive range of emotions (De Silva, 2004; Gunes & Piccardi, 2006).

The Psychological approaches and those automating the recognition of users' expressions in speech are more practical for studying data in unique settings. (Ganglbauer et al., 2011). However, they are still expensive and difficult to implement because they bind people with cables and electrodes, making them unable to move and act freely. However, as a norm in UX assessment, UX assessment techniques must be quick, light, and affordable, allowing iterative procedures, requiring little skill to perform well, and being appropriate for various user groups (Väänänen-Vainio-Mattila, Roto, & Hassenzahl, 2008).

UX and usability are conceptually distinct, and it is erroneous to continue presuming that measurements for usability are the ideal fit for UX evaluation. Nevertheless, several scholars, including Sagar and Gupta (2018), support this argument. However, the usability community regards heuristic evaluation principles as empirical rules of thumb, norms, and conventions observed and evaluated over time (Jimenez, Rusu, Roncagliolo, Inostroza, & Rusu, 2012; Neilsen, 1995). It is uncertain whether the principles of heuristic evaluation and Cognitive Walkthrough by Polson et al. (1992) apply to evaluate UX (Noor, Zainuddin, Ibrahim, & ..., 2020). Heuristic evaluations and cognitive walkthrough methods do not substitute user testing with real users. User testing can provide more detailed insights into how users interact with a product and help identify usability issues that may not be apparent through heuristic evaluation or cognitive walkthrough alone.

There are multiple sets of usability heuristic principles. However, Neilsen's (1995) principles are regularly employed (Krawiec & Dudycz, 2020a; Quiñones & Rusu, 2017). In addition to Neilsen's (1995) principles, the study of Krawiec and Dudycz (2020a) identified nine other sets of usability heuristics: Cognitive Engineering Principles for Enhancing Human-Computer Performance by Gerhardt- Powals (1996), Weinschenk and Barker classification by Weinschenk and Barker (2000), and Heuristic Evaluation in E-Government Website Development by Sivaji et al.(2011). Others include First Principles of Interaction Design by Tognazzini (2015), Guidelines for Designing Mobile Apps by Al-nuiam and Al-Harigy (2015), The Eight Golden Rules of Interface Design by Shneiderman and Plaisant (2006), SMASH: A Set of SMArtphone's uSability Heuristics by Inostroza et al.(2016), Usability Heuristics for Mobile Applications by Durães Dourado and Dias Canedo (2018), and Heuristics for Assessing the Usability of the Public Information Bulletins in Poland by Krawiec and Dudycz (2020b).

Modern usability assessment techniques strongly emphasize fact-based rather than subjective-based tools, which conveniently help researchers evaluate whether deliverables are usable. However, realizing these expectations has not always been the outcome. In most cases, deliverables are vulnerable to input changes (Namayala et al., 2022). Moreover, the community that assesses UX and usability continues to hold that higher usability indicates acceptable UX, and they view it as a requirement for measuring UX.

As a consequence of the literature review, the study found that assessing UX in the FOSS community is still challenging because it is situational, temporal, context-based, and difficult to comprehend (Karapanos et al., 2009). Meanwhile, the author is also skeptical if assessed organizations are adequately informed, prepared, and agree on factors responsible for delivering products with desirable UX. A consistent understanding of UX is likewise lacking in the body of knowledge (Hassenzahl, 2018). Nevertheless, the dynamics of the FOSS community are also different from other software development communities (Çetin, Verzulli, & Frings, 2007; Masson et al., 2017; Namayala et al., 2022; Raza et al., 2013; Terry et al., 2010).

Moreover, just a handful of UX methodologies are created explicitly for software application review (Rivero & Conte, 2017), and those available are vulnerable to several limitations. For example, none of the methods helps measure the early stage of projects, and several are not networked. They lack practicability and a deeper UX understanding (Vermeeren et al., 2010). Nevertheless, there is little evidence that available UX techniques adequately identify issues that may impact UX (Marques et al., 2021).

Like Kocaballi et al. (2019), this study has identified gaps in present UX evaluation methods, which invite more comprehensive analyses and improvement efforts. Generally, the available UX evaluation techniques do not solve the problem that leads to undesirable UX in many software-developing communities, including the FOSS community (Marques et al., 2021). The FOSS community must, therefore, adopt earlier studies' endorsements to address imminent UX issues in the community by reconceptualizing present HCI, usability, and UX methods and styling them to match its dynamics (Namayala et al., 2022; Terry et al., 2010).

2.4 Conceptual Framework

As a result of encountering problems in getting dimensions from all specialties and thoroughly analyzing current conceptual frameworks, this study has adopted a bottom-up approach and snowballing to create its UX conceptual framework., depicted in **Figure 2.7**. The designed framework believes that it has gone beyond the standard dimensions of the user, the product, and the interaction contexts proposed by several scholars such as Hassenzahl and Tractinsky (2006), Forlizzi and Ford (2000), Mashapa (2013), and Arhippainen and Tähti (2003). The standard dimensions do not adequately explain the big picture of UX parameters from multi-faceted UX domains (Hellweger & Wang, 2015).

In this framework, proposed UX dimensions with elements or attributes largely depend on UX components (user, product, context supporting the interactions between user and products, objectives/goals, and period or timeline). Therefore, UX components significantly influence the features included in the FOSS community's UX dimensions (functionality, usability, psychological, cognitive, non-instrumental qualities, interactivity, trustworthiness, ambiguity, and engagement). The UX maturity influencing factors are influenced by UX dimensions, endorsements from earlier studies, UX scholars' opinions, and other UX stakeholders' perceptions and understanding. The UX maturity influencing factors and how prepared the FOSS community is to assess its UX maturity, in turn, affects the performance of the FOSS community in improving its UX maturity.

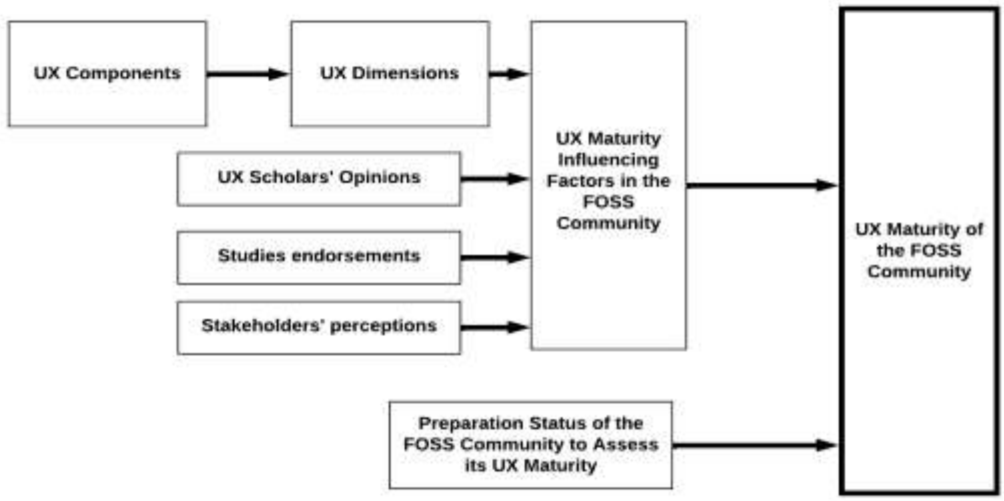

Figure 2.7: Conceptual Framework for Optimizing UX

Academics and practitioners have classified present UX frameworks into three different perspectives: product-centered, user-centered, and interaction-centered; however, it is still reasonably complex to design for UX (Forlizzi & Battarbee, 2004), and present UX frameworks do not provide a valid representation of all present variables. There is little agreement among practitioners and researchers about dimensioning UX (Bongard-Blanchy et al., 2015; Hussain, Mkpojiogu, & Husin, 2021; Provost & Robert, 2013). It is also unclear how to establish the list of variables per dimension. Nevertheless, researchers have yet to explore several dimensions of UX, and no single research has attempted to encompass them all. Therefore, the current dimensioning appears insufficient for a complete design and evaluation of digital applications. (Berni & Borgianni, 2021; Hussain et al., 2021), and finding UX features that are genuinely ubiquitous has proven difficult (Roto & Rautava, 2008). **Table 2.9** demonstrates the UX dimensions of current frameworks.

The existing knowledge confirms that design, business, philosophy, anthropology, cognitive, and social science principles contributed to the current UX frameworks (Forlizzi & Battarbee, 2004). However, depending on the context and domain of usage, there is still room to design and dimension UX more effectively.

Table 2.9: Present Frameworks' Features, Attributes, and Dimensions of UX

No.	Author(s)	Features, Attributes, or Dimensions
1.	Jordan (2000)	Functionality, usability, and pleasure (physio, socio, psycho, and ideo)
2.	Forlizzi and Battarbee (2004)	Fluent, cognitive, expressive, experience, and co-experience
3.	Mahlke (2005)	Instrumental qualities, non-instrumental qualities, affective reactions, judgments, and behavior or emotional consequence
4.	Hassenzahl and Tractinsky (2006)	Beyond the instrumental, the experiential and affect and emotion.
5.	Mahlke and Thüring (2007)	Instrumental, non-instrumental quality perceptions, emotional user reactions, and appraisal
6.	Kort et al. (2007)	pragmatic, aesthetic, and semantic elements as UX design objectives.
7..	Desmet and Hekkert (2007)	Aesthetic experience, the experience of meaning, and emotional experience
8.	Roto and Rautava (2008)	Utility, Usability, Social value, and Enjoyment
9..	Robert and Larouche (2012)	Functional, physical, perceptual, cognitive, social, and psychological
10.	Hellweger and Wang (2015)	Context, Usability, Product Properties, Cognition, Needs, Purpose, and Time.
11.	Hassenzahl (2018)	Hedonic (stimulation, identification, communicating identity, provoking valued memories and evocation), pragmatic (functionality and usability), and emotional or behavioral outcomes, including appeal, pleasure, and satisfaction.
12.	Hussain et al. (2021)	Pragmatics, Hedonic, Affectivity, Aesthetics, Self-determination, Trust, Engageability, Interactivity, Sociability, and ubiquity

The study has proposed new UX dimensions (see **Table 2.10**). As also concluded by Robert and Lesage (2017b, 2017a), the author is confident that the proposed UX dimensions better represent where to start when perfecting UX dimensions and consequently establish factors that affect UX maturity, particularly in the FOSS community. Nevertheless, it better connects product features to user needs and values (Hassenzahl & Tractinsky, 2006) and aids in the scientific development of new UX dimensions frameworks with the same level of detail (Provost & Robert, 2013). In line with what Berni and Borgianni (2021) found, the proposed UX dimensions may be a possible reference for future research on the design of UX in applications.

Table 2.10: Proposed UX Dimensions and Factors

No.	Dimension	Factors	Reference(s)
1.	Functionality	Compatibility, task completion, security, reliability, and performance	Hassenzahl (2003, 2018), Hellweger and Wang (2015), Robert and Lesage (2017b), and McNamara and Kirakowski (2006).
2.	Usability	Learnability, efficiency, error rates, satisfaction, accessibility, and memorability	Hassenzahl (2003, 2018), Mahlke (2005), McNamara and Kirakowski (2006), Nielsen (2010, 2012), Hustak and Krejcar (2016), and Hellweger and Wang (2015).
3.	Psychological	Emotions, motivations, and perceptions	Robert and Lesage (2017b), and Robert, Larouche (2012), and Thüring and Mahlke (2007).
4.	Cognitive	Information architecture, navigation, mental models, attention, and memory	Robert and Lesage (2017b), Forlizzi and Battarbee (2004), and Hellweger and Wang (2015).
5.	Non-instrumental qualities	Aesthetics, branding, user engagement, user delight, and social factors	Hassenzahl (2003, 2018), Hussain et al.(2021), (Kort et al.(2007), Mahlke (2005), and Mahlke and Thüring (2007)
6.	Interactivity	Feedback, Interactivity modes, Contextual interactivity, Responsiveness, and Collaborative interactivity	Hussain et al.(2021), Othman et al. (2011), and Schild et al. (2012).
7.	Trustworthiness	Privacy, Security, Authenticity, Dependability, and Transparency	Hussain et al.(2021).
8.	Ambiguity	Clarity, consistency, complexity, and flexibility	Hussain et al.(2021).
9.	Engagement	Personalization, challenges, and novelty	Hussain et al.(2021) and Schild et al. (2012).

Functionality

Functionality is a critical component of UX and encompasses several essential factors for creating effective and useful products for users. It mainly represents technical issues that only affect the outcome and could include the device's use, maintainability, and reliability (Hassenzahl, 2018; Hellweger & Wang, 2015; McNamara & Kirakowski, 2006). Among other incentives bought by functionality in UX, it assists users in achieving their goals and needs. Steve Jobs from Apple Inc. once said, *“..Design is not just what it looks and feels like; it is how it works.”*

Usability

Several definitions for usability exist. However, researchers and practitioners commonly use the definition of ISO (2018), which defines usability as “the extent to which specified users can use a product to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context.” Moreover, according to Nielsen (2010) and Hustak and Krejcar (2016), usability is made by five influencing factors: learnability, efficiency, memorability, error rates, and satisfaction.

Usability is a significant component of UX that helps users, developers, and managers of projects in the software development community and has occasionally been confused with UX (Noor et al., 2020). For example, it assists users in completing tasks accurately while maintaining a pleasant mood, developers in determining the system's effectiveness, and finally, managers in expanding their customer base

Psychological

The psychology dimension is an essential component of the UX design process because it aids in thoroughly understanding potential users of systems or products. It focuses on users' emotional and cognitive responses when interacting with a product or system. According to Rusell (1980, 2003) and Russell and Barrett (1999), emotions are often built from two “core affect,” valence (positive or negative) and arousal, which is a result of activating or deactivating the nervous systems. Several tools for assessing emotions consider active, interested, excited, strong, enthusiastic, proud, alert, inspired, determined, and attentive as lexicons for positive affect and afraid, jittery, nervous, ashamed, scared, hostile, irritable, distressed, upset and guilty as lexicons for negative affect.

Although there are still principles that overlap with the cognitive dimension, nine principles guide the psychological phenomena in UX design: The Principle of Least Effort, The Principle of Perpetual Habit, The Principle of Socialization, The Principle of Emotional Contagion, The Principle of Identity, The Principle of Beauty, The Magical Number Seven Plus or Minus Two, The Psychology of Mistakes, and Focus, Attention, and Concentration.

Cognitive

The Cognitive dimension examines how humans acquire, analyze, and store information in their memory after interacting with systems or products. It consists of analyzing, evaluating, reflecting, learning, and creating processes that allow users to comprehend, gather information and experience, progress, and consolidate competence (Robert & Lesage, 2017a; Robert & Larouche, 2012).

In other words, cognitive processes connect environmental stimuli information to the brain to arrive at a semantic and aesthetic interpretation of the inputs (Hassenzahl, 2003). Moreover, they also enable evaluation depending on the user's preferences. Cognitive generally refers to users' desire to grow and improve themselves, make sense of a situation, and live an artistic experience.

Non-instrumental qualities

Non-instrumental qualities refer to aspects of the user's experience that are not directly related to the functionality or usability of the product or system.(Goh & Karimi, 2014; Hassenzahl & Tractinsky, 2006). They can significantly impact user satisfaction and overall UX. Some critical components of non-instrumental qualities in UX include aesthetics, branding, user engagement, user delight, and social factors.(Hassenzahl, 2001, 2003). Aesthetics refers to the product or system's visual design and overall appearance. It is divided into different scopes regarding human senses, such as visual, haptic, and acoustic (Mahlke, 2008), and symbolic values linked to messages the product communicates, such as cheerful, friendly, expensive, childish, tedious, or rude (Crilly, Moultrie, & Clarkson, 2004).

Nevertheless, branding refers to how the product or system is presented and marketed. Effective branding can help create a positive association with the product and encourage user loyalty. User engagement is the degree to which users feel engaged and invested in the product or system. Engaging products can enable users to use them more and explore their features and capabilities. User delight is the degree to which users feel a sense of pleasure or enjoyment when using the product or system. Products that evoke delight can help to create a positive emotional experience for users. Finally, Social factors refer to how the product or system facilitates social interaction and user connections. Social elements can be essential for products designed for collaboration or communication.

When evaluating an interactive product, non-instrumental attributes are crucial because they serve as deciding factors and help attain behavioral goals related to products or systems (Goh & Karimi, 2014).

Interactivity

The interactivity dimension stores the UX characteristics that assess user interaction. It describes how well a product or system allows interaction between the user and the product or system. Several scholars have defined it as a multifaceted concept (Heeter, 2000) or concentrating on the medium's functional aspects (Sohn & Lee, 2006). For instance, Laurel (1997) described it as a three-dimensional concept: frequency, range, and significance. Steuer (1992) further conceptualized the idea based on three elements: speed, scope, and mapping, which allow users to manipulate content more easily. Coyle and Thorson (2001) highlighted mapping, speed, and user control as three essential elements of interactivity based on the functional approach. Perceived control, perceived responsiveness, and perceived personalization are three underlying interactivity characteristics, according to Sohn, Leckenby, and Jee (2000). Moreover, McMillan and Hwang (2002) highlighted three frequently mentioned interactive characteristics: communication direction, user control, and time.

The interactivity dimension is critical to UX because it affects individuals' information processing and decision-making abilities (Sohn & Lee, 2006).

Trustworthiness

The trustworthiness dimension is often a quality attribute connected to the systems' trustworthiness. Privacy, credibility, dependability, and transparency are some of these characteristics. Bedford (2016) from Nielsen and Norman Group specified that various products, including websites, must develop confidence and be reliable to convert visitors into customers. According to Nielsen (1999), products such as websites can communicate trustworthiness in four ways: design quality, upfront disclosure, extensive and up-to-date material, and integration with the rest of the product or web.

Ambiguity

According to Linse (2017), knowledge creation is often obscure to the core. The term "ambiquitous" is expressed in various ways in the literature. Most of the time, synonyms such as uncertainty, equivocalness, and complexity are used interchangeably and sometimes with different connotations. Based on epistemological practice, ambiguity can be separated from individual flaws and instead associated with contextual and entwined contradictions that manifest in human behavior (Gherardi, 2012; Nicolini, 2011). According to Weick (1995), ambiguity and uncertainty arise due to confusion from multiple interpretations or ignorance due to a lack of interpretations of statements, situations, or artifacts.

As discovered by Axelsson et al.(1977), there are four main occurrences of ambiguity: the ambiguity of intent, the ambiguity of understanding, the vagueness of narrative, and the ambiguity of organization. Several researchers, including Linse (2017), acknowledge the lack of desirable effort to negotiate ambiguity in software design and development, exploiting or reducing ambiguity's generative and transformative power. Ignoring ambiguity and not engaging in negotiating activities means choosing ignorance over enlightenment. Nevertheless, there is evidence that users cease or change their behavior regarding specific systems, deliverables, or products because of ambiguity issues.

Engagement

The definition of user engagement remains ``clumsy" and fails to provide meaningful enlightenment when communicated among researchers and designers. Generally, it is described based on the depth of the actor's cognitive, temporal, emotional, and behavioral investment in interacting with digital systems (O’Brien, 2016). It mainly assesses users' interaction with technological artifacts. Moreover, user engagement represents a desire to use an application more often (Lalmas et al., 2014), depicts users' deliberate choices to attain what they want, and includes information flow, presence, and playfulness.

Users who seek to satisfy their pragmatic and hedonic needs while using technology invest much of their time, emotions, and attention, which is technically called user engagement. Moreover, several studies, including Lalmas et al. (2014), confirm that user engagement (engaging interaction *and* long-term engagement) is a multifaceted and complex phenomenon of three dimensions: behavioral, cognitive, and affective.

Like many other fields, the computer science UX design society may reap numerous remunerations by embracing strategies that foster and guarantee higher user engagement. These benefits may include understanding UX (McCarthy & Wright, 2007), motivating stakeholders (Fredricks, Blumenfeld, & Paris, 2004), and analyzing online interaction (Lehmann, Lalmas, Yom-Tov, & Dupret, 2012; Thomas, O’Brien, & Rowlands, 2016). Others include guiding autonomous systems (Bohus & Horvitz, 2009) and appraising technologies well-being, civil services, and social networks, to mention a few (Jung & Lee, 2016; Kim & Kleinschmit, 2012; Linnemeier, Lin, Laput, & Vijjapurapu, 2012; Wiezer, Bakhuys Roozeboom, & Oprins, 2013).

2.5 Related Work

The exhausted literature shows that the FOSS community lacks consistent interventions to optimize the design for UX in developing desktop-based projects. For example, the FOSS community does not consider endorsements from previous studies. It lacks community-based UX maturity influencing factors, is unprepared to assess its UX maturity, and does not adequately involve stakeholders and other UX experts. The survey of Namayala et al. (2022) supports a conclusion reached by demonstrating the absence of UXCMMs dedicated to the FOSS community to account for its unique dynamics. Nevertheless, available UCD and HCI practices may not be valid in this community, such as conducting user research and usability tests, creating personas, using design patterns and agile methods, and engaging with the community (Bornoe & Stage, 2014). The FOSS community has unique characteristics compared to other software-developing communities, which may demand unique treatments (Terry et al., 2010).

On a small scale, some scholars have indicated an interest in enhancing the UX of the FOSS community. However, they still mix it from usability (Bargas-Avila & Hornbæk, 2011; Sauer, Sonderegger, & Schmutz, 2020; Tractinsky, 2018) and forecast tactics for optimizing usability to also apply in optimizing UX. As a result, the FOSS community has witnessed several efforts to enhance usability, such as developing the Open Source Usability Maturity Model (OS-UMM) by Raza et al. (2012a). The OS-UMM has resulted from four different empirical studies that involved developers (Raza, Capretz, & Ahmed, 2010), users (Raza et al., 2012b), industry (Raza et al., 2013), and contributors (Raza & Capretz, 2010) within the FOSS community (Raza et al., 2012a). It has adapted the structures of the CMM and QMMG, and its performance scale is a derivative of the BOOTSTRAP methodology by Wang and King (2000). Moreover, OS-UMM consists of five maturity stages: recognized, preliminary, defined, standardized, and streamlined, and eleven significant factors are organized into four dimensions: usability methodology, design, assessment, and documentation (see **Table 2.11** and **Figure 2.8**). Nonetheless, OS-UMM is based on self-assessment methodologies that use questionnaires as the primary assessment tools that identify the usability maturity level by establishing the degree to which project managers and developers agree with each questionnaire statement using the proposed rating scales and thresholds (Raza et al., 2012a).

Table 2.11 **The Maturity Stages and Description of OS-UMM**

No.	Name	Description
1	Preliminary	The FOSS project lacks a consistent and well-organized technique for achieving usability. There are no resources or proof that teams practice usability and have precise methods for gathering requirements, designing, gathering feedback, documenting, assessing, and training.
2	Recognized	The project team acknowledges the importance of usability in the success of their project and creates infrastructure for usability implementation. It also demonstrates an interest in usability by collecting user requirements and comments. Project developers learn about the UCD approach and the need for documentation.
3	Defined	The level lays the groundwork for establishing usability. Project managers comprehend, define, and apply UCD principles. The project team can gather and fulfill users' requirements and expectations, collect users' input, and employ critical technological abilities to provide users with a convenient usability bug reporting facility and conduct necessary usability testing. Finally, team members can improve a project's understandability, learnability, and attractiveness.
4	Streamlined	The project team has obtained adequate resources to satisfy the needs of its users, and it has built a management system for recording user feedback and taking appropriate action to address it. New development team members must understand usability principles and guidelines, and project teams frequently confer with usability specialists on the definition and application of UCD principles. Through a holistic monitoring structure, the team frequently monitors initiatives' understandability,

No.	Name	Description
		learnability, operability, and attractiveness. Furthermore, regularly use quantifiable usability metrics to assess and maintain project documentation.
5	Institutionalized	The level regards usability as an asset required for obtaining and maintaining success. FOSS initiatives have sufficient resources and abilities to collect user input and understand user expectations. The project team commits to usability learning and UCD methodology to improve understandability, learnability, and attractiveness. It uses quantitative metrics to assure usability assessment success and constantly improve the bug reporting service.

Moreover, the names of the eleven factors that OS-UMM contains include User Requirements (URs), User Feedback (UFB), Usability Learning (UL), User-Centred Design (UCD), Understandability(U), Learnability (L), Operability (O), Attractiveness (A), Usability Bug Reporting (U), Usability Testing (UT), and Documentations (D).

Dimension No.	Dimension	Practice No.	Key Usability Factors
1	Usability Methodology	1	Users' Requirements
		2	Users' Feedback
		3	Usability Learning
2	Design Strategy	4	UCD Methodology
		5	Understandability
		6	Learnability
		7	Operability
		8	Attractiveness
3	Assessment	9	Usability Bug Reporting
		10	Usability Testing
4	Documentation	11	Documentation

Figure 2.8: OSS-UMM Usability Factors per Dimension

SOURCE: Raza et al.(2012a)

By developing an assessment methodology and completing case studies, OS-UMM helps assess the current usability maturity of FOSS projects (Raza et al., 2012a). However, it primarily depends on the subjective opinions of project managers and developers. It also only employed a small amount of data from FOSS project developers and managers in its creation. Nevertheless, it does not guide how to improve quickly after completing an evaluation. Finally, conducted case studies did not account for independent assessments.

2.6 Summary of the Chapter

This chapter provides a literature review regarding technological advancement in optimizing the design for UX in the development process of desktop-based FOSS projects. It has further examined theories that guided the study's implementation. This chapter has built the foundation for executing this study through gap identifications. Generally, the analysis of the existing technologies in optimizing the design for UX to enhance the adoption of desktop-based FOSS projects revealed that they do not fit well in the FOSS community. Chapter three presents the research methodologies the study adopted to address identified technological gaps and consequently improve the adoption of desktop-based FOSS projects.

CHAPTER THREE
METHODOLOGY

3.1 An Overview

The preceding chapter, i.e., chapter two, provides an overview of the relevant literature timeline that served as a foundation for optimizing the design for UX in desktop-based FOSS projects. This chapter explains the path taken to investigate and endorse solutions. It started with the chapter's overview, followed by study location and selection justification, research approach, research design, and study settings. It has also discussed the study population, sampling procedures and sample sizes, data collection techniques, data processing and analysis, ethical considerations, and validity and reliability. Moreover, the chapter comprises the development techniques of FOSS-UXMM and the implications of theories guiding this study: Technological Acceptance Model (TAM) schema theory and gestalt theory. Finally, the chapter concludes with a summary.

3.2 Study Location

The researcher conducted this study online in sixty-two desktop-based FOSS projects purposively selected from five (5) FOSS source code hosting platforms or repositories (see **Appendix XIII**). The study utilized features of internet technology when collecting data. Although multiple repositories were visited, many desktop-based FOSS projects were sourced from SourceForge.net because it is the oldest repository. It was created in the early 2000s. Examined FOSS projects were grouped into eight groups: accounting (11), office (6), project management (9), games and entertainment (9), knowledge management (6), development (5), and scientific and engineering (16) (see **Figure 3.1**). Studied repositories accommodate stakeholders from multiple parts of the world with roles with different academic backgrounds and experiences and play different as stipulated by the core-periphery structure of FOSS (Jiang, Cheng, & Wang, 2017).

Theoretically, FOSS repositories are defined as file archives and hosting facilities for software source code, documentation, web pages, and other works accessible publicly or privately (Falessi & Reichel, 2015). They are frequently used by FOSS projects and other multi-developers' initiatives to preserve the revision and version history of source codes to impose version control (Zolkifli, Ngah, & Deraman, 2018). Generally, repositories contain bug-tracking systems, release management tools, mail lists, and wiki-based project documentation and are often used to retain the authors' copyrights.

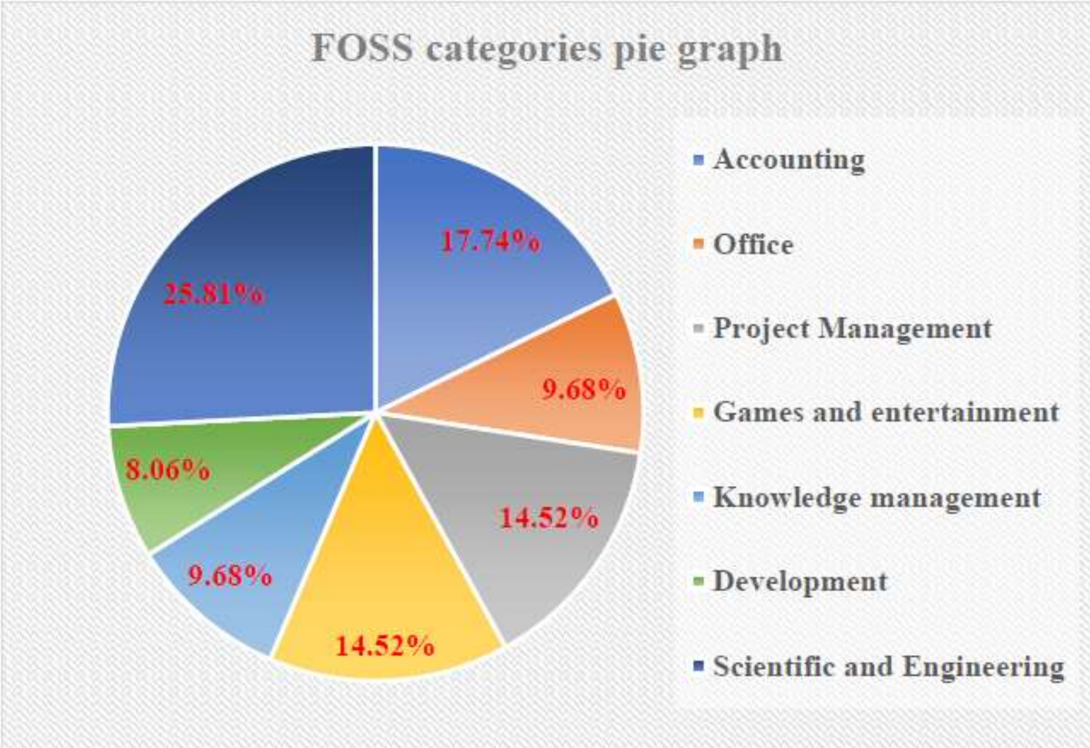

Figure 3.1: Categories of Software Explored

GitHub, SourceForge, GitLab, Bitbucket, Assembla, Phabricator, Launchpad, Beanstalk, GitBucket, Gogs, Gitea, Apache Allura, and Rhode Code are typical examples of the FOSS projects' source code hosting facilities (Güemes-Peña, López-Nozal, Marticorena-Sánchez, & Maudes-Raedo, 2018). Throughout the research, the principal researcher (the author) and other data enumerators were based at UDOM's new computer labs because they had state-of-the-art internet connectivity and other necessary software.

3.3 Research Approach

According to Creswell and Guetterman (2019), research approaches represent research plans and procedures covering everything from general assumptions to detailed data collection, analysis, and interpretation. This study adopted a multilevel triangulation mixed research approach that used qualitative and quantitative techniques. The mixed research approach provides a comprehensive and deep understanding of events that would be impossible if the study relied on qualitative or quantitative methods alone (Creswell & Guetterman, 2019; Gay, Mills, & Airasian, 2006; Kaplan & Maxwell, 2006). It helps reduce biases and ensure correct data judgment (Benítez, Van de Vijver, & Padilla, 2022).

Four primary research approaches exist in mixed research: triangulation, embedded, explanatory, and exploratory; however, triangulation is the standard and well-known technique, particularly among new researchers (Cresswell, Plano-Clark, Gutmann, & Hanson, 2003). The multilevel triangulation design approach has several advantages; for example, it supports formulating research teams, is efficient, and brands intuitive logic. With this approach, different levels of analysis are addressed using qualitative or quantitative methods with equal weights, and the study merges gathered findings to infer one overall interpretation.

3.4 Research Design

According to Kothari (2004), the research design represents a conceptual framework for conducting research. It often epitomizes the techniques involved in the research process: data gathering, data analysis, and report authoring (Creswell & Guetterman, 2019), which allows research operations to be as efficient as possible, yielding the most information with the least effort and time (Kothari, 2004; van der Merwe, Gerber, & Smuts, 2020).

This study has adopted a Design Science Research (DSR) methodology or Constructive Research Design by Hevner et al. (2004) to optimize the design for UX in the development process of desktop-based FOSS projects. According to Iivari and Venable (2009), DSR is a "*research activity that invents or builds new, innovative artifacts for solving problems or achieving improvements, which means. DSR creates new means for achieving general (unsituated) goals as its major research contributions. Such new and innovative artifacts create a new reality, rather than explaining existing reality or helping make sense of it."* DSR helped identify problems, specify research objectives, design and develop the artifact, demonstrate and evaluate its performance, and communicate results to other stakeholders. The DSR was chosen for its capacity to bring practical value by creating valuable artifacts and its scientific rigor in defining design theories (Baskerville, Baiyere, Gergor, Hevner, & Rossi, 2018).

DSR is an appropriate research technique when researchers must collaborate closely with organizations to test novel ideas in a real-world setting (Dresch, Lacerda, & Antunes, 2015) because it provides the ability to create and appraise an artifact with qualitative or quantitative methods. Multiple researchers regard DSR as the study paradigm that guides and dictates verdicts on selected research methodology. It comprises six key activities or steps (Lapão, da Silva, & Gregório, 2017; Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007; Teixeira et al., 2017). (1) Problem identification and motives. (2) Defining objectives of a solution (3) Designing and developing artifacts (constructs, models, and methods) (4) Explaining how to address the problem using produced artifacts. (5) assesses the solution by comparing objectives and actual observed results using developed artifacts. (6) Inform other researchers and practitioners about the problem, artifact, value, and efficacy (see **Figure 3.2**).

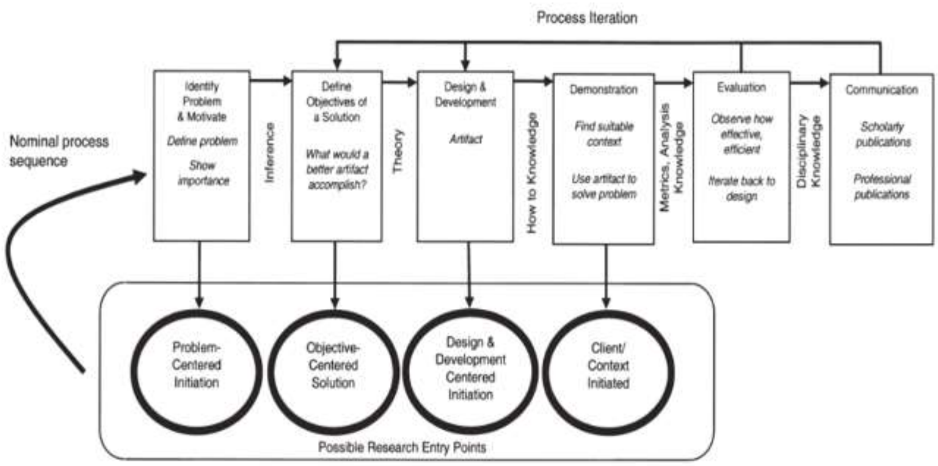

Figure 3.2: DSR Methodology Processes

SOURCE: Peffers et al. (2007)

Although the steps in the DSR methodology have numbers, researchers do not always have to begin with the first step; instead, they usually work their way outward from the research's starting point (Peffers et al., 2007). As an outcome, DSR always produces a valuable artifact that may be a product, a process, a technology, a tool, a methodology, a technique, or a procedure. It may also be a combination of process, technology, tool, methodology, technique, or procedure (Venable & Baskerville, 2012).

Figure 3.3 indicates the flow chart of the iterative execution of research tasks to answer the study's objectives and research questions.

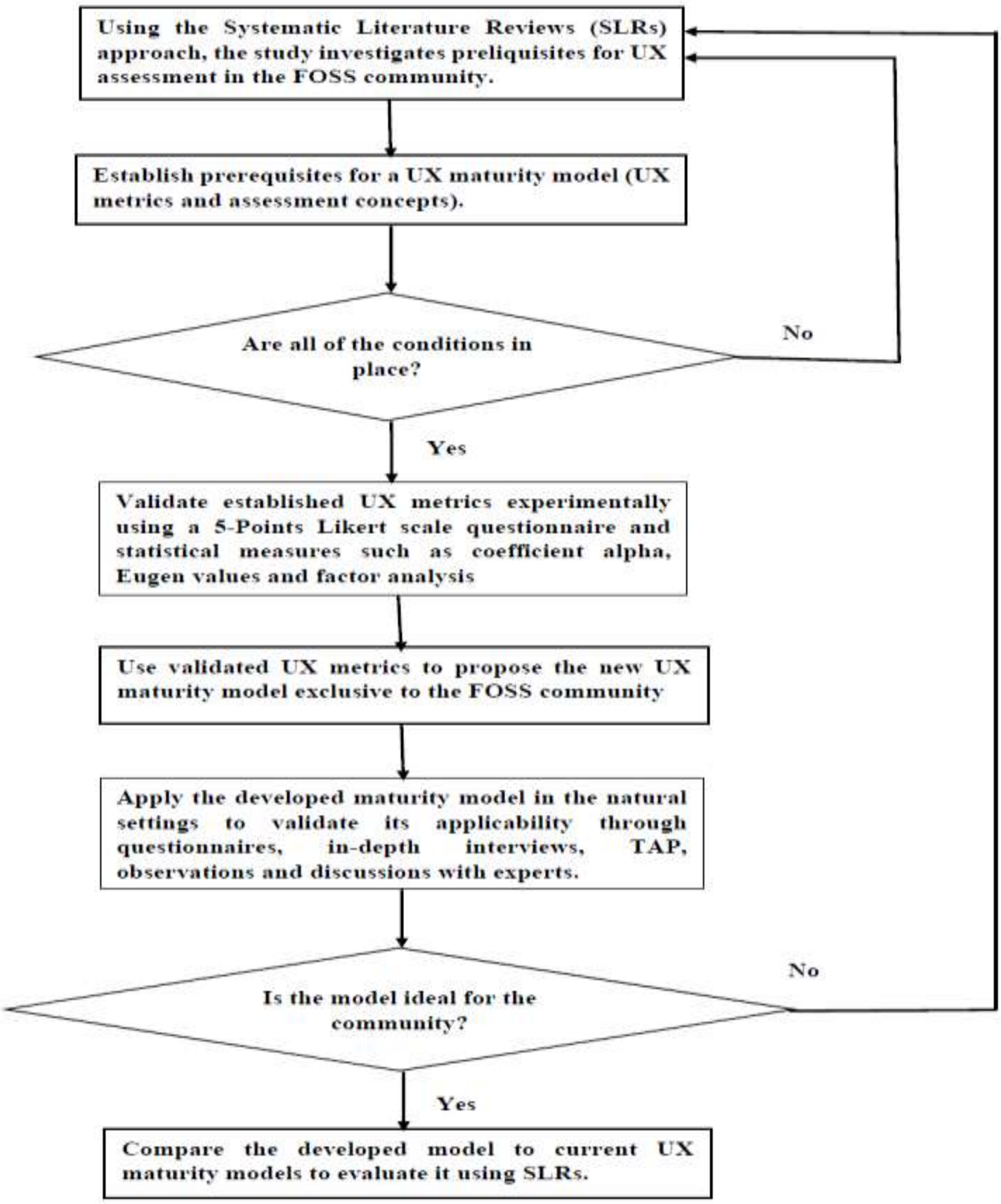

Figure 3.3: Research Tasks

3.5 Study Settings

A study setting refers to the physical or virtual environment in which a study is conducted. It includes the location, the physical and social context, and the conditions under which the study is completed. The study setting is essential to research design, as it can influence the results. For example, if an investigation is conducted in a laboratory, the controlled environment may limit the generalizability of the results to real-world settings. On the other hand, if a study is conducted in a naturalistic setting, the lack of control over variables may limit the study's internal validity.

This study adopted virtual settings that used online environments to study phenomena in virtual worlds, social networks, or other digital platforms. The author designed the settings to accommodate various activities sequentially and, in some cases, concurrently. These activities happened in six stages. (i) the Systematic Literature Review (SLR); (ii) interviews with UX scholars and stakeholders; (iii) seeking UX scholars' and other stakeholders' opinions, (iv) proposing the model; (v) validating the model and (vi) comparing the developed model with other UXCMMs.

3.5.1 The Systematic Literature Review (SLR)

Usually, the SLR aids in determining the variables under investigation's current state of the art (Namayala et al., 2022). The study accomplished the SLR to uncover first the preliminary list of analytical lenses for analyzing the readiness of desktop-based FOSS projects for UX maturity. The author argued that assessing the UX maturity of FOSS projects will not make a difference if the FOSS community is not prepared for higher levels of UX maturity. Second, the study completed the SLR to identify the initial list of UX maturity influencing factors exclusive to the FOSS community. The initial list of lenses and UX maturity influencing factors were provided to experts for more insights, and collected opinions were analyzed and ranked using the fuzzy Delphi method (FDM).

While performing the SLR, the study adapted procedures proposed by Kitchenham and Charters (2007), Budgen and Brereton (2006), and Kitchenham et al. (2010), which assisted in developing a contextualized SLR protocol plan. The SLR may bring several advantages to the investigation when appropriately conducted. For example, it may produce unbiased outcomes (Ali et al., 2017; Kitchenham & Charters, 2007).

3.5.2 Interviews with UX Scholars and Other stakeholders

Interviews help researchers learn more about respondents' viewpoints on a given concept, initiative, or circumstance (Boyce & Neale, 2006). Interviews as a data collection method are widespread in qualitative research (Peters & Halcomb, 2015), particularly when examining people's viewpoints, experiences, arguments, and explanations (Philipps & Mrowczynski, 2021). They often provide more information than traditional data collection techniques (Boyce & Neale, 2006).

This study interviewed several UX scholars and experts with the help of internet technologies, such as ZOOM and Skype, to dig deeper into lenses for determining the readiness of the FOSS community to assess its UX maturity and UX maturity influencing factors.

3.5.3 Seeking FOSS Experts' and Other Stakeholders' Opinions

The study first sought stakeholders' perceptions regarding the readiness of the FOSS community in assessing UX maturity, which was preceded by identifying and ranking lenses. In its second phase, the study collected opinions regarding factors influencing UX maturity and how they could be organized.

The study adopted a 5-point Likert scale ranging from "Strongly disagree" (1) to "Strongly agree" (5) to validate analytical lenses for identifying the preparation of desktop-based FOSS projects for UX maturity in real FOSS projects. Nevertheless, it adopted the ranking scale questionnaire ranging from Extremely Unimportant (1) to Extremely Important (7) to collect experts' opinions on analytical lenses and the UX maturity influencing factors exclusive for the FOSS community. Before adopting the ranking scale, the study adopted separate questionnaires that helped add/remove lenses and UX maturity influencing factors relevant to the community by seeking experts' opinions. Following analysis, the study concluded whether the identified analytical lenses applied to the FOSS community and whether the experts agreed with the suggested UX maturity influencing factors.

3.5.4 Proposing a New UX Maturity Model

The detailed explanation of how the UXCMM was developed will be explained in a separate section. However, in a nutshell, the study used the UX maturity influencing metrics found significant by being agreed among experts to propose FOSS-UXMM that may adequately address the community’s dynamics. The study adapted the commonly used scales and structures and triangulated DSR by Hevner et al.(2004), the guidelines by de Bruin et al. (2005), and Becker et al. (2010) to develop this model.

3.5.5 Validating the Proposed Model

Two case studies were completed with real projects and stakeholders in natural settings to validate the model and justify its applicability. In each case study, developers or project owners were requested to share their agreement to statements in a 5-point Likert scale questionnaire whose numbers varied between maturity levels.

3.5.6 Comparing the Proposed Model with Current Models

Finally, the study proposed a framework to compare the model with those already used to assess its superiority. However, comparing existing and proposed models was intricate due to the lack of standard vocabulary. This discovery confirms the findings of Monteiro and Maciel (2020). However, studies may still use lenses referred to as "generic" and "applicable" to any comparative analysis of information systems (Abrahamsson, Warsta, Siponen, & Ronkainen, 2003a; Avison & Fitzgerald, 1995). Several other studies, including Namayala et al. (2022), Lacerda and von Wangenheim (2018), and Jokela et al. (2006), have used these lenses in analyzing IS.

Due to the overwhelming number of available UXCMMs, the study has just explained how the proposed model has been improved using multiple analytical lenses, including 'Identity,' 'Approach,' 'Inputs analysis,' 'Practical applicability,' 'Empirical evidence,' 'Other organizational viewpoints,' 'Scope,' and 'Concrete guidance.’

3.6 Study Population

According to Banerjee and Chaudhury (2010), Casteel and Bridier (2021), and Umair (2018), a population is a collection of people or other things, such as objects, events, organizations, countries, species, and organisms, where the study seeks to obtain specific information. The provided explanation implies that the population does not exclusively refer to humans but instead explains the parameters that enable the researcher to explicitly define subpopulations within the research, such as the target population, sampling frame, and sample (Lavrakas, 2013; Salkind, 2010; Van den Broeck, Sandøy, & Brestoff, 2013). Generally, the population is the sum of all the things, subjects, or members that meet requirements. However, it often symbolizes the people to whom the findings apply and has significant outcomes (Casteel & Bridier, 2021).

The populations for this study included desktop-based FOSS projects, experts, end users, developers, project owners, and other FOSS stakeholders of different races, ages, educational levels, socioeconomic positions, and residences.

3.7 Sampling Techniques and the Sample Sizes

3.7.1 Sampling Technique

It is often expensive and infeasible to research the entire population. As a result, standard practices insist that researchers must use sampling techniques to select the true representative of the population for further analysis and infer the findings back to the population (Bhalerao & Kadam, 2010).

This study used the purposive sampling technique to select sixty-two (62) FOSS projects from five (5) repositories. The inclusions/exclusions criteria of selected projects included those with an activity level of 90% and above, weekly downloads count of 100 and above, an active mailing list and other discussion forums, and regular updates. The study calculated activity levels from user ratings. For example, a user rating of 4.9 out of 5 has an activity level of 98%.

As an initiative to randomly identify FOSS stakeholders and other UX experts as respondents to the investigation, the study also adopted the purposive sampling technique. However, FOSS stakeholders and UX experts had to qualify for the inclusion/exclusion criteria. The inclusion/exclusion criteria for FOSS stakeholders included those who were active and regularly playing one or multiple roles identified by the core-periphery of the FOSS community, demonstrating a willingness to participate in the study without reimbursement, and having an extensive ability to accomplish assigned roles.

The inclusion/exclusion criteria of UX experts included those who demonstrated extraordinary academic ability on UX by either publishing peer-reviewed articles or having significant knowledge contributions to the UX knowledge body and willingness to participate in the investigation free of charge. The author collected stakeholders' e-mails from discussion forums and sent personalized e-mails explaining research objectives and requesting their participation consent. E-mails of experts were primarily extracted from published articles, including journal papers, conference proceedings, books, and book chapters. Nevertheless, other experts were selected based on snowballing.

According to Andrade (2021), a purposive sample is one whose characteristics are defined for a specific purpose related to the investigation. The research community commonly categorized purposive sampling as the non-probabilistic sampling technique and was selected in this study because of its effectiveness when randomization is impossible. For example, when the population is massive, with limited resources, time, and labor, the research does not seek to provide conclusions that infer the entire population (Andrade, 2021; Etikan, Musa, & Alkassim, 2016).

3.7.2 The Sample Size

According to Newman (2003), the sample size is the number of individuals or observations included in a study. However, for decades, determining sample size has been a research topic in usability engineering (Lewis, 1982). For example, Nielsen (2000) recommended that using a small sample size of five (5) in usability and UX research can discover over 85% of usability and UX-related issues. Nielsen (2000) came to this conclusion after inserting the average probability ($p = 0.31$) of detecting UX and usability issues in a single session into Virzi's (1992) mathematical model.

Several other studies disagree with Nielsen's (2000) specified sample size, claiming it does not apply to all UX study circumstances. Some UX and usability study circumstances require a larger sample size to achieve an 80% detection rate in complex and sophisticated systems (Spool & Schroeder, 2001). Adams (2009) further found that the eight-person sample size is insufficient. On the other hand, Sauro and Lewis (2016) pointed out many other flaws in Nielsen's (2000) proposal and projected alternative practical methods for determining the ideal sample size for UX research.

Nevertheless, as the contributions to the “five users are (not) enough” debate, Hwang and Salvendy (2010) have proposed sample sizes of 10 ± 2 using a binomial model. Lewis (2001) presented a binomial model with a correction term or the Good-Turing (GT) adjustment to account for undiscovered UX and usability issues, which required 16 participants. Finally, Schemattow (2009) presented a mathematical model dubbed the "zero-truncated logit-normal binomial distribution," or $LNB_{zt,}$ to account for variance, problem visibility, and unobserved events. It proposes a sample size of 56 and confirms that visibility variances are a fact and that strong incompleteness occurs when the sample size is smaller than 30 (Schmettow, 2012).

Schmettow's (2012) study confirmed that LNB_{zt} was not the last model in estimating the sample size in usability and UX research. There are several other efforts to estimate the sample size. For example, as illustrated in **Figure 3.4**, Kelvar (2021) from UserZoom, the recent study has recommended ideal sample sizes for three UX research scenarios based on significance levels: identifying UX issues, determining UX metrics, and comparing options.

Identify Usability Issues

Problem / Insight Occurrence	Sample Size Needed
40%	4
30%	5
20%	9
10%	18
5%	37

Estimating Parameters KPI

Margin of Error (+/-)	Sample Size Needed 90% Confidence
24%	10
15%	28
10%	65
8%	103
5%	268
3%	749
2%	1,689

Comparing Options

Difference to Detect (90% Confidence)	Sample Size Within Subjects	Sample Size Between Subjects
50%	17	22
30%	29	64
12%	93	426
10%	115	614
5%	246	2,468
3%	421	6,866
1%	1,297	61,822

Figure 3.4: Proposed Sample Sizes for UX-Related Research

SOURCE: Kelvar (2021)

On the other hand, best research practices require a sample representing the population to be adequate to infer findings from the population (Singh, 2011). As a result, suggestions to use standard formulas, including those proposed by Rose et al. (2014), to calculate an ideal sample size from the unknown population with the desired precision continue to erupt. The Rose et al.(2014) formula says

$$n_r = \frac{4pq}{d^2}$$ Were

n_r is a required sample size, p is the population proportion, the researcher obtains its value from previous research, $q = 1 - p$ and d is the precision. The 95% confidence interval assumes values between 5% and 8%. If the population (p) proportion is unknown, it carries the weight of 0.5 for maximum heterogeneity. Therefore, by putting these values into Rose et al.'s (2014) formula, we get $n_r = \frac{4*0.5*0.5}{0.08^2} = \frac{1}{0.064}$ = **156**. However, using Rose et al.'s (2014) formula to calculate sample size in usability and UX studies is not a regular practice. According to Schmettow (2012), UX and usability problems are likely to be excessively complicated and varied, and building user interfaces with seeded UX and usability problems necessitates significant development time and money.

The sample size for this study comprised one hundred seventy-one (171) respondents. With this sample size, the study complied with Kelvar's (2021) proposition at the 90% Confidence Interval and the requirements of Schemattow (2009), Nielsen (2000), and Lewis (2001). Looking into other studies, the author found that Raza et al.(2012a) did not calculate sample sizes using conventional formulas, such as Yamane (1967) when they developed the OS-UMM. They asked 102 respondents to collect users' perceptions (Raza et al., 2012b) and asked 105 respondents to realize how industries perceived usability influencing factors (Raza et al., 2013)

When examining the FOSS community's willingness to evaluate its UX maturity, this study first contacted eleven (11) UX experts to identify and rank the analytical lenses for determining the readiness of the FOSS community to assess its UX maturity using the Fuzzy Delphi Method. Using a Likert scale questionnaire, it reached seventy (70) FOSS stakeholders to validate the analytical lenses' relevance in the FOSS community. In addition to this, fifteen (15) independent experts were recruited to verify all involved tools before deploying the actual study.

When determining and ranking the UX maturity influencing factors exclusive for the FOSS community using the Fuzzy Delphi method, the study consulted twelve (12) experts. All tools involved in the study were piloted with five (5) independent reviewers. Finally, the study contacted twenty-five (25) project owners or developers while validating the proposed UX maturity model. Eighteen (18) users pretested tools for validating the FOSS-UXMM (see **Table 3.1**).

Table 3.1: Sample Sizes Specified on Completed Tasks

No.	Activity	Type of data collected	Sample size
1	Gaining more profound insights into the factors affecting UX maturity in the FOSS community	Qualitative	15
2	Identify and rank lenses for determining the preparation of the FOSS projects for UX maturity.	Qualitative and quantitative	11
3	Investigate and understand how FOSS stakeholders view metrics that impact its readiness to assess its UX maturity	quantitative	70
4	Pilot tools for investigating the readiness of the FOSS community to assess its UX maturity.	Qualitative and quantitative	15
5	Identify and rank UX maturity influencing factors exclusive to the FOSS community	Qualitative and quantitative	12
6	Pre-test tools for identifying and ranking UX maturity influencing factors exclusive for the FOSS community	Qualitative and quantitative	5
7	Validate the FOSS-UXMM	quantitative	25
8	Pilot tools for validating FOSS-UXMM	quantitative	18
9	**Total Sample size**		**171**

3.8 Data Collection Methods

Basavanthappa (2007) defines data collection as the systematic process of acquiring and measuring information on variables of interest to answer specified research questions, test hypotheses, and evaluate outcomes. The current corpus of research confirms the existence of multiple data-gathering methodologies that differ depending on the domain in question and the types of data collected. Educational researchers' most popular methods are questionnaires, interviews, focus groups, observation, and documentary reviews (Robert & Larry, 2019).

This study adopted documentary reviews, observations, questionnaires, and in-depth interviews when collecting data. Using multiple data-gathering techniques (triangulation) and multiple sources improves the trustworthiness of results and allows the inclusion of different interpretations and meanings in data analysis (Robert & Larry, 2019; Shanks & Bekmamedova, 2018).

3.8.1 The Documentary Reviews

The documentary review is the systematic collection, documentation, analysis, interpretation, and arrangement of data (Bretschneider, Cirilli, Jones, Lynch, & Wilson, 2017). According to Bretschneider et al.(2017) and O'Leary (2017), physical evidence or artifacts, such as flyers, posters, agendas, handbooks and training materials, personal documents, such as calendars, e-mails, scrapbooks, blogs, Facebook posts, duty logs, incident reports, reflections/journals, and newspapers, and public records, such as student transcripts, mission statements, annual reports, policy manuals, student handbooks, strategic plans, and syllabi, are the three types of documents reviewed in the documentary reviews.

The documentary review is a crucial social study that has possibly been the most commonly employed throughout the history of sociology and other social sciences. For leading sociologists, it has been the primary and, in some cases, the only method. Furthermore, it establishes a method for detecting, analyzing, and extracting usable data from existing documents.

In this study, the documentary review started with reviewing critical components of UX identified by Thüring and Mahlke (2007), which include usability, emotions, and aesthetic impression or appearance. The components are further decomposed into sub-constructs that describe the nature of the critical aspects of UX with quantifiable characteristics. The study then gathered background information on UX assessment, metrics, and the usage of CMM and its derivatives in evaluating UX-related processes. The collected findings were later used to trigger brainstorming sessions when the Fuzzy Delphi method was executed.

3.8.2 Observations

According to Ekka (2021), observation gathers information by observing behavior, events, or physical characteristics in their natural setting. The Observation supports qualitative and quantitative approaches (Girard & Cohn, 2016) and has various advantages. For example, it is a universally accepted and followed practice because it allows researchers to watch what people do and does not rely on people's willingness or ability to offer required information. Observation can be overt, in the sense that everyone is aware observers watch them, or covert, where no one knows observers watch them.

The study followed ten steps to create the observation checklist (see **Appendix XI**). The steps comprise considering research questions, identifying the key variables, determining the data collection method, developing the checklist items, organizing the items, pilot testing the checklist, training observers, using the checklist, analyzing data, and reviewing the checklist. The formed checklist assisted in understanding how stakeholders contributed to mailing lists of the selected projects and helped comprehend stakeholders' readiness to participate in the study and the project's activity levels. It further guided in knowing stakeholders' behavior toward adopting UX assessment techniques in the FOSS community and identifying projects with many weekly download counts. The study implemented the checklist between mid-July 2021 and November 2022 to allow proper analysis.

3.8.3 Questionnaires

According to Bhandari (2022) and WHO (2001), a questionnaire is a set of questions or items used to collect data that may be qualitative or quantitative regarding respondents' views, experiences, or opinions. It is ideal for quickly stashing a large amount of data from many responders (Fletcher, 2015; Rowley, 2014). The success of a questionnaire lies in the proper setup of the questions. When adequately developed and administered, it is essential for making statements about specific groups, individuals, or entire populations (Roopa & Rani, 2012).

As an initiative to administer selected samples in the study, various questionnaires were employed. The study first used an interview guide (se**e Appendix I**) to interview FOSS stakeholders and other experts to understand better the factors influencing the UX maturity-impacting elements discovered in the literature review. The study sent e-mails to stakeholders and experts (see **Appendix II)** to introduce the fuzzy Delphi method for identifying and raking analytical lenses to examine the readiness of the FOSS projects for maturity. It used a questionnaire (see **Appendix III)** to establish possible lenses for determining its readiness for UX maturity. The study then adopted the questionnaire (see **Appendix IV**) to develop the agreement levels between experts on the final lenses to evaluate the preparation of FOSS projects for UX maturity (see **Appendix V**). Nevertheless, the study sent e-mails (see **Appendix VII)** to introduce the fuzzy Delphi method for determining factors influencing UX maturity in the FOSS community. It used questionnaires (see **Appendix VIII**, **Appendix IX**, and Appendix X) to establish possible factors, establish experts' opinions, and rank the factors influencing the UX maturity of the desktop-based FOSS projects.

Several used questionnaires are claimed to meet the validity and reliability because they were derived using triangulation methods, which involved empirical extraction of variables from commonly used UX assessment questionnaires. These questionnaires include SUS by Brooke (1996), VisAWI by Moshagen and Thielsch (2010), AttrakDiff by Hassenzahl et al.(2003), SUMI by Kirakowski and Corbett (1993), meCUE by Minge et al.(2017), UEQ by Hinderks (2015), PANAS by Watson et al.(1988), QUIS by Wallace et al.(1988) and PANAS-X by Watson and Clark (1994). The triangulation methods could help to avoid several limitations and create UX maturity metrics that can help in the smooth assessment of the organization's UX maturity. They may also offer a unique empirical study of UX maturity metrics that may provide better and more consistent methods.

The study ensured that the designed questionnaires were highly credible by taking different initiatives. First, it adapted questionnaires short enough to attract many respondents. Second, the analysis was performed on all developed questionnaires to ensure they only included variables correlated to the scales and represented valid constructs. The study used different approaches to identifying components' significance, including a screen plot test, the interpretability criterion, the proportion of total variance accounted, and the Eugen score one criterion. Nevertheless, each adopted questionnaire had distinct usage objectives, benefits, and drawbacks (Santoso et al., 2016) (see **Table 3.2**).

Table 3.2: Key UX Components and Measuring Questionnaires

UX Component	**Measuring Questionnaires**
Usability	QUIS by the Human-Computer Interaction Laboratory (HCIL), University of Maryland, PSQ by Davis (1989), ASQ by Lewis (1991), CSUQ by James Lewis at IBM, PSSUQ by IBM, SUMI by Kirakowski, SUS by Brooke, PUTQ by Lin et al. (1997), WAMI by Kirakowski, USE by Lund (2001), AttrakDiff by Hassensahl et al. (2003), AttrakDiff 2, UEQ, ISQ, UMUX, DEEP, UMUX LITE, AltUsability, SUPR-Q, SUPR-Qm, UEQ-S), and UEQ+
Emotion	PANAS, SUMI by Kirakowski, PANAS-X), USE by Lund (2001), AttrakDiff by Hassensahl et al. (2003), AttrakDiff 2, Emotions in Consumer Behavior, UEQ, UMUX), EMO-CHeQ), SUPR-Q, SUPR-Qm, UEQ-S, meCUE, UEQ+
Appearance	WAMI, AttrakDiff 2, UEQ, VisAWI, DEEP, VisAWI-S, SUPR-Q, SUPR-Qm, UEQ+

SOURCE: Apraiz Iriarte and Ganix (2020)

3.8.4 In-depth interviews

As an initiative to gather more details on UX maturity metrics identified in the reviewed literature, the study adopted the interview guide to conduct in-depth interviews with UX specialists, project owners, managers, developers, experts, and other practitioners in the FOSS community. By thoroughly examining fewer participants than other data collection techniques (Boyce & Neale, 2006), in-depth interviews may better understand UX maturity-related metrics and their evaluation techniques. It may also provide more data than traditional data collection methods (Jacobvitz, Curran, & Moller, 2002).

The 15 key informants were involved in the completed in-depth interview, which included five (5) project owners and managers, five (5) developers, and five (5) experts. The author of this work hypothesized that these participants could have access to important information by working directly with FOSS projects. Nonetheless, all interview sessions lasted between 30 and 60 minutes and were audio-taped for future reference and data extraction.

3.9 Data Processing and Analysis

Data in their raw form are useless; they require a conversion process to turn into valuable information. Turning study data into helpful information is called data processing and analysis. Data processing structure includes data organization, strategies for reducing data, and analysis techniques. LeCompte and Schensul (1999) define data processing and analysis as reducing massive data into meaningful data, leading to reasonable interpretation. In many cases, the nature of the data collected during the investigation determines the data processing and analysis methods used. Research questions or hypotheses, on the other hand, can also influence the decision. For example, practitioners often use Content, Narrative, Discourse, Thematic, and Grounded Theory to analyze qualitative data, and Crosstabs, Trends, MaxDiff, Conjoint, TURF, Gap, SWOT, and Text to analyze quantitative data.

3.9.1 Data Capturing

The study captured massive data electronically by utilizing an online survey powered by Google Forms to ensure the efficiency and accuracy of the collected data. It also manually captured other data that required enumerators to key them to the central database for analysis. According to Barchard et al.(2020), Data entry errors are unavoidable when captured manually. Meanwhile, simple data-entry mistakes, such as entering a wrong number or skipping a line, can significantly impact a study's outcomes (Barchard & Pace, 2008, 2011; Hoaglin & Velleman, 1995; Kruskal, 1960; Wilcox, 1998). For example, they can change the direction of a correlation or turn a significant t-test into a non-significant one (Barchard & Verenikina, 2013).

Several methods, such as histograms and scatter plots, are commonly used to validate data entry errors (Abzalov, 2016). Others include determining univariate and multivariate statistics to find outliers and other influential data points (Osborne & Overbay, 2004; Tabachnick & Fidell, 2019). However, best practices emphasize item-by-item data inspection, which supports single or double-data entry to locate all problems (Barchard et al., 2020).

This study, therefore, opted for a double-entry data collection method to ensure the manually captured data were accurate and valid. Double-entry allows the data-entry individuals to correct errors immediately (Scott, Thompson, Wright-thomas, Xu, & Barchard, 2008) because it accommodates two enumerators entering data independently in two independent files with subsequent or simultaneous data comparison and compilation of a final dataset.

As a result, there were no substantial errors in data entry between the lead investigator and other data enumerators, according to the study.

3.9.2 Analyzing Data

The study adopted content analysis to analyze qualitative data. The content analysis has six processes: familiarization, coding, creating, assessing, defining, identifying themes, and writing up (Caulfield, 2020), and it usually gives codes to the transcripts depending on the research objective. It permits verbal and visual communication and textual data analysis (Guyan, 2018; Roberts, 2015). In addition, the study utilized quotations to explain data acquired through interviews, secondary data, and observation methods. Regarding the analysis software, NVivo was employed to summarize, analyze, and group qualitative data into several themes to acquire a deeper understanding and profound knowledge of the problem.

Moreover, IBM SPSS Software and Excel were employed to analyze quantitative data. The IBM SPSS software was selected due to its flexibility, scalability, and convenience for projects of different sizes and complications (Meulman & Heiser, 2013). In contrast, Excel was adopted for its suitability in fuzzification and defuzzification of experts' opinions in the Fuzzy Delphi Method. As the data analysis curve, the study performed a descriptive statistical analysis, such as central tendency and dispersion measures, and presented the findings using numbers and percentages in tabular forms, charts, and graphs. Finally, the study used a 95% Confidence Interval to perform statistical tests to establish the statistical significance of constructs.

3.10 Ethical Considerations

The Department of Computer Science and Engineering (CSE) at the University of Dodoma's College of Informatics and Virtual Education (CIVE) provides numerous Doctor of Philosophy (PhD) degrees, including Computer Science. This thesis fulfills the requirements of the PhD in Computer Science program. After extensive scrutiny at the department and college level, the researcher, CSE, and CIVE management agreed on the study's title, problem statement, and objectives. Subsequently, the researcher followed laid procedures to secure authorization to complete the survey from the Director of Postgraduate Studies at the University of Dodoma and the permanent secretary of the President's Office, Regional Administration, and Local Government (PO, RALG).

The author collected stakeholders' e-mails from discussion forums and sent personalized e-mails explaining research objectives, requesting participants' consent, and providing a brief presentation on the importance of the research. He observed the importance of anonymity, privacy, and confidentiality, adhered to several data protection acts, and guaranteed the utilization of obtained data for only achieving the study's objectives. The personal safety of participants was also ensured by highlighting necessary precautions and critical issues that must be observed. Nevertheless, all participants were free from coercion and could voluntarily participate and withdraw at any stage and time when they felt like doing so.

Finally, the study has observed proper citation, copyright conditions, and mandatory permission protocols to ensure compliance with research ethics and avoid plagiarism. Regarding research data security, the study used passwords and end-to-end encryptions to secure communications and electronic documents. In some cases, physical security was implemented to secure research documents.

3.11 Study Reliability and Validity

Reliability and validity are not interchangeable. Mohajan's (2017) study shows that validity aims to enhance the transparency and accuracy of the data collected, while Goffman's (1992) study indicates that reliability refers to the state where the study instruments can yield similar results over time. Generally, reliability and validity include credibility, dependability, confirmability, and transferability. The methodologies used in research aimed at enhancing the reliability and validity of the completed investigation.

For example, the study adopted triangulation of data collection tools comprised of questionnaires, field observations, document reviews, and in-depth interviews. It also involved multiple investigators, including a lead investigator, co-authors in published papers, and two other postgraduate students employed as data enumerators. Finally, its design adopted a mixed methodology that included qualitative and quantitative data. According to Moon (2019), Saunders et al.(2019), and Campbell and Fiske (1959), triangulation is one way of improving the validity, reliability, and legitimacy of research findings.

Moreover, the actual research was preceded by a pilot study and only included active FOSS projects from five repositories with 90% and above activity levels. Finally, the study only contacted active, trusted, and knowledgeable stakeholders with reliable contacts using e-mails, forums, portals, Internet Relay Chats, Go To Meetings, and ZOOM collaboration software.

3.12 Development of FOSS-UXMM and Validation

According to the exhausted literature review, adopting generic UXCMMs to assess firms has been standard practice. This conclusion is confirmed by the findings from Namayala et al. (2022) and Lacerda et al. (2018). However, the organizational dynamics vary, leading to organizations demonstrating different behaviors and characteristics that may inadequately be accounted for by generic UXCMMs. The exhausted literature also confirms that generically developed UXCMMs lack large-scale deployments across multiple contexts and domains, making their applications debatable (Lacerda & von Wangenheim, 2018).

This study has attempted to overcome the challenges of generically developed UXCMMs by adhering to the recommendations of other studies, such as Ali et al. (2018), that insist on developing new domain-specific models using alternative techniques, such as fuzzy mathematics. It has experimentally created essential prerequisites for creating UXCMM that may account for the FOSS community dynamics. While establishing these fundamentals, the study has tried implementing

UCD practices and involving every FOSS-known stakeholder highlighted by the core-periphery structure of the FOSS community. Although there are several ways of explaining the core-periphery structure, the onion model (see **Figure 3.5**) dominates. Detailed explanations of the onion model are provided by the studies of Amrit and Van Hillegersberg (2010), Christian and Vu (2021), and Mockus, Fielding, and Herbsleb (2002).

Figure 3.5: The Onion Model of the FOSS Community
SOURCE: Amrit & Van Hillegersberg (2010)

The contemporary body of knowledge presents no universally accepted guideline for creating maturity models (Becker et al., 2009; de Bruin et al., 2005; Monteiro & Maciel, 2020; Namayala et al., 2022). As a result, this study has proposed a hybrid technique when creating the FOSS-UXMM to overcome the limitations of individual guidelines (see **Figure 3.8**). The hybrid technique results from triangulating approaches that promote the systematic development of UX maturity models. These approaches include the seven procedures of DSR by Hevner et al. (2004), the model development checklist in the FOSS community by Namayala et al. (2022), steps by Peffers (2007), and guidelines by de Bruin et al. (2005), Becker et al.(2010) and Dresch et al.(2015).

In creating a proposed hybrid technique, the study has exhausted a literature review on available guidelines and frameworks for building models. It then combined the identified processes to provide modified procedures suitable for creating FOSS-UXMM. The discussion of general guidelines and models for developing UXCMMs is as follows:-

3.12.1 A Model by Becker et al. (2010)

This model comprises seven phases (see **Figure 3.6**) and provides a helpful framework for developing UXCMMs. However, it has several limitations that need adaptation to suit specific context-related issues of the FOSS community. These limitations include limited guidance for model construction, emphasizing the development of quantitative models, assuming a linear development, lacking theoretical focus, and providing limited guidance on model dissemination.

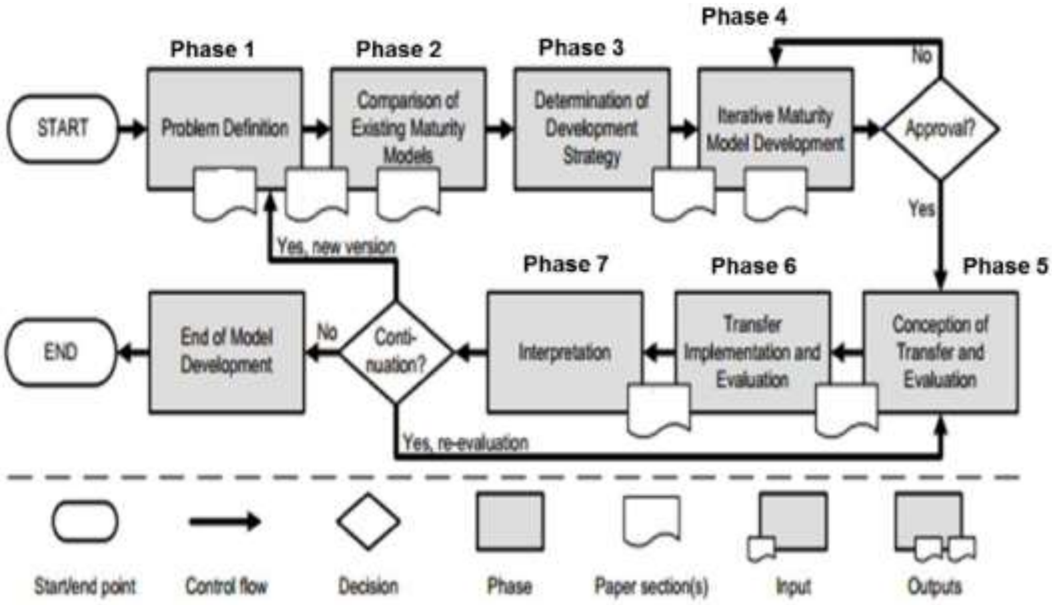

Figure 3.6: **Steps for Models' Development**

SOURCE: Becker et al.(2010)

Phase 1: Problem definition

The problem definition phase expresses the context and associated domain components to determine the relevance of a problem. It helps to find inconsistencies in the current systems or models that need fixing.

Phase 2: Comparing existing maturity models

After reviewing the existing models, this phase contrasts the defined problem with current models. The performed comparison aids in determining the best approach to take when creating a similar model to resolve the problems defined.

Phase 3: Determining the development strategy

A thorough assessment of the existing UXCMMs in the second phase guides the selection of a sound strategy that can used throughout the development cycle. Multiple options for a chosen strategy exist, for example, innovative model development from scratch, upgrading a current model to fit the required application context, fusing elements from multiple models into one, or moving structures and contents from existing models to a new application domain.

Phase 4: Iterative model development

This phase is consistent with DSR's build-and-evaluate cycle phase. It includes iterating the procedures for choosing the design level, the technique or strategy for development, and testing the model and reporting outcomes.

Phase 5: Conception of transfer and evaluation

This phase aligns the guidelines for research as design science communication. It attempts to ascertain the various methods of disseminating the evaluation's findings to the academic and user sectors. Additionally, it describes a well-thought-out selection of the different formats the model's intended communication can take. Nevertheless, if the evaluation incorporates group differentiation, it must include the criteria for directing the results at the various user groups in transmitting the results.

Phase 6: Transfer implementation and evaluation

This phase is comparable to the design evaluation guidelines used in the DSR phase, and it ensures the availability of the model and supporting tool to the audience as described in the conception of the transfer and evaluation phase. In a different language, it evaluates if the model achieves the required goals when applied in the real world. Ideally, the evaluation process entails a thorough demonstration of the model's utility, quality, and efficacy using properly conducted evaluation methods.

Phase 7: Interpretation

At this stage, evaluated results are reviewed and assessed to see if the model achieves the anticipated outcome. The outcomes from the reviewed results suggest that the model needs re-evaluation, update, or rejection.

3.12.2 A Model by de Bruin et al.(2005)

This methodology explains standard and systematic procedures to adhere to when developing UXCMMs, and it contains six crucial steps: Scope, Design, Populate, Test, Deployment, and Maintenance (See **Figure 3.7**). As a requirement, the six phases identified must be implemented in the given order. The decisions made during each phase impact the phase that follows. Nevertheless, some phases, mainly the Populate, Design, and Test, are often iteratively implemented.

Although de Bruin et al.(2005) proposed steps provide a valuable framework for UXCMMs development, particularly in health informatics, it has several challenges that need additional research. These challenges include a lack of emphasis on theory, limited guidance on model evaluation and refinement, requirements specification challenges, conceptual modeling, and complex stakeholder analysis.

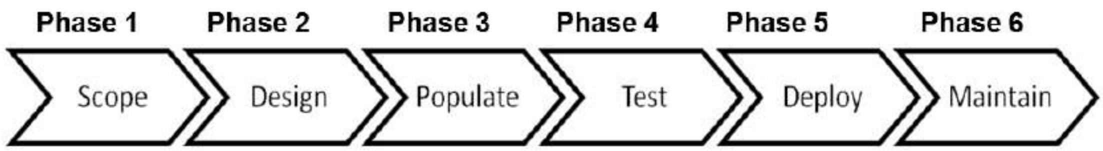

Figure 3.7: Steps for Developing Model

SOURCE: de Bruin et al. (2005)

Phase 1: Scope

The focus of this stage is to explicitly identify the domain of use and goal of the UXCMM, and its role is consistent with DSR's principles 1 and 2. The first principle stresses that DSR can be used to create distinctive artifacts appropriate for a particular use. The second principle emphasizes that the artifact must solve the identified problem. The scoping phase typically identifies the issue that the model aims to tackle. It is preceded by a thorough literature assessment to determine related work and current domain problems. Studying historical and current domain concerns may also yield potential model stakeholders.

Phase 2: Design

The design phase determines the target stakeholder audience for the models and gathers the requirements for the target users. Ideally, the design step should identify several user categories, ascertain their justifications for utilizing the model, establish the conditions under which they may use it, and describe the expected outcomes. From doing so, the phase helps to populate the model's architectural specifications or relevant artifacts while solving the problem by collecting this knowledge, which is the desired quality of DSR.

Phase 3: Populate

The populating phase makes concept diagrams outlining the domain components and requirements necessary for the model to fulfill the needs of the stakeholders and address the stated issue. The components must represent all potential domain constituents for the artifact to be considered rigorous.

Phase 4: Test

The test step assesses how effectively the model complies with the fundamental requirements of typical model characteristics and considers the target audience's needs to address the identified problem. The model should ideally be constructed and reviewed repeatedly until it achieves the specified objectives. In other words, the test phase evaluates the model's internal and external validity to determine its relevance and rigor. Building and evaluating the artifact is required, as suggested by the DSR, because they ensure that it complies with the fundamental theoretical requirements (internal validity) and is acceptable to the intended users regarding its components (requirements, activities, techniques, outputs, and the list continues).

Phase 5: Deploy

The goal of the deployment phase is to make the produced model applicable and helpful in the context of usage, which includes confirming its applicability and generalizability. This stage is critical to ensuring that the model is strategically made available to the appropriate audience, in the right way, and for the proper domain of model usage. This phase is connected to communication when compared to DSR.

Phase 6: Maintain

Due to the change and dynamics of people's requirements and contextual domain problems, the model must maintain two essential core characteristics: extensibility and adaptability. As a result, this stage struggles to keep the model's flexibility to adapt to the changing environment.

3.12.3 Proposed Hybrid Methodology for Developing FOSS-UXMM

Based on the work of de Bruin et al. (2005) and Becker et al. (2010) described above, a hybrid model with six procedures was suggested (see **Figure 3.8**). Its components were chosen following the DSR principles by Hevner et al. (2004).

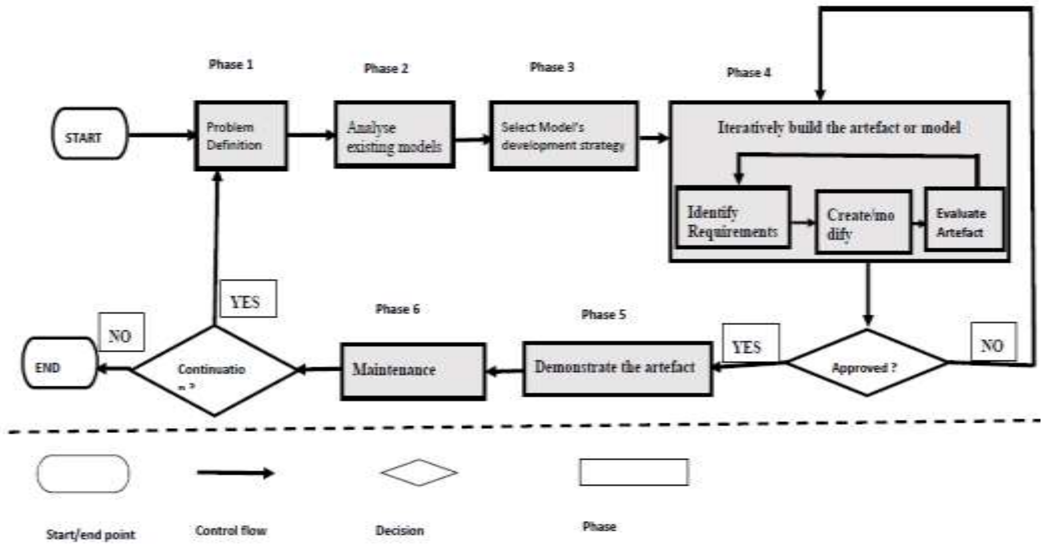

Figure 3.8: Hybrid Methodology for Developing FOSS-UXMM

The steps of the proposed model are explained as follows:-

Phase 1: Problem definition

This phase emphasizes reviewing existing literature on the UX maturity models in the FOSS community to define the problem. The reviewed literature may clarify several issues regarding the UX maturity models, for example, their applicability and whether they measure the right things.

Phase 2: Analyze existing models

After the problem definition, the study used the second phase to examine the reviewed literature to determine the optimal technique to address the problem of managing the UX maturity of the FOSS community.

Phase 3: Select the model's development strategy

At this phase, the study chose to construct FOSS-UXMM from scratch after thoroughly evaluating available possibilities for building models.

Phase 4: Iteratively build the model

At this phase, the study adopted the build-and-evaluate cycle of DSR to develop FOSS-UXMM iteratively. During the development, tasks for establishing requirements, developing the model, and evaluating the outcomes were iterated.

Identify requirements

The study chose the FOSS-UXMM's desired elements, activities, and methods during identifying requirements. It completed literature reviews, observations, and inquiries of subject matters by engaging experts and other FOSS stakeholders.

Develop or modify the model.

The study developed a novel maturity model (FOS-UXMM) based on the collected requirements to meet desktop-based FOSS project-specific needs.

Evaluating results

In this subphase, the study assessed the model's contents and the chosen model structures' rigor and relevance. It performed two case studies to establish how well the model addresses the target audience's demands in resolving the given problem (external validity) and basic requirements of typical model characteristics (internal validity).

Phase 5: Demonstrate the model or the artifact

At this phase, the study explained the model to the intended users in terms and language they can understand to ensure it is strategically available to the right audience in the right way and within the proper domain of use. According to the recommendation of conveying the artifact to management and techno-centric audiences, demonstrating the model comes after the deploying phase (de Bruin et al., 2005).

Phase 6: Maintenance

This phase is final, and it covers the model's practical application to address the situation. It allows wider deployment of the model to several domains and determines its generalizability. It practically documents and updates flaws found while using the model. Unlike other models, the hybrid model for developing UXCMM facilitates the creation of fluid and extensible models adaptable to the agile environment.

3.12.4 Reliability and Validity of FOSS-UXMM

The study adopted the coefficient alpha by Cronbach (1951) to analyze internal consistency when assessing the reliability and validity of UX maturity metrics or influencing factors. To understand what coefficient alpha value was satisfactory to retain a variable (UX maturity metric), the study has adopted the suggestions of two studies, Nunnally and Bernstein (1994) and Taber (2018), claiming a reliability coefficient of 0.70 or above is considered satisfactory. Nevertheless, other researchers have suggested somewhat lower values. For example, Oosterhof (2001) contended that a reliability coefficient of 0.60 or higher is acceptable, and Price et al. (1981) claim that a reliability value of 0.55 or higher is sufficient.

Moreover, as Campbell and Fiske (1959) recommended, the study adopted Principal Component Analysis (PCA) to test and validate the direction of correlation to the UX maturity metrics. According to Jolliffe and Cadima (2016), PCA is a descriptive method for reducing unnecessary dimensions from large datasets while maintaining interpretability and minimizing information loss. This operation is possible by generating new uncorrelated variables that optimize variance successively. The PCA often adopts Eigenvalue as the reference point (Kaiser, 1970), mainly the Kaiser criterion or Eigenvalue one-criterion well explained by Kaiser (1960) and Stevens (2012), and retained UX metrics possessing an Eigenvalue threshold of greater than one.

The study also adopted the normal distribution tests and parametric and non-parametric approaches, including Pearson correlation coefficient (p-value), Spearman correlation, and Partial Least Square (PLS), to deal with hypotheses testing. According to recommendations from multiple studies, such as Fornell and Bookstein

(1982) and Jöreskog and Wold (1982), the PLS approach helps deal with research concerns such as complexity, non-normal distribution, inadequate theoretical information, and problems stemming from a small sample size.

The study performed the inter-rater agreement and inter-rater reliability analysis using Kendall's coefficient of concordance (W) (Kendall & Smith, 1939; von Eye & Mun, 2014) to provide the extent of agreement among experts. The inter-rater agreement and inter-rater reliability analyses resulted from the likelihood that respondents from the same project and organization may have different views. Nevertheless, the study has adopted Cohen's kappa coefficient (κ) (Cohen, 1960) and Fleiss' kappa coefficient (Fleiss, 1971; Fleiss, Levin, & Paik, 2003). Kendall's, Cohen's, and Fleiss's kappa coefficients are all assumed to have values between 0 and 1, with 0 indicating no agreement and 1 indicating perfect agreement (see **Table 3.3)**.

Table 3.3: Kappa Benchmarked Values

Kappa level	Benchmarked value	Descriptions
1	$k < 0.44$	Poor
2	$0.44 \leq k \leq 0.62$	Moderate
3	$0.62 < k \leq 0.78$	Substantial
4	$k > 0.78$	Excellent

Finally, the study tested the model's applicability in natural settings and real stakeholders with two case studies on ongoing active FOSS projects with at least 90% activity. As part of the validation process, the study required participants to declare how much they agreed with a set of statements at each level of maturity.

3.13 Consequences of Adopted Theories to Guide the Study

The current study triangulated the Technology Acceptance Model (TAM), Schema Theory (ST), and Gestalt Theory (GT) to create an alternative model for optimizing the design for UX in the development process of desktop-based projects. This section explains the implications of each theory in completing the investigation.

3.13.1 The Implications of TAM

The critical implication of the TAM is to lay a foundation for understanding user acceptance and adoption of UX practices. They include uplifting user acceptance, attitudes and perceptions, behavioral intentions, perceived usefulness, perceived ease of use, and user feedback and iterations. Generally, TAM implicates the study by ensuring the FOSS-UXMM can provide a framework for assessing and improving the level of UX practices and processes within organizations or communities. It can guide the development of strategies, guidelines, and interventions to enhance user acceptance, adoption, and satisfaction with UX practices.

Although limited literature explains the link between UX models and TAM, some scholars, such as Mlekus et al. (2020), have developed UX-exclusive TAM versions (see **Figure 3.9**)

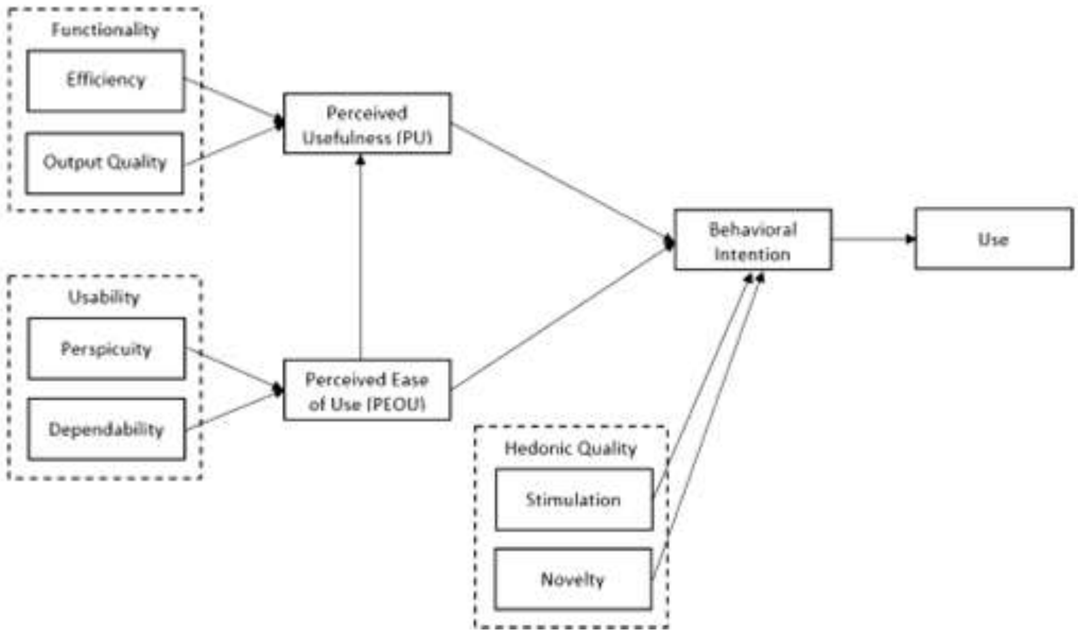

Figure 3.9: User Experience Technology Acceptance Model

SOURCE Mlekus et al. (2020)

3.13.2 The Implications of Schema Theory

According to the widely accepted guidance from ST, schemas are fundamental parts of human knowledge that can aid in developing intuitive interface designs (Huber, Wolf, Meyer, Jokisch, & Nowack, 2018). Furthermore, it is an essential component of human knowledge. It could be a good starting point for creating intuitive human-computer interaction environments that allow technology to anticipate human ability, decisions, and desires (Huber et al., 2018). According to Carsten et al. (2006) and Turner (2008), interface designs supported by schemas should allow effective system interaction based on the subconscious application of basic prior knowledge.

In optimizing the design for UX in desktop-based FOSS projects, schema theory can contribute knowledge by considering user mental models, designing for consistency, employing effective information architecture, incorporating progressive learning, and implementing error prevention and recovery mechanisms. This knowledge can lead to improved user experiences and increased UX maturity in the design process.

3.13.3 The Implications of Gestalt Theory (GT)

Contemporary practices approve of the GT as an essential block in building interactive systems that ease the learning process and the UX (Preece et al., 1994). The GT has affected various fields of study, including visual design. It has a historical context and prior application in aesthetics (Lim, Stolterman, Jung, & Donaldson, 2007), language patterns (Buffart & Jacobs, 2021), and diagram design (Lemon, Allen, Carver, & Bradshaw, 2007).

Regarding Optimizing the design for UX in the development process of desktop-based FOSS projects, Gestalt theory can enhance designers' understanding of visual perception, organization, and user attention by considering factors such as figure-ground relationship, visual hierarchy, closure, simplicity, and minimalism. The Gestalt model can guide the creation of visually engaging interfaces, intuitive and effective in facilitating user experiences.

GT principles helped the study understand how users' brains work. This knowledge is fundamental to accomplishing innovative design, knowing which elements are more effective and how each element influences perceptions, behavior change, and providing direct attention. Nevertheless, understanding how users’ brains work assists in problem-solving skills. For example, it aids goal-oriented and intuitive interface designs (Gkogka, 2018).

Finally, the GT assisted the study in uncovering how several stakeholders in the FOSS community perceive UX. Do they perceive it as a whole, or do they perceive each component individually? According to Rock and Palmer (1990), the features of a whole object are unique from those of the separate components that make up the object.

3.14 Chapter Summary

This chapter presented in-depth discussions of the study's methodology used during the investigation to achieve the desired objectives. Topics covered include the study area and the reasons for its selection, the research approach, the research design, the research settings, the study population, the sampling techniques and sample sizes, the study's validity and reliability, and the methods used for data collection. The chapter also covered the data analysis methods, the UX maturity model creation method, the implications of theories that guided the study, and the ethical standards noticed during the investigations. The thesis allocates chapter four for the study’s findings and their discussion.

CHAPTER FOUR
RESULTS AND DISCUSSION

4.1 An Overview

The research methods used in this study were provided in chapter three. This chapter presents and discusses the study's findings. The second subsection shows the results from the completed SLR regarding the applicability of current UXCMMs in assessing the UX maturity of desktop-based FOSS projects. The third subsection presents the analytical lenses for determining the preparation of desktop-based FOSS projects for UX maturity. Subsection four specifies factors affecting the UX maturity of desktop-based FOSS projects. Subsection five reports the development of a novel Free and Open open-source software UX maturity model (FOSS-UXMM), and subsection six validates the FOSS-UXMM. Subsection seven compares the FOSS-UXMM with other User Experience Capability Maturity Models (UXCMMs), and finally, subsection eight summarizes the chapter.

4.2 The Pertinency of Contemporary UXCMMs in FOSS Projects

4.2.1 Introduction

This subsection addresses research objective one, "**To examine the practical applicability of contemporary UXCMMs in the desktop-based FOSS projects.**" The results of this subsection address whether the current UXMMs are ideal for assessing the UX maturity of desktop-based FOSS projects. Can stakeholders use current UXCMMs to do self-assessments efficiently and cost-effectively? Finally, what variables do current UXCMMs consider significantly affecting UX maturity?

The study findings confirm the availability of numerous generically developed UXCMMs that organizations may adopt to evaluate their UX maturity (Kashfi, Feldt, & Nilsson, 2019; Namayala et al., 2022). However, their relevance in FOSS projects remains uncertain, leaving several research questions unanswered (Namayala et al., 2022; Raza et al., 2012a; Terry et al., 2010). For example, are current UXCMMs appropriate for assessing the UX maturity of desktop-based FOSS projects? Are they well-tested in genuine and active desktop-based projects? Finally, are there UXCMMs explicitly designed for analyzing desktop-based FOSS projects?

As a methodology, the study completed the SLR to contribute knowledge in response to the research questions listed above, which are also asked in the survey of Namayala et al. (2022)

4.2.2 Execution of the SLR

In executing the SLR, the study considered articles published between January 1985 and December 2020, written in English, and hosted in five (5) articles' repositories: Google Scholar, IEEE Xplore, ScienceDirect, Springer Link, and ACM Digital Library. Search strings (see **Figure 4.2**) were used to retrieve articles from the selected data sources. The inclusions/exclusions criteria adopted in this study are well stipulated in **Figure 4.1**.

Inclusions criteria
1. Publications that rigorously explain generic UX capability models or present enough details on UX capability models specific to the FOSS community
2. Publications present enough details on generic UX maturity models or explain FOSS specific UX maturity models.
3. Publications that present generic UXCMMs or explain FOSS specific UXCMMs.
Exclusion criteria
1. All publications that did not meet defined inclusions criteria or had no online accessible full documents
2. All publications that were not composed in the English language or were not peer-reviewed, such as books, work in progress, and technical reports

Figure 4.1: **Inclusions and Exclusions Criteria**

SOURCE: Namayala et al. (2022)

Repository	Contextualized search string(s)
IEEE Xplore Library	Search string one: ((('All Metadata':model OR framework OR prototype OR paradigm) AND 'All Metadata': capability OR maturity) AND 'All Metadata':UX or 'user experience') Search string two: (((('All Metadata': FOSS OR OSS OR FLOSS OR F/LOSS OR 'Open-Source Software' OR 'Free and Open Source Software' OR 'free/libre open-source software' OR 'free/open-source software' OR 'open source') AND 'All Metadata': model OR framework OR prototype OR paradigm) AND 'All Metadata': capability OR maturity) AND 'All Metadata': UX or 'user experience')
Google Scholar	Search string one: (UX OR 'user experience') AND (maturity OR capability) AND (model OR framework OR paradigm) Search string two: (UX OR 'user experience') AND (maturity OR capability) AND (model OR framework OR paradigm) AND (FOSS OR OSS OR 'open-source software' OR F/LOSS)
ACM digital library	Search string [[All: ux] OR [All: user experience]] AND [[All: maturity] OR [All: capability]] AND [[All: model] OR [All: framework] OR [All: prototype] OR [All: paradigm]] AND [Publication Date: (01/01/1985 TO 12/31/2020)] Search string two: [[All: ux] OR [All: user experience]] AND [[All: maturity] OR [All: capability]] AND [[All: model] OR [All: framework] OR [All: prototype] OR [All: paradigm]] AND [[All: foss] OR [All: oss] OR [All: floss] OR [All: f/loss] OR [All: 'open-source software'] OR [All: 'free and open-source software'] OR [All: 'free/libre open-source software'] OR [All: 'free/open-source software'] OR [All: 'open source']] AND [Publication Date: (01/01/1985 TO 12/31/2020)]
Science Direct	Search string one: (UX OR 'user experience') AND (maturity OR capability) AND (model OR framework OR prototype) Search string two: (UX OR 'user experience') AND (maturity OR capability) AND (model OR framework) AND ('open-source software' OR FOSS OR FLOSS)
Springer Link	Search string one: (UX OR 'user experience') AND (maturity OR capability) AND (Model OR framework OR prototype OR paradigm) Search string two: (UX OR 'user experience') AND (maturity OR capability) AND (Model OR framework OR prototype OR paradigm) AND (FOSS OR OSS OR FLOSS OR F/LOSS OR 'Open-Source Software' OR 'Free and Open-Source Software' OR 'free/libre open-source software' OR 'free/open-source software' OR 'open-source')

Figure 4.2: Contextualized Search Strings

SOURCE: Namayala et al. (2022)

After reviewing the retrieved articles and removing duplicates, eight (8) papers were found relevant and rich in information about UXCMMs (see **Table 4.1**). Many of these articles were published between 2014 and 2020.

Table 4.1: **Papers Selected in SLR**

Article ID	Article title	Author(s)/Year	Category	Repository
A1	AGILE UX Model: Towards a Reference Model on Integrating UX in Developing Software Using Agile Methodologies	(Peres et al., 2014)	Conference	GS and IEEE
A2	User Experience Maturity Model for eCommerce Websites	(Anchahua et al., 2018)	Conference	GS and IEE
A3	Corporate User-Experience Maturity Model	(Van Tyne, 2009)	Conference	SL and GS
A4	A UX Maturity Model: Effective Introduction of UX into Organizations	(L. Chapman & Plewes, 2014)	Conference	SL and GS
A5	Specification of a UX process reference model for the strategic planning of UX activities	(Kieffer, Rukonić, Kervyn de Meerendré, & Vanderdonckt, 2020)	Conference	GS
A6	From Snake-Oil to Science: Measuring UX Maturity	(Sauro et al., 2017)	Conference	ACM and GS
A7	Measuring UX Capability and Maturity in Organizations	(Rukonić et al., 2019)	Conference	SL and GS
A8	Enhancing user value of educational technology by three-layer assessment	(Vuorio, Okkonen, & Viteli, 2017)	Conference	ACM

SOURCE: Namayala et al. (2022)

Nevertheless, the study's expected outcomes, research questions, a quality checklist, and a review protocol plan were considered to define the articles' data extraction strategy. Finally, statistical analysis techniques with the help of Excel and SPSS software were adopted to analyze collected data and present findings.

4.2.3 Findings and Discussions

Following data analysis, the study discovered that existing UXCMMs used diverse development methodologies, scales, and assessment techniques. Nevertheless, they were created by different authors and comprised various variables primarily based on the authors' working experience. These findings confirm results from the studies of Lacerda et al. (2018), Rukonić et al. (2019), and Sauro et al. (2017).

Identified UXCMMs were not linked in any way, and knowing the structure and operations of one UXCMM could not assist in understanding the structure and functions of another UXCMM. However, many shared some characteristics, such as PAs and Key Performance Areas (KPAs). They mainly considered the availability of dedicated UX resources, when and how to involve UX practices, user needs, user research, UCD practices, organizational culture, and attitude towards UX to express UX maturity. These observations confirm results from the studies of Feijó (2010), Fraser and Plewes (2015), and Nielsen (2006).

However, UXCMMs under discussion lack standard guidelines, approaches, checklists, detailed documentation, and frameworks to dictate their development methodologies. This observation was also identified by Biberoglu and Haddad (2002), Otto et al.(2020), and de Bruin et al.(2005). In addition, identified models lack the standardized vocabulary that researchers can use to compare them in understanding performance supremacy. The observation is also confirmed by the study of Monteiro and Maciel (2020).

Identifying the differences between the UX maturity levels of the acknowledged UXCMMs was challenging and nearly impossible. This observation confirms the findings of Sauro et al. (2017), who argued that current UXCMMs are nothing but sales gimmicks. Likewise, contemporary UXCMMs have not been tested or confirmed in the FOSS community, casting doubt on their general applicability and suitability. They also do not have adequate documentation to explain their practical relevance, and most significantly, none of UXCMM was developed explicitly to assess the UX maturity of the FOSS projects despite the FOSS community owning different dynamics than other software-developing communities (Namayala et al., 2022).

In its discussion, the study of Namayala et al. (2022) was unconvinced with the pattern of using the generic UX maturity influencing factors in creating UX maturity models for every industry, mainly the FOSS community. While providing recommendations for future studies, Namayala et al. (2022) advised more explorations of how the body of knowledge can experimentally create the FOSS community's specific metrics that may address its pertinent dynamics. It also recommended experimentally created FOSS-exclusive UX maturity models that may provide an ideal assessment of UX maturity.

4.2.4 Conclusion and Way Forward

Generally, available UXCMMs in the current form and structure may not be applied in the FOSS community. The FOSS community's structure has confirmed its rigidity to accept them, plus other UX/HCI practices. These findings warrant the need for empirically developed FOSS-exclusive UX maturity models or to adapt those available to account for the FOSS community's dynamics. The established or modified UX maturity models must accommodate UX maturity influencing factors exclusive to the FOSS community that may be established by involving the right stakeholders from active FOSS projects. However, identifying project stakeholders with knowledge of factors for designing products with desirable UX and projects ready for achieving higher levels of UX maturity when assessed with appropriate UXCMM has been challenging.

To date, the FOSS community does not have a consistent procedure for identifying FOSS projects with UX-knowledgeable stakeholders and those prepared for higher levels of UX maturity. However, identifying projects with UX/HCI stakeholders and accommodating UX/HCI practices may be significant in sourcing reliable perceptions on various UX influencing factors pertinent to the community. As a result, the FOSS community may need to investigate how it may develop a consistent approach for identifying desktop-based projects ready for UX maturity-related activities that can be utilized steadily across desktop-based FOSS projects and produce desirable results.

4.3 Application of the Fuzzy Delphi Method to Determine Analytical Lenses for Determining the Preparation of Desktop-based FOSS Projects for UX Maturity

4.3.1 Introduction

This thesis subsection addresses research objective two, "**To determine analytical lenses for analyzing the preparation of the FOSS projects for UX maturity."** The results of this section may serve as a foundation for understanding how desktop-based projects are prepared or flexible to implement requirements that, when added together, represent higher levels of UX maturity. Results may help determine whether the FOSS community is moving in the right direction (Cohen, 2022) and better implement institutional UX practices and assessment measures. They may also show how much the FOSS community embraces UX concepts, how well it integrates UX practices into its culture, and how open or closed it is toward changing its design decisions while favoring UX practices.

Achieving higher levels of UX maturity is a dream of any software project, including FOSS (Pernice et al., 2021). However, the extent and pace of this aspiration may vary depending on factors like the organization's culture, available resources, and the software's specific nature (Cheng & Guo, 2018). Ideally, UX maturity is a continuous endeavor to produce more substantial UX processes, technology, and tools. At higher levels of UX maturity, organizations create robust UX designs that may engage relevant UX professionals, methods, practices, principles, and tools (Von-Wangenheim et al., 2010). However, several projects, particularly in the FOSS community, have low usability maturity levels, which may imply low UX maturity levels (Raza et al., 2012a) and frequently overlook and undervalue usability and UX issues (Cheng & Guo, 2018).

Implementing robust UX design in developing desktop-based FOSS projects that guarantee organizations achieve higher levels of UX maturity necessitates extreme dedication, preparations, and a workforce with the necessary UX skills (Cheng &

Guo, 2018; Cohen, 2022; Inal et al., 2020). However, the preparation overpowers all other prerequisites because it may account for dedication and resource allocation, including a skilled workforce. It amounts to completing all upfront UX-related tasks before the organization is considered a UX-matured. Ideally, when a FOSS community or any other organization is not prepared to achieve higher levels of UX maturity, adopting UX maturity models (UXCMMs) that evaluate organizational strengths and weaknesses in implementing UX-related tasks may not make a difference. Therefore, the absence of proper preparation for implementing UX-related activities runs the danger of holding sessions that may not provide the necessary insights into UX and could waste time and effort (Otchere, 2022).

Although some FOSS projects have dedicated UX teams and actively prioritize UX as preparation for reaching higher levels of UX maturity, others are still in the early stages of embracing UX principles. With the current setup, the FOSS community misses adequate preparations and resources to implement its activities (Nichols & Twidale, 2006), including UX-related tasks. Decision-makers do not value UX (Cajander, Gulliksen, & Boivie, 2006; Hedberg et al., 2007; Wale-Kolade & Nielsen, 2016), lack institutionalization of UX-related activities, where nobody oversees their integration in the same manner as projects' core functions (Ardito, Buono, Caivano, Costabile, & Lanzilotti, 2014; Wale-Kolade & Nielsen, 2016). Although non-code contributors are significant in the well-being of FOSS projects (Carillo, Huff, & Chawner, 2014), the community's structure and culture do not accept them (Cheng & Guo, 2018; Christian & Vu, 2021; Crowston, Wei, Howison, & Wiggins, 2012; Terry et al., 2010).

HCI/UX practitioners are not smoothly recruited into FOSS projects (Cohen, 2022; Hedberg & Iivari, 2009; Llerena, Rodriguez, Llerena, et al., 2018; Neeman, 2012). As a result, few FOSS projects rarely involve UX teams and professionals, mainly at the very end. However, it is tricky for them to jump on a moving train and address issues with the user interface, task flows, and functionality (Rajanen, 2023). Nevertheless, several other UX/HCI practitioners who try to contribute to FOSS projects face difficulties (Ardito, Buono, Caivano, Costabile, & Lanzilotti, 2014; Wale-Kolade & Nielsen, 2016) and are challenged by several FOSS community-specific limitations, however not as severe as decades ago (Inal et al., 2020) due the emerged substantial community's transformation (Franco-Bedoya, Ameller, Costal, & Franch, 2017; Steinmacher, Robles, Fitzgerald, & Wasserman, 2017). Besides other prerequisites, having the right UX teams with the necessary skills is critical for preparing the FOSS projects for higher levels of UX maturity because they critically plan, execute, monitor, and evaluate UX-related activities to develop the right UX tools, methods, principles, and strategies.

Despite other researchers' significant initiatives to elevate the status of accommodating UX teams and other HCI/UX specialists within the FOSS community (Inal et al., 2020), it has been challenging to foster collaboration between UX experts and contributors, increasing UX awareness and education, and creating standardized frameworks or guidelines for preparing and evaluating UX maturity. Nevertheless, it is still debatable how currently offered interventions prepare organizations for successfully meshing the knowledge of UX teams and HCI/UX specialists with those of organizations and the software development life cycle (Bruun, Larusdottir, Nielsen, Nielsen, & Persson, 2018; Kou & Gray, 2018; Marsden & Holtzblatt, 2018).

Preparing the FOSS community for UX maturity assessment may help determine whether the community is moving in the right direction regarding implementing UX-related activities (Cohen, 2022). It may also show how much the community embraces UX concepts, how well it integrates UX practices into its culture, and how open or closed it is toward changing its design decisions while favoring UX practices. For example, desktop-based FOSS projects ready for UX maturity are forecasted to achieve higher UX maturity levels when the right UXCMMs are applied. They may also be successful and widely adopted. However, the completed literature review shows limited research papers examining how prepared the FOSS community is to assess projects' UX maturity. As a result, it is still challenging to determine whether there has been an improvement in how the FOSS community prepares for UX maturity(Cheng & Guo, 2018; Inal et al., 2020). The lack of studies examining how the FOSS community is prepared for UX maturity raises several questions and speculations for the FOSS community. For example, does the FOSS community prepare for UX maturity? Is there a consistent way of analyzing and reporting the preparation for UX maturity in the community? Are there FOSS-specific analytical lenses and consistent procedures to follow? Finally, what is the relationship between analytical lenses and the UX maturity of desktop-based FOSS projects?

This subsection provides distinctive theoretical and methodological contributions to the body of knowledge by offering novel and consistent processes for evaluating how the FOSS projects are prepared for the UX maturity assessments. It has also completed an empirical study to validate formulated analytical lenses in real desktop-based FOSS projects. The subsection has responded to research questions.

RQ1: What analytical lenses are suitable for analyzing the preparation of desktop-based FOSS projects for UX maturity?

RQ2s: What is the relationship between the identified analytical lenses and the preparation of desktop-based FOSS projects for UX maturity?

The remainder of the work is organized into subsections to facilitate discussion and presentation of results. The second subsection is for background information, and the third is devoted to the study's methodology. The findings are presented in the fourth subsection. The discussion of the findings is presented in the fifth subsection. Finally, subsection six provides conclusions and a way forward.

4.3.2 Theoretical Background

4.3.2.1 Free and Open-Source Software

In the current economy, businesses must seek ways to lower expenses without sacrificing operations quality (VonFange & Lavigne, 2011), and FOSS is a viable alternative compared to expensive proprietary software (Abramova et al., 2016; Shekgola et al., 2021). They have a low Total Cost of Ownership (TCO) (Bwalya et al., 2019; S. Dhir & Dhir, 2017; Namayala et al., 2022; Patino-Toro et al., 2022; V. R. Sanchez et al., 2020). Therefore, they may often be adopted to close the digital divide gap caused by income differences (AlMheiri et al., 2018). They certify four mandatory user privileges: (1) can be used for any purpose, (2) developers are free to modify their source codes to fit needs, (3) can be redistributed without authors' consent, and (4) its source codes can be improved (Baum, Huff, Clingan, & Bostrom, 2013; Free Software Foundation, 2012; Kilamo et al., 2020; Srinivasa, 2017). However, adhering to the terms specified by the Open Source Software Initiative is still essential when redistributing the modified source codes (2010).

FOSS projects offer quality technology (Kuwata et al., 2014) and the opportunity to disengage from technical bonds by developing local competencies (Pezer, Lazic, & Odak, 2017; V. R. Sanchez et al., 2020). They are also highly flexible, dependable, stable, and secure (Nagy et al., 2012; V. R. Sanchez et al., 2020; Umm-E-Laila et al., 2017). FOSS projects often provide a few variations in vulnerability severity from proprietary software, and the current software development paradigm sees FOSS projects as something to invest in and work on for businesses to adopt (Butler et al., 2022). According to the Open-Source Software Foundation (FSF), "free" refers to the freedom to use the software, not necessarily free of charge. However, unlike proprietary software that demands upfront payment and a cathedral model, the FOSS projects primarily adopt a bazaar model (Benson, 2016; Prasad & Reddy, 2013). They are built-in modular forms with single or multiple developers responsible for one or all the modules (Mockus, Fielding, & Herbsleb, 2000).

Adopting FOSS components for integration into other software is becoming standard practice in the software development industry (Butler et al., 2022). One reason for adopting FOSS is to bypass resource-intensive software development processes (Kazimierczak et al., 2020; Petersen et al., 2018; Spinellis, 2019). As a result, recent estimates indicate that the FOSS source code makes up 80–90% of all contemporary software (Nagle & Hoffman, 2020). Nevertheless, Spinellis (2019) argues that with today's technological advancement in software development, if most of the code in somebody's product is not FOSS, they are likely wasting effort and cash reinventing the wheel. The current body of knowledge acknowledges the presence of several officially described and less formal or developer-led interventions to overcome difficulties with adopting FOSS projects or their components (Butler et al., 2022). For example, Silva et al. (2023) evaluated how cost, quality, usability, interoperability, and security influence users' intention to use FOSS projects.

Regarding full-fledged FOSS applications adoption, server-based applications, such as Apache and MySQL, enjoy a considerable share (Bahamdain, 2015; Nagy et al., 2012; Oreku & Mtenzi, 2013; Robles, Steinmacher, Adams, & Treude, 2019), particularly in developed countries (AlMheiri et al., 2018; Patino-Toro et al., 2022). They are considerably superior to proprietary software in ethics and social responsibility (FSF, 2011; Kuwata et al., 2014), and are now critical in almost every industry and are extensively relied on by the public and private sectors, including many non-IT firms (Marois et al., 2022). They can efficiently support multiple dimensions, such as health, education, and economics (Androutsellis-Theotokis et al., 2010; Tsou & Smith, 2011) and are de facto standards, sparking the development of new complementary products and services in fields like cloud computing, big data, and the Internet of Things (IoT) (Petrov & Obwegeser, 2018).

In contrast, desktop-based FOSS applications are not widely adopted (Masson et al., 2017; Nagy et al., 2012; Oreku & Mtenzi, 2013), especially in most public sectors in developing countries, such as Tanzania (Silva et al., 2023). As a result, many government agents and individuals using desktop computers are still paying for proprietary or resorting to counterfeit or pirated software (Amrollahi, Khansari, & Manian, 2014; Jokonya, Kroeze, & van der Poll, 2013). The contributions of desktop-based FOSS projects to a global market share are comparatively minor. (Colombo et al., 2014). This argument is supported by data from Statcounter (2019), which indicates that 73.75% of the market share is for Windows desktop operating systems, and only 2.09% is reserved for Linux desktop operating systems. In addition, the lack of production-ready Linux desktop software and the continued successes of Microsoft Office serve as additional justifications for the low approval of desktop-based FOSS projects (Yeates, 2015).

The ongoing debate as to why desktop-based applications are not highly adopted indicates that desktop-based FOSS projects offer little evidence of attempts to satisfy the persistent requirements of users and are relatively complex for non-technically savvy individuals to install, maintain, and use. Silva et al. (2023) state that many current problems in desktop-based FOSS projects are linked to low quality and poor UX. They are also associated with incompatibility with other commonly used hardware and software, user reluctance, poor performance, community attitudes, lack of support, and inadequate community awareness (Silva et al., 2023). However, poor UX leads (Berendes et al., 2022; Cheng & Guo, 2018; S. Dhir & Dhir, 2017; Masson et al., 2017). However, poor UX has been recurring as the leading cause of desktop-based FOSS applications' adoption misfortune (Cheng & Guo, 2018; S. Dhir & Dhir, 2017; Masson et al., 2017). As a result, instituting initiatives that may increase FOSS projects' UX increases adoption.

4.3.2.2 UX in the Free and Open Source Software Community

UX is a relatively new phenomenon (Law & Van Schaik, 2012; Wallach et al., 2017), and several related issues are unexplored. It lacks a universally accepted definition (Pettersson, Lachner, Frison, Riener, & Butz, 2018; Roto et al., 2010; Zarour & Alharbi, 2017) and can pride different meanings based on how it is approached. The All-About UX website has over twenty-seven definitions of UX. However, the descriptions provided by ISO 9241-210 (2019b) and Hassenzahl and Tractinsky (2006) are commonly adopted. Despite disagreements on how UX can be expressed, it is agreed that UX encompasses more aspects of the system interactions than usability and functionality (Hassenzahl et al., 2021; Law & Abrahão, 2014).

The FOSS community differs from other software development communities (Bach & Twidale, 2010), justifying its rigidity in adopting current UX/HCI practices. Studies, such as Terry et al. (2010), show that to adopt current UX/HCI practices, the FOSS community must modify them. Current initiatives for enhancing UX in the FOSS community exist mainly due to the close and ongoing contact between developers and other core users (Twidale & Nichols, 2005). This conclusion supports findings from Namayala et al.(2022), Jokela et al. (2006), and Lacerda and von Wangenheim (2018).

There are FOSS projects with desirable UX, such as Mozilla Firefox (Souley & Sambo, 2013). However, perceptions of UX design and its assessment in the FOSS community vary between independent developers and developers from well-coordinated projects such as TensorFlow, the Open Source Machine Learning Framework with more than 3300 contributors (Rajanen, 2023). Because of differences in UX perceptions among FOSS projects, several projects continue to have poor UX and offer room for improvement. Being the organizational initiative (Chapman & Plewes, 2014), UX in the FOSS community is evaluated by considering UX maturity, which expresses the projects' ability to execute UX best practices, research, and related activities.

4.3.2.3 UX Maturity in Free and Open-Source Software

Like UX, scholars can express UX maturity differently (Vallerand, Lapalme, & Moïse, 2017). For example, Pernice et al.(2021) define it as "the measures of an organization's desire and ability to deliver User-Centred Design successfully," while Molich (2021) describes it as "the ability of an organization or a product team to define and meet UX goals that help to achieve business goals." UXCMMs present various stages explaining the continual improvement of implementing UX-related processes from worst to best (Paulk et al., 1993). By doing so, they can measure the organizational strengths and flaws and make suggestions for improvement (Pernice et al., 2021).

Despite discrepancies in definitions, several measures may be used to understand the status quo of UX maturity in the FOSS community. For example, the institutionalization of organizational UX, understanding the flexibility and rigidity of organizations toward UX culture, creating UX assessment frameworks, and employing the UX capability maturity Models for UX (UXCMMs). However, UXCMMs dominate and complement other techniques but cannot replace them (MacDonald, 2019; MacDonald, Sosebee, & Srp, 2022).

Although various UXCMMs exist, analyzing the UX maturity of the FOSS community is a bit challenging undertaking. It varies depending on the communities involved and the project's size, resources, and dynamics. While some FOSS projects have more advanced methods, others still rely on informal feedback or have limited resources to provide comprehensive assessments.

4.3.2.4 Preparation of the FOSS Community for UX maturity

Several initiatives are underway in the FOSS community that are indirectly preparing the community for UX maturity. They include examining factors influencing the adoption of FOSS projects, adopting generically developed UX maturity models, and validating their applicability in the community. For example, Shekgola et al. (2021) explored the potential driving forces behind South Africa's adoption of FOSS projects for records management. Likewise, the study of Johnston et al. (2013) provided insights into the factors influencing or inhibiting OSS adoption within Western Cape schools.

However, much of the initiatives have been on creating UXCMMs experimentally and accounting for as many UX metrics as possible. For example, Rukonić et al. (2019), Raza et al. (2012a), Pernice et al.(2021), and Chapman and Plewes (2014). Other aspects are not highly discussed. For example, how to increase the knowledge of human resources in the community to hire or include HCI/UX professionals with backgrounds in non-IT fields like psychology, social science, and management to help improve current models and their contributions. Nevertheless, compared to other quality attributes, preparing for UX maturity lacks a more thorough picture and comprehension of UX problems (Kashfi et al., 2017), and practitioners do not uniformly apply most available practices across projects and the community (Kurosu, 2017; Namayala et al., 2022; Raza et al., 2012a).

Namayala et al. (2022), Kashfi et al. (2019), Guo (2012), Roto et al.(2010), Ardito et al.(2014), and Vermeeren et al.(2010) have confirmed that the FOSS community is still mixing usability from UX where practitioners and researchers mistakenly use usability and UX design interchangeably(A. Dhir & Al-Kahtani, 2013; Guo, 2012; Kashfi et al., 2017; Nguyen, 2020). Some practitioners and researchers contend that usability capability maturity models (UCMMs) may also fit in assessing UX. This misconception has added more complexities to whether the FOSS community is preparing to determine UX or usability maturity. Alves et al.(2014) and Lallemand et al.(2015) have been concerned with misconceptions.

4.3.2.5 Related Works

This research did not find specific studies examining how prepared the FOSS community is for its projects' UX maturity. However, the significance of UX in the community is crystal clear to many stakeholders, and there have been initiatives to enhance the UX, such as creating specialized UX teams or contributors to improve usability and user-friendliness. For example, Namayala et al. (2022) investigated the relevancy of available UXCMMs in the FOSS community and whether the community has tools, resources, and methods exclusive for assessing its maturity. Cheng and Guo (2018) highlighted the importance of understanding how UX issues, including assessment, are currently raised, discussed, negotiated, and eventually addressed by the FOSS community. Nonetheless, Llerena et al.(2018) examined and adapted the focus group discussions to suit the FOSS community to raise FOSS projects' UX maturity.

4.3.2.6 Limitations of Related Works

There is limited evidence to support the availability of studies specific to determining the preparation of the FOSS projects for UX maturity. Available initiatives mainly explore efforts that may make UX-related activities compatible with the FOSS culture (Iivari, 2010; Rajanen & Iivari, 2013, 2015b) and build credibility and trust by demonstrating UX expertise through statistics and data or making a modest but steady entry point into the FOSS community (Bach & Twidale, 2010). However, they are established on the expertise and motivation of players (Rajanen & Iivari, 2015a) and lack consistent procedures, processes, and analytical lenses. Thus, they are regarded as informal with unguaranteed outcomes (Hasan & Al-Sarayreh, 2015).

Meanwhile, present interventions lack methodological rigor and inadequately address the FOSS community's complex structure and massive data (Zhao, Deek, & McHugh, 2010). They do further not use emerging technologies, such as artificial intelligence (AI) and machine learning (ML), to develop more intuitive and customized techniques to support UX designers in making data-driven design decisions and predictive analysis to forecast future user behaviors, needs, and preferences (Dana, 2023; Donahole, 2021; Yang et al., 2020). AI and ML can further be used to automate consistency verifications of interfaces and collect and report systems' state information (Yang et al., 2016).

Finally, to the author's knowledge and completed literature review, this is the first study, and the accumulated data and findings add to earlier analyses of the FOSS community's readiness to evaluate its projects' UX maturity.

4.3.3 The Research Methodology

This empirical study is a mixed research design using qualitative and quantitative approaches and underwent three phases to collect, analyze, and present the findings. In the first phase, the literature review was exhausted to establish preliminary analytical lenses that might be used to analyze the FOSS community for UX maturity. The final list of ten analytical lenses (see **Table 4.2**) was established using the traditional Delphi method (DM) steps: brainstorming and narrowing down.

Table 4.2: The Analytical Lenses that May be Relevant for the FOSS Projects

Id	Analytical lenses (Items)	Source(s)
A1	Availability of UX strategic plans	Hussein and Mahmud (2012) and
A2	Accommodate UX professionals as decision-makers.	Cheng and Guo (2018), Inal et al. (2020), and Kashfi et al. (2019)
A3	Clear UX roles for each stakeholder	Gibbons (2023)
A4	Value UX-related tasks.	Wale-Kolade and Nielsen (2016)
A5	Adopting UCD practices in design	Raza et al. (2010)
A6	Decision-makers buy-in and support UX assessment initiatives.	Hussein et al.(2015), Nichols and Twidale (2006) and Inal and Güner (2016).
A7	Availability of FOSS-exclusive UX maturity assessment tools	Namayala et al. (2022), Terry et al. (2010), Isomursu et al.(2012), Law et al.(2014) and Yeates (2015).
A8	Accommodate UX principles in FOSS design.	Wale-Kolade and Nielsen (2016), Iivari (2006) and Llerena et al.(2018).
A9	Uniform understanding of UX concepts.	Garcia et al.(2019), Lallemand et al. (2014), and Kuusinen (2015).
A10	Quantify UX maturity assessment metrics.	Sauro and Lewis (2016),

The fuzzy Delphi method (FDM) analysis was adopted in the second phase to establish agreement among UX experts and rank the proposed analytical lenses based on priority. The FDM combines fuzzy theories and classical DM to improve vagueness and ambiguity (Lee, Chen, Tsui, Che, & Chiang, 2021; Murray, Pipino, & Van Gigch, 1985). The total value for the defuzzification of triangular fuzzy numbers was computed using the fuzzy set, defuzzify, presented by Chen and Hwang (1992). To make it easier for experts to respond to the questionnaire, the author replaced the Fuzzy value with a 7-point Likert scale, and the link between the two scales is illustrated in **Table 4.3**.

Table 4.3: Triangular Fuzzy Scoring and 7-Points Likert Scale

Likert Scale Scoring	Linguistic variable	Fuzzy Scale scoring
1	Extremely unimportant	(0.00, 0.00, 0.10)
2	Very unimportant	(0.00, 0.10, 0.30)
3	Unimportant	(0.10, 0.30, 0.50)
4	Moderately important	(0.30, 0.50, 0.75)
5	Important	(0.50, 0.75, 0.90)
6	Very important	(0.75, 0.90, 1.00)
7	Extremely important	(0.90, 1.00, 1.00)

SOURCE: Habibi et al. (2015)

The execution of the FDM started with converting all collected linguistic variables into triangular fuzzy numbers (TFNs) that produced fuzzy scales written as (L, M, U), where L indicates the smallest possible value, M represents a respectable value, and U represents the largest possible value. In the second step, the author computed each fuzzy number's average response (fav) using the formula Manakandan et al. (2017) described.

$$fav = \left(\left(\frac{\sum_{i=1}^{n} L_i}{n} \right), \left(\frac{\sum_{i=1}^{n} M_i}{n} \right), \left(\frac{\sum_{i=1}^{n} U_i}{n} \right) \right)$$

For the sake of this study, the fuzzy average numbers are represented as (m1, m2, m3).

In the third step, the study identified the threshold (d(m,n)), which denotes the distance between the fuzzy scale m(L, M, U) and its average n(m1,m2,m3). The threshold is essentially used to identify the level of agreement between experts (Ishikawa et al., 1993) and was calculated using the formula.

$$d(m,n)=\sqrt{\frac{1}{3}\left[\left(m_1-L\right)^2+\left(m_2-M\right)^2+\left(m_3-U\right)^2\right]}$$

If the threshold value (d(m,n)) of each screened item does not exceed 0.2, the variables satisfy one of the three conditions for obtaining expert consensus (Chen, 2000; Cheng & Lin, 2002), but the total percentage of d(m,n) ≤ 2 must also be 75% or above (Chu & Hwang, 2008; Murry & Hammons, 1995).

The author derived the crisp value (Z*) that identifies the alpha-cut aggregate level of fuzzy assessment in the fourth stage using the formula,

$$Z^*=\max\left(A_1, A_2, A_3\right)$$

$$A_1=\frac{\left(m_1+m_2+m_3\right)}{3}, A_2=\frac{\left(m_1+2m_2+m_3\right)}{4}, A_3=\frac{\left(m_1+4m_2+m_3\right)}{6}$$

If all resulting Z* values are greater than the α-cut value = 0.5, the item indicates expert agreement and is thus accepted (Bodjanova, 2006).

The acceptance of all criteria is tested in the fifth and final stage. For this investigation, a threshold of 0.5 was chosen to designate low-ranked items. As a result, all entries with Z* less than 0.5 were considered relevant for exclusion and deletion.

The FDM requires knowledgeable participants who can provide valuable findings (Marois et al., 2022). As a result, the final selection of experts followed Okoli and Powlowski's (2004) principles. According to Booker and McNamara (2004), experts are competent individuals who are knowledgeable and have knowledge as a consequence of the training, practice, and experience they have received. As a result, only scholars working as academicians who published, read, and reviewed scholarly publications on UX/usability maturity assessment were invited (see **Table 4.4**). Nevertheless, these experts were either active contributors to UX/usability evaluation research or utilized UX/usability evaluation research for more than ten years. The initial list of scholars containing names and contacts was continually updated using snowballing.

The recruitment process of the experts took place between mid-September 2021 and November 2022, with an initial response rate of 41.38% (24/58). Finally, 11 experts agreed to participate and signed consent forms. The sample of 11 experts is based on Yusoff et al.'s (2021) recommendation that seven is the minimum number of experts required, which is supported by Mustapha et al. (2021), who reported that seven samples are adequate in FDM if the experts are highly homogeneous. As a result, eleven experts are sufficient to gather knowledge and reach an expert consensus.

Table 4.4: Profiles of UX/Usability Professionals

Gender	Experience (years)	Age	Country	Title	Research Areas
Male	18	61	Tanzania	Professor	eLearning, HCI, and success in IS
Male	10	67	Finland	Professor	UX, UX design, and automatic HCI.
Male	11	58	Germany	Professor	UX, HCI, ubiquitous computing, mobile interaction, and human-centered AI.
Male	22	55	USA	Professor	Persuasive Technology, UX, Interaction Design, and HCI.
Female	25	55	Finland	Professor	UX design and HCI
Male	19	49	China	Associate Professor	Conversational Agents (Chatbots), Affective

					Computing, and HCI
Male	16	48	France	Associate Professor	Cognitive science, emotion, HCI, and computational modeling
Male	23	65	Brazil	Professor	HCI and accessibility
Male	15	54	Italy	Professor	End-User Development, HCI, Visual Interactive Systems, Theory of Visual Languages, and WWW Interfaces
Male	15	45	USA	Senior lecturer	UX, Mobile HCI, and Social Media
Female	12	52	Tanzania	Senior lecturer	HCI, FOSS, and ICT4D

In the third phase, the study empirically validated the identified lenses in the actual setting by consulting stakeholders selected from the sixty-two FOSS projects purposely selected from FOSS source code hosting facilities referred to as repositories.

The 5-point Likert scale questionnaire was adopted to collect stakeholders' opinions. Involved stakeholders were purposely selected based on their involvement in the selected projects' mailing lists and other discussion forums. Personalized e-mails explaining the study's objectives and requesting participation consent were sent to respondents. Used e-mails were collected from projects' mailing lists, and the recruitment process of the stakeholders occurred between mid-September 2021 and November 2022, with an initial response rate of 6.68% (218/3,265). Finally, 70 participants agreed to participate and signed consent forms. **Table 4.5** provides details of the participants' specialties.

Table 4.5: Stakeholders' Specialties

Sn.	Specialty	Frequency	Valid Percent
1	Software Engineer	16	22.9%
2	Computer Scientist	12	17.1%
3	Programmer	11	15.7%
4	Interaction Designer	7	10.0%
5	HCI Specialist	13	18.6%
6	Other	11	15.7%

The stakeholders played different roles in the desktop-based projects specified by the core-periphery structure (see **Table 4.6**).

Table 4.6: Roles Played by Engaged Stakeholders

Sn	Role	Frequency	Valid Percent
1	Project Owner	6	8.60%
2	Project Manager	13	18.60%
3	Core Developer	7	10.00%
4	Contributors	12	17.10%
5	Active User	21	30.00%
6	Passive Users	6	8.60%
7	Bugs fixers/reporters	5	7.10%

Finally, they had different experiences working with desktop-based FOSS projects that ranged between 0 to 15+ years (see **Table 4.7)**

Table 4.7: Stakeholders' Working Experience in FOSS Projects

Sn	Experience in FOSS	Frequency	Valid Percent
1	0 - 5 Years	18	25.7%
2	5 - 10 Years	28	40.0%
3	10 - 15 Years	14	20.0%
4	15+ Years	10	14.3%

A regression equation below established from the study's conceptual framework was adopted to explain the relationship between analytical lenses and their effects on the readiness of the FOSS community to assess the UX maturity of its projects.

$$m = \gamma_0 + \gamma_1 t_1 + \gamma_2 t_2 + \gamma_3 t_3 + \gamma_4 t_4 + \gamma_5 t_5 + \gamma_6 t_6 + \gamma_7 t_7 + \gamma_8 t_8 + \gamma_9 t_9 + \gamma_{10} t_{10} + \varepsilon$$

M is the preparation of the desktop-based FOSS projects for UX maturity, the dependent variable. γ_0 is constant, $\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6, \gamma_7, \gamma_8, \gamma_9, and\ \gamma_{10}$ are the path coefficients, and $t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9\ \ and\ t_{10}$ are the ten (10) analytical lenses, independent variables, and ε is the error component considering all unknown variables.

The case study adopted z-values of skewness and Kurtosis to test for data normality and the Variance Inflation Factor (VIF) to test data multicollinearity. Finally, before performing an actual study on the relationship between analytical lenses and the UX maturity of desktop-based FOSS projects, we conducted a pilot study involving fifteen (15) stakeholders selected under the same inclusions/exclusions criteria as those of the fundamental research. Pilot studies often assist in designing and modifying the principal analysis (Thabane et al., 2010). They may prevent costly and time-consuming analyses of fatal defects (Polit & Beck, 2017), mainly test the reliability and validity of constructs, and are regarded as confirmatory tests to improve the quality and efficiency of the actual research (In, 2017).

4.3.4 Results

4.3.4.1 Findings from Analysis of Expert Consensus on Analytical Lenses

As the analysis shown in **Table 4.8** shows, the threshold values for all items of preparing FOSS projects for UX maturity are less than 0.2, suggesting that experts' opinions agree. Nevertheless, the average value of all items' threshold value (d) is less than 0.2, which is 0.112. Generally, the items have achieved expert agreement because the average threshold value (d) is smaller than 0.2. Cheng and Lin's (2002) and Chang et al.'s (2011) investigations support this finding. Likewise, the overall percentage of expert agreement is 81%, greater than 75%, suggesting that the items meet the expert agreement requirements. All Alpha-Cut defuzzification levels are greater than the α-cut value (0.5), confirming the experts' agreement on the items. This conclusion is supported by the findings from the studies of Bodjanova(2006) and Tang and Wu (2010). Finally, the analytical lenses for analyzing the preparation of the desktop-based FOSS projects for UX maturity are ranked based on their priority. Decision-makers' buy-in and support of UX assessment initiatives are ranked first, and the value of UX-related tasks and adopting UCD design practices are ranked last.

Table 4.8: Findings from the Analysis of Experts' Opinions

Items	**Triangular Fuzzy Numbers**			**Defuzzification**		
	Average of Fuzzy Numbers	**Threshold Value (d) by Item**	**Percentage of UX experts' opinions with d≤ 2**	**Crisp Values (Z*)**	**Ranking**	**Status**
A1	(0.46,0.70,0.87)	0.136	75%	0.692	7	Accepted
A2	(0.76,0.91,0.98)	0.073	92%	0.896	4	Accepted
A3	(0.80,0.94,1.00)	0.036	92%	0.925	2	Accepted
A4	(0.37,0.59,0.80)	0.136	92%	0.590	9	Accepted
A5	(0.37,0.59,0.80)	0.136	92%	0.590	9	Accepted
A6	(0.86,0.97,1.00)	0.027	92%	0.958	1	Accepted
A7	(0.49,0.71,0.88)	0.136	50%	0.701	6	Accepted
A8	(0.45,0.66,0.85)	0.182	75%	0.660	8	Accepted
A9	(0.80,0.93,0.99)	0.073	83%	0.919	3	Accepted
A10	(0.63,0.79,0.93)	0.182	67%	0.787	5	Accepted
	Total	0.112	81%	n/a	n/a	n/a

4.3.4.2 Findings from the Completed Case Study

The coefficient alpha was **0.718** for the 5-point Likert questionnaire and its variables, which confirms the set threshold for reliability and validity, according to Taber (2018). Likewise, the PCA Eigenvalue value for the questionnaire and its constructs ranged between **1.001** and **4.282**, confirming the convergent validity.

The study utilized SPSS to compute the coefficient values of the regression equation(see **Table 4.9**). The findings show that the constant was significant at p<0.01, and the remaining coefficients were insignificant at p < 0.01 and p<0.05. The regression equation coefficients had R^2 = 0.25, adjusted R^2 = 0.070, F ratio = 1.522, and p = 0.154, and it was insignificance at p< 0.01 and p<0.05.

Table 4.9: Results from Multiple Linear Regression Analysis of the Research Model

Independent variables	Parameters	Coefficient values	t-values
Constant	γ_0	8.244**	9.598**
A1	γ_1	1.176	1.658
A2	γ_2	0.395	0.451
A3	γ_3	-0.342	-0.368
A4	γ_4	-0.851	-1.241
A5	γ_5	0.242	0.557
A6	γ_6	-0.554	-0.737
A7	γ_7	-1.020	-1.851
A8	γ_8	-0.096	-0.221
A9	γ_9	-0.366	-1.290
A10	γ_{10}	1.059	2.353

* Significant at P < 0.05. ** Significant at P < 0.01

In addition, collected data from the completed case study were not approximately normally distributed (see **Table 4.10**). Theoretically, the factor is approximately normally distributed if the z-values of Skewness and Kurtosis are between ±1.96 in a relatively small sample size (n < 50) (Ghasemi & Zahediasl, 2012). For the medium-sized samples (50 < n <300), at an absolute z-value of between ±3.29, the distribution of the factor is normal (Kim, 2013), and finally, for sample size >300, the normality of the data depends on the histograms and the absolute values of skewness and kurtosis.

Table 4.10: Findings from Testing of Data Normality

	Mean	Std. Deviation	Skewness			Kurtosis		
Factor	Statistic	Statistic	Statistic	Std. Error	Z-values	Statistic	Std. Error	Z-values
A1	1.50	0.504	0.000	0.287	0.000	-2.060	0.566	-3.637
A2	1.44	0.500	0.235	0.287	0.820	-2.003	0.566	-3.537
A3	1.40	0.493	0.417	0.287	1.455	-1.880	0.566	-3.321
A4	1.50	0.504	0.000	0.287	0.000	-2.060	0.566	-3.637
S5	1.51	0.503	-0.058	0.287	-0.204	-2.056	0.566	-3.631
A6	1.51	0.503	-0.058	0.287	-0.204	-2.056	0.566	-3.631
A7	1.43	0.498	0.295	0.287	1.029	-1.970	0.566	-3.479
A8	1.50	0.504	0.000	0.287	0.000	-2.060	0.566	-3.637
A9	1.50	0.504	0.000	0.287	0.000	-2.060	0.566	-3.637
A10	1.46	0.502	0.176	0.287	0.613	-2.028	0.566	-3.581

The VIF values range from 1 to infinity. The value 1 denotes a lack of correlation, and values 1 to 5 represent a moderate connection, frequently not severe enough to warrant attention. A value above 5 indicates a potentially high correlation. In this instance, the regression output's coefficient estimates and ρ-values are probably inaccurate. **Table 4.11** shows VIF values for each predictor, and the study interprets them as A9, which is utterly unrelated to other predictors. A5, A7, A8, and A10 have moderate connections with other predictors; however, they are not enough to warrant attention. Finally, the remaining predictors, A1, A2, A3, A4, and A6, have potential degrees of correlation that may impact the formed regression model and warrant further investigation.

Table 4.11: Findings of Data Multicollinearity Testing

Coefficients[a]

Model	Collinearity Statistics	
	Tolerance	VIF
A1	0.146	6.850
A2	0.097	10.314
A3	0.089	11.296
A4	0.156	6.408
A5	0.389	2.571
A6	0.130	7.692
A7	0.247	4.050
A8	0.394	2.539
A9	0.910	1.099
A10	0.365	2.736

a. Dependent Variable: Readiness of the FOSS community to assess its UX maturity

Regarding the FOSS stakeholders' opinions on the relevance of the agreed analytical lenses to determine the readiness of the desktop-based FOSS projects for UX maturity, overall findings show that 47.57% of respondents agree that established analytical lenses influence the preparation for UX maturity of the FOSS, and 52.43% strongly agree (see **Table 4.12**). The FOSS projects foster many factors that make them an unpleasant place for UX designers and practices (Ardito, Buono, Caivano, Costabile, & Lanzilotti, 2014; Wale-Kolade & Nielsen, 2016). For example, FOSS culture does not accept UX professionals into executive teams. 72.2% of respondents strongly agreed, and 27.8% agreed that integrating UX experts into decision-making affects the preparation for UX maturity of desktop-based FOSS projects. The finding is supported by Molich and Winter (2022), who indicated that when decision-makers lack UX knowledge, they base UX design decisions on Highest Paid Personals Opinions (HiPPO), which has been a critical stumbling stone in integrating UX-related activities.

Nevertheless, the lack of unique tools, techniques, and frameworks exclusive for evaluating UX maturity and appropriately accounting for the FOSS-specific dynamics has led to inadequate assessment and planning for UX maturity. The findings show that 72.7% of respondents strongly agreed, and 27.3% agreed that the availability of FOSS-exclusive UX maturity assessment tools affects the preparation of desktop-based FOSS projects for UX maturity. 71.7% of participating stakeholders strongly agreed, and 28.3% agreed that the availability of quantified UX metrics affects the preparation of desktop-based FOSS projects for UX maturity.

Findings show that 72.9% of respondents strongly agreed, and 27.1% agreed that the FOSS community's UX strategic plans affect the preparation of desktop-based FOSS projects for UX maturity, confirming findings from the study of Hussein and Mahmud (2012). 71.7% strongly agreed, and 28.3% agreed that a clear UX role for each stakeholder affects the preparation of desktop-based FOSS projects for UX maturity. Likewise, although many FOSS projects value UX-related activities, 64.8% of respondents strongly agreed, and the other 35.2% agreed that valuing UX-related activities affects the preparation of desktop-based FOSS projects for UX maturity.

The findings further show that 64.2% of respondents strongly agreed, and 35.8% agreed that adopting UCD practices in design affects the preparation of desktop-based FOSS projects for UX maturity. In many FOSS projects, decision-makers do not support UX maturity assessment initiatives such as devoting substantial resources, confirming the findings of Hussein et al. (2015) and İnal and Güner (2016). However, 72.4% strongly agreed, and 27.6% agreed that decision-makers buy-in and support of UX assessment initiatives influence the preparation of desktop-based FOSS projects for UX maturity.

63.7% of respondents strongly agreed, and the other 32.7% agreed that accommodating UX principles in FOSS design affects the preparation of desktop-based FOSS projects for UX maturity. This observation confirms findings from Cajander et al.(2006) and Wale-Kolade and Nielsen (2016). Finally, understanding UX maturity assessment concepts across the FOSS community, projects, and stakeholders is inconsistent. However, 71.7% of respondents strongly agreed, and the other 28.3% agreed that a uniform understanding of UX concepts influences the preparation of desktop-based FOSS projects for UX maturity. The collected results confirm the findings from Inal et al. (2020).

.

Table 4.12: Findings on Analytical Lenses Influencing UX Maturity of Desktop-based FOSS Projects

Id	Analytical Lens	SD	D	N	A	SA
1	The FOSS community's UX strategic plans affect the preparation of desktop-based FOSS projects for UX maturity.	0	0	0	27.1%	72.9%
2	Integrating UX experts into decision-making affects the preparation of desktop-based FOSS projects for UX maturity.	0	0	0	27.8%	72.2%
3	Clear UX roles for each stakeholder affect the preparation of desktop-based FOSS projects for UX maturity.	0	0	0	23.6%	76.4%
4	Valuing UX-related activities affects the preparation of desktop-based FOSS projects for UX maturity.	0	0	0	35.2%	64.8%
5	Adopting UCD practices in design affects the preparation of desktop-based FOSS projects for UX maturity.	0	0	0	35.8%	64.2%
6	Decision-makers' buy-in and support of UX assessment initiatives influence the preparation of desktop-based FOSS projects for UX maturity.	0	0	0	27.6%	72.4%
7	The availability of FOSS-exclusive UX maturity assessment tools affects the preparation of desktop-based FOSS projects for UX maturity.	0	0	0	27.3%	72.7%
8	Accommodating UX principles in FOSS design affects the preparation of desktop-based FOSS projects for UX maturity.	0	0	0	32.7%	67.3%
9	A uniform understanding of UX concepts influences the preparation of desktop-based FOSS projects for UX maturity.	0	0	0	27.1%	72.9%
10	The availability of quantified UX metrics affects the preparation of desktop-based FOSS projects for UX maturity.	0	0	0	28.3%	71.7%
Total		**0%**	**0%**	**0%**	**47.57%**	**52.43%**

4.3.5 Discussion of the Findings

UX maturity is the organizational endeavor (Chapman & Plewes, 2014) that needs the participation of every stakeholder with or without technical backgrounds in programming. However, findings indicate that the FOSS community still has several practices that prevent stakeholders who are not technically savvy from participating. They also limit the integration of other UX/HCI practices and deliver desktop-based FOSS projects with undesirable UX. Generally, desktop-based FOSS projects are not ready for UX maturity, which may be a significant reason for many experienced difficulties, such as poor UX, lost opportunities for improvement, inconsistent UX practices, a lack of collaboration and information sharing, decreased competitiveness, and a limited capacity for community growth.

Knowing how the FOSS community is prepared for UX maturity is still challenging and inconsistent., which limits the adoption of current initiatives to enhance UX maturity. Best practices suggest adopting analytical lenses when analyzing IS, including preparation for UX maturity. The analytical lenses support adequate analysis and provide a broad overview of the organizational characteristics one could investigate (Coral & Bokelmann, 2017; Jokela et al., 2006). Although tons of lenses for analyzing ordinary IS are rigorously discussed by studies, such as Farooq et al.(2014), Iivari et al.(1998), and Namayala et al. (2022), they may not be suitable for studying the preparation of desktop-based FOSS projects for UX maturity. They are not comprehensively tested in the FOSS community to generalize their applicability. Nevertheless, it is still challenging to standardize vocabulary and lenses for preparing desktop-based projects for UX maturity (Monteiro & Maciel, 2020). Therefore, initiatives to develop new analytical lenses may help the FOSS community have self-assessing mechanisms and prepare for UX maturity.

There are a few approaches the body of knowledge can consider when discovering the preparation of the FOSS projects. They may not be comprehensive and tailored for desktop-based FOSS projects. They only provide a few guidelines and criteria that projects may use to assess their preparation for UX maturity. This study has completed comprehensive work to answer RQ1 and RQ2:

In responding to RQ1, **"What analytical lenses are suitable for analyzing the preparation of desktop-based FOSS projects for UX maturity?"** It has created ten (10) analytical lenses using the FDM that engaged eleven experts from several universities. All identified analytical lenses were agreed upon by experts and, therefore, are essential in analyzing desktop-based FOSS projects to understand how they are prepared for UX maturity. According to experts, identified lenses vary in importance, where the lens decision-makers buy-in and support UX assessment initiatives is ranked first, and lenses that value UX-related tasks and adopt UCD practices in design are ranked last.

In responding to RQ2**: "What is the relationship between the identified analytical lenses and the preparation of desktop-based FOSS projects for UX maturity?"** the study has consulted seventy stakeholders whose opinions confirm that the identified analytical lenses relate to the preparation of desktop-based FOSS projects for UX maturity.

The contribution of this study is vivid. For example, by proposing unique analytical factors (lenses) that explain the willingness of the FOSS community to assess its UX maturity, the study may help the FOSS community address the gap of the desktop-based projects lacking experimentally tested lenses and consistent methods that may explain their preparation for UX maturity. The lenses may benefit owners, developers, contributors, and other stakeholders by providing consistent methods to improve desktop-based projects' preparation for UX maturity and help users decide which project has elements for achieving higher levels of UX maturity before adopting. Generally, enhancing preparation for UX maturity may, in turn, enhance user satisfaction and engagement, increase marketplace competitiveness, and improve reputation and credibility. It can also ensure desktop-based projects remain user-centric, accessible, and relevant in a rapidly evolving technological landscape.

4.3.6 Conclusion and Way Forward

4.3.6.1 Conclusion

Like any other software, desktop-based FOSS projects must evaluate their UX maturity effectively and consistently to improve and ascend to higher UX maturity levels that may make them succeed. However, the UX maturity assessment commonly done by adopting UXCMMs alone will not make a difference if prerequisites for UX maturity are unmet. For example, the FOSS community may prepare for UX maturity by engaging the right UX specialists, methods, practices, and other tools. The preparation for UX maturity may contribute to the overall quality of desktop-based FOSS projects and enhance adoption while fostering a more inclusive and user-centric open-source ecosystem. However, FOSS projects still reject several UX maturity prerequisites.

This study acknowledges that UX research is still evolving, particularly in the FOSS community, confirming the findings from Rose et al. (2020). The contemporary pattern supports a growing interest in adopting or innovating UX/HCI practices to suit FOSS projects, and a reference can be made to ongoing UX conferences, workshops, and seminars. However, less is done to realize how the FOSS community is prepared to achieve higher levels of UX maturity. No studies precisely determine how prepared desktop-based projects are for UX maturity. However, the FOSS community has indirectly been undertaking activities that may enrich it with preparation indicators. The interventions are not consistently applied across projects. The saturation point of available solutions has yet to be reached, necessitating additional research and alternative activities before the FOSS community can decide its readiness for UX maturity. This finding confirms the results from the survey of Otchere (2022). As a result, this study has contributed knowledge by conducting empirical research that adopted the FDM to create and rank ten (10) analytical lenses and validated them by performing the case study in real FOSS projects.

Nevertheless, developing countries may not benefit from this study’s findings in the same way as developed countries, mainly European countries, where the foundations for adopting and improving the UX maturity of the FOSS projects have been laid. Developing countries may require different approaches in preparation for UX maturity to be on the same page as European countries. They may be required to first recognize UX and FOSS in Information and Communication Technology (ICT) policy and stop funding software projects whose alternative FOSS projects exist, such as Microsoft Office. To address this gap, we plan to complete a study that may institutionalize the adoption of FOSS and UX/HCI practices in Tanzanian: A Case of Higher Learning Institutions (HEIs).

Like many other studies, this work is susceptible to several limitations. For example, according to Faber and Fonseca (2014), it may have adopted a relatively small sample size. It may have involved parties unfamiliar with preparing desktop-based FOSS projects for UX maturity (Ghazi, Petersen, Reddy, & Nekkanti, 2019). Nevertheless, the study has been based on the author's interpretations of the findings. It may be affected by constraints imposed by the 5-point Likert scale for collecting data in the case study. The adopted sampling technique, purposive sampling, may also challenge the study. Finally, the author may have ignored several other analytical lenses that may significantly contribute to the FOSS community's preparedness.

4.3.6.2 Way Forward

Assessing preparation for UX maturity is not a static endeavor, demanding the FOSS community periodically examine whether further essential criteria or lenses develop. As a result, the study has found two areas that require more investigation to make the FOSS community stay on the right track owing to technical advances. First, future researchers must thoroughly examine analytical lenses to determine the state of readiness of desktop-based FOSS projects for UX maturity. The studies must consider soliciting feedback from all FOSS stakeholders rather than just UX scholars to understand lenses better. Therefore, they may include users, practitioners, experts, and developers. Second, future researchers must perform additional case studies to investigate the relevance of the suggested lenses in assessing the readiness of desktop-based FOSS projects for UX maturity. Increasing the number of respondents and projects may help to generalize the lenses for a larger population.

4.4 Determining Factors Affecting the UX Maturity of Desktop-based Projects: An Empirical Study

4.4.1 Introduction

This subsection addresses the third objective, "**To establish UX maturity influencing factors of the desktop-based FOSS projects.**" It reports the completed empirical study that identifies fundamental factors affecting UX maturity in the

FOSS community and understands how stakeholders perceive and deal with identified factors. The empirical study reviewed the literature as a methodology to establish the initial factors that triggered the brainstorming sessions with twelve UX experts to solicit more profound insights and an understanding of factors. The Fuzzy Delphi method enabled the acquisition of experts' opinions and ranking of thirty-six (36) UX maturity influencing factors. Finally, the thematic analysis was adopted to group them into six themes.

Assessing UX maturity is now a de facto requirement for any organization as it helps identify strengths and weaknesses in implementing UX-related tasks (Chapman & Plewes, 2014; Pernice et al., 2021). However, describing, evaluating, and tracking UX and its maturity is still challenging, particularly in the FOSS community (Hellweger & Wang, 2015; Novák et al., 2023). Zarour and Alharbi (2017) attest that UX is relatively new, and much has not been explored in the concept. UX lacks a globally accepted definition (Coelho, Andrade, & Darin, 2022; Pettersson et al., 2018; Roto et al., 2010; Zarour & Alharbi, 2017) and assumes different explanations based on how it is approached (Roto et al., 2010). Alben (1996), Nielsen and Norman (2017), Hassenzahl and Tractinsky (2006), Sutcliffe (2009), Anthony et al. (2016), ISO 9241-210 (2015), and Hassenzahl (2007) have contributed to developing the classified and robust definitions of UX. However, ISO 9241-210’s (2015) and Hassenzahl and Tractinsky's (2006) definitions are commonly used.

Despite disagreements in definitions, UX is agreed to be a broader concept that goes beyond usability and functionality by including all aspects of the end user's interaction with the company's services and products (Nielsen & Norman, 2017). It significantly affects the success and quality of any software, including FOSS (Luther, Tiberius, & Brem, 2020; Weichert, Quint, & Bartel, 2018). UX is often a result of a robust UX design. Designing a robust UX design that improves organizational efficiency (Castillo-Vergara, Alvarez-Marin, & Placencio-Hidalgo, 2018) demands adopting UX/HCI best practices and assessing the organizational capacity to implement them. However, the FOSS community has limited tools and methods to analyze the ability to implement UX-related activities (Namayala et al., 2022).

Nevertheless, UX assessment tools and methods from other software-developing communities cannot be directly adopted (Terry et al., 2010).

Due to the attached benefits, developing products with desirable UX is now a choice of many businesses (Da Silva et al., 2018; Hokkanen & Väänänen-Vainio-Mattila, 2015; Nguyen-Duc et al., 2021), including the FOSS community. For example, UX may help businesses increase return on investment (ROI), win competitors, engage more users, enhance customer loyalty, and acquire positive brand reputations (Al-Azzawi, 2014).

The design for UX is an organizational endeavor rather than a function or ability of individuals. (Chapman & Plewes, 2014; Cheng & Guo, 2018; Stone et al., 2016). As a result, its success requires the knowledge and dedication of an entire organization (Stone et al., 2016). For example, implementing the design for UX may require organizations to focus on quality, consistent research, well-planned design procedures, resources, and tools (Rukonić et al., 2019). Best practices in UX assessment suggest determining organizational strengths and weaknesses, referred to as UX maturity, to implement UX-related activities for better UX design outcomes. Organizations with higher UX maturity better implement UX-related tasks and are likely to develop products with desirable UX, and the opposite is true (Von-Wangenheim et al., 2010). Nevertheless, UX-matured organizations provide complete user-centric products, well-defined and repeatable UX processes, effective metrics for determining user satisfaction, and well-experienced UX designers capable of delivering UX-focused plans.

Generally, UX maturity holistically communicates the organization's status quo on serving UCD best practices at any time. However, it assumes different definitions in the knowledge body (Vallerand et al., 2017). For example, Pernice et al. (2021) define UX maturity *as the* "measure of an organization's desire and ability to deliver UCD successfully, " while Molich (2021) describes UX maturity as "the ability of an organization or a product team to define and meet UX goals that help to achieve business goals."

UX Capability maturity models (UXCMMs) or frameworks are often adopted to quantify organizational UX maturity attained at a given period (Pernice et al., 2021) and propose a roadmap to improve UX-related activities and advance to higher levels. They show evolutionary plateaus toward developing a mature UX-related process by specifying maturity levels, ranging from no or ad hoc process improvement to integrated continuous improvement (Paulk et al., 1993). Many versions of UXCMMs have lately emerged. For example, a corporate UX maturity by Nielsen (Nielsen, 2006), the 6 levels of UX maturity by Pernice et al. (2021), the UX maturity model by Feijó (2010), the Keikendo UX Maturity Model by Carraro (2014), the UX Maturity Model for e-commerce websites by Anchahua et al. (2018), the Corporate User-Experience Maturity Model by Van Tyne (2009) and a UX Maturity Model: Effective Introduction of UX into an organization by Chapman and Plewes (2014). Each UXCMM aspires to assess organizational UX maturity with novelty. However, Sauro et al. (2017) claimed they are just sales gimmicks because they are architecturally indistinguishable and tentatively made from the same concepts and ingredients.

Although many editions of generically developed UXCMMs exist and are assumed to apply in any software community and context, it is still unknown if we can use them in the FOSS community because they have not been rigorously tested in actual FOSS projects (Lacerda & von Wangenheim, 2018; Namayala et al., 2022; Raza et al., 2012a). Namayala et al. (2022) showed that no UXCMM was created for the community, implying they may not have the FOSS community's specific UX maturity influencing factors. Nevertheless, many UXCMMs have failed to define or are not clear on what makes a UX-matured system (Buis et al., 2023; Kocaballi et al., 2019; Namayala et al., 2022), imposing challenges of not containing the proper contents or factors influencing UXCMMs. It is also uncertain if they measure UX maturity correctly (Sauro et al., 2017). Finally, the development process of many UXCMMs did not adhere to formalized development methods or models (Monteiro & Maciel, 2020; Otto et al., 2020). Thus, they are not empirically developed and often accommodate relatively few UX maturity influencing factors originating from the authors' years of working experience (Lacerda & von Wangenheim, 2018; Namayala et al., 2022; Rukonić et al., 2019; Sauro et al., 2017).

This study, therefore, aims to offer novel contributions to knowledge by empirically identifying and ranking UX maturity influencing factors exclusive to the FOSS community. It has responded to the research questions: -

RQ1 How do the FOSS community's stakeholders understand UX maturity and linked terms?

RQ2 What UX maturity influencing factors affect the UX maturity of the FOSS community?

The specified UX maturity influencing factors may help researchers, academicians, practitioners, and FOSS stakeholders develop the correct UXCMMs for the FOSS community. Nevertheless, areas suggested for future studies may inspire future researchers to engage in the topic and help the FOSS community assess its UX maturity precisely and practically.

The rest of this paper is organized as follows. The second section explains the study's background information. The third section presents the adopted research methodology, the fourth section presents the findings, and the fifth section discusses the results. Finally, section six is devoted to the conclusion and way forward.

4.4.2 The Background Information

4.4.2.1 Free and Open-Source Software (FOSS)

FOSS is software distributed with licenses that permit programmers to examine and modify the source code in desired ways for bug repair, enhancing functionality, and customizing to meet their demands (Amega-Selorm & Awotwi, 2010; Nagy et al., 2012; Pickett, 2019). According to the Free Software Foundation (FSF, 2011) and Open Source Initiative (2010), FOSS has a Low Total Cost of Ownership (TCO) compared to proprietary software and is therefore considered an alternative solution (Abramova et al., 2016; Bahamdain, 2015; Biswas, 2010; Bouras et al., 2013; Petrinja & Succi, 2012; Shekgola et al., 2021). The word "free" in FOSS refers to the liberty to use the software, but it may not necessarily be at an accessible cost. However, FOSS must certify four mandatory user privileges that allow the use of the software for any purpose, modify source codes to fit needs, redistribute the software without authors’ consent, and improve the source codes (Engard, 2010; Free Software Foundation, 2012; Srinivasa, 2017; Stallman, 2002).

The FOSS projects supply chain differs from other software-developing communities (Kuwata et al., 2014). For example, they are often built and maintained by the entire FOSS community (Haaland et al., 2010) in dynamic environments (Méhat et al., 2015). FOSS community is self-governed, with no executives overseeing the whole community (Engard, 2010). Its structure is highly complex (Scacchi, Feller, Fitzgerald, Hissam, & Lakhani, 2006) and does not formally describe activities and their order of implementation (Senyard & Michlmayr, 2004). As a result, it may have unique UX challenges, such as different UX maturity influencing factors and demand special treatment. However, the community still uses generically developed UX maturity influencing factors, which may not be a replica of the actual image, resulting in incorrect outcomes. This notion is reinforced by recommendations by Terry et al. (2010), which state that generically designed UXCMMs and contained ingredients in their current state should not be directly adopted because they will likely overlook FOSS context-specific variables. After all, they have various drawbacks that require particular adaptations.

Generally, the FOSS community lacks empirically established studies investigating what UX maturity influencing factors are appropriate for FOSS projects. As a result, evaluating UX and its maturity has remained rather complex, subjective, and slow (Novák et al., 2023). However, having UX maturity influencing factors pertinent to FOSS projects is necessary (Petrinja & Succi, 2012). Among other benefits, understanding UX maturity influencing factors exclusive to the FOSS community may bring several advantages to the FOSS community, such as improved user-centricity, enhanced usability, increased adoption, and community engagement. In addition to this, they may contribute to FOSS projects' overall success and sustainability by creating FOSS projects that meet user needs and expectations and attracting and retaining a loyal user base.

Based on the above annotations, The FOSS community requires either wholly novel ways to uncover UX maturity influencing factors and new UXCMMs or contextualization of existing UX/HCI methods, tools, and theories to meet its dynamics (Bornoe & Stage, 2014; Dawood et al., 2019; Raza et al., 2013; Terry et al., 2010). Petrinja and Succi's (2012) observations confirm that generically developed UXCMMs are irrelevant to the FOSS community, sealing evidence for a requirement to examine the relevancy of UX maturity influencing factors included in current UXCMMs. These findings are confirmed by systematic literature reviews and mapping studies of Namayala et al. (2022) and Choma et al. (2022).

4.4.2.2 UX Capability Maturity Models in the FOSS Community

In the FOSS community, evaluating UX, including capability and maturity, is still thought-provoking (K. J. Stol & Ali Babar, 2010). Nevertheless, traditional quality models are extremely hard to adopt (Adewumi et al., 2013). However, the FOSS community does little to develop tools like UXCMMs and other quality models specific to its projects (Namayala et al., 2022). Present UXCMMs and other quality models do not address the FOSS community's unique characteristics (Adewumi et al., 2016; Namayala et al., 2022), leaving gaps that demand the development of new UXCMMs and other quality models (Haaland et al., 2010).

Precise evaluation of FOSS UX maturity is imperative, just as it has been in other software-developing communities. Thus, the FOSS community must adapt or develop UXCMMs to meet the needs of its user base intensified beyond technically equipped (Dawood et al., 2019; Llerena, Rodriguez, Castro, & Acuña, 2019; Llerena, Rodriguez, Gomez-Abajo, Castro, & Acuña, 2018; Raza et al., 2012a). The community must adopt and employ accepted development procedures, such as planned releases, peer review of modifications, submission and review of changes, and supply of a test suite (Adewumi et al., 2016). However, as confirmed by an exhausted literature review, the FOSS community often adopts ad hoc interventions, and its structure does not smoothly accept valuable contributions from UX professionals (Bach et al., 2009; Nichols & Twidale, 2005). As a result, practiced interventions may be incorrect, inconsistent, and difficult to replicate across projects.

Like several other capability maturity models (CMMs), many current UXCMMs have adopted structures from the ancestors of CMMs, including the Capability Maturity Model (CMM) by the Software Engineering Institute (2006), Crosby's Quality Management Maturity Grid (QMMG) by Humphrey and Sweet (1987) and

Total Quality Management (TQM) by Lakhe and Mohanty (1994). However, UXCMMs exclusive to the FOSS community are still limited (Namayala et al., 2022). Many available efforts establish usability capability maturity models (UCMMs) and other quality models for selecting high-quality FOSS projects or components to include in other applications (Adewumi et al., 2016). For example, the Open-source Usability Maturity Model (OS-UMM) by Raza et al. (2012a), the Open Source Maturity Model (OSMM) by Golden (2008), the CapGemini Open Source Maturity Model (C-OSMM) by Dujinhouwer and Widdows (2003), the Qualification and Selection of Open Source software (QSOS) by Semeteys (2008), the Open Business Readiness Rating (OpenBRR) by Taibi et al. (2007), and Software Quality Observatory for Open Source Software (SQO-OSS) by Samoladas et al. (2008). However, they are not highly adopted (Li et al., 2009; Petrinja & Succi, 2012).

4.4.2.3 Factors Affecting UX Maturity

Formalizing UX maturity influencing factors is critical to uniformly assess UX maturity across FOSS projects (Petrinja & Succi, 2012). They must be based on organizational context (Traynor, 2022). However, experimentally established UX maturity influencing factors are missing (Cheng & Guo, 2018; Choma et al., 2022; Raza et al., 2012b). Those currently used pay little attention to UX standards (Traynor, 2022) and do not adequately explain the crucial UX maturity influencing factors, particularly in the FOSS community (Petrinja & Succi, 2012).

Little has been done to establish UX maturity influencing factors (Osinusi, 2020), particularly in the FOSS community. However, multiple initiatives to develop usability maturity influencing factors exist. For example, Raza et al. (2012a) systematically and experimentally conducted four different empirical studies, Raza et al. (2013), Raza and Capretz (2010), Raza et al.(2012b), and Raza et al. (2010) that sought perceptions from users, developers, industry and contributors to develop the Open Source Usability Maturity Model (OS-UMM). However, Raza et al. (2012a) did not test the model to determine if it fits in assessing the UX of actual FOSS projects, insinuating that used factors do not guarantee their relevancy in creating FOSS UXCMMs (Namayala et al., 2022).

Several available UX maturity influencing factors that currently form the generically developed UXCMMs are the derivatives of professional working experience (Sauro et al., 2017) lacking formalized techniques, and others are derived from organizational behaviors and standards such as ISO 33020 (Traynor, 2022). To a large extent, the FOSS community may continue delivering projects with poor UX if it does not formalize its methods for establishing UX maturity influencing factors since unsystematically formulated UX maturity influencing factors may often fail to account for organizational-specific dynamics and contexts.

4.4.2.4 Related Studies

This study found limited to no studies reporting the empirical development of UX maturity influencing factors, notably in the FOSS community. Encountered UX maturity influencing factors lack development methodological rigor and do not demonstrate the context in which they can be more effective; for example, the UX maturity influencing factors used in models developed by Rukonić et al. (2019), Pernice et al.(2021), and Chapman and Plewes (2014). Nevertheless, of all the studies we examined, Pernice et al. (2021) from the Nielsen and Norman Group have thoroughly discussed and classified UX maturity influencing factors today regarded as de facto. They grouped them into four main clusters: a strategy that incorporates user experience leadership, planning, and resource prioritization; a culture that fosters the development of user experience careers and practitioners; a process that demonstrates the systematic application of user experience research and design methods; and, finally, the outcome that consciously defines the effects of UX works.

4.4.2.5 Limitations of the current studies

Using generic UX maturity influencing factors derived from the owners' years of working experience for every company may not be the best option. It is an unrealistic endeavor since every organization has unique dynamics affected by its goals, strategy, environment, technology, and size and requires personalized solutions (Ahmady, Mehrpour, & Nikooravesh, 2016). Therefore, we cannot use the same UX maturity influencing factors in every community, including FOSS, because they may skip specific issues. Generally, there is nothing like a general solution, particularly in the UX assessment, due to the subjectivity, holistic, dynamic, and situational nature of UX (Bosley, 2013; Hassenzahl, 2010; Namayala et al., 2022).

Nevertheless, to the authors' knowledge accumulated from the completed literature review, the FOSS community still lacks comprehensive studies on determining UX maturity influencing factors, which may be regarded as baselines. As a result, this is the first analysis, and the data gained supplements previous studies of UX maturity influencing factors in the FOSS community.

4.4.3 The Research Methodology

The study adopted a mixed research design approach involving quantitative and qualitative techniques and was conducted online with the help of Internet technologies. It examined sixty-two (62) active FOSS projects hosted in FOSS source code hosting facilities. The projects' inclusion/exclusion criteria included projects regularly updated with an activity level of 90% and above, weekly downloads count of 100 and above, active mailing lists, and other discussion forums. The activity levels were calculated from projects' user ratings. For example, a user rating of 4.9 out of 5 provides an activity level of 98%. Examined projects were grouped into seven application areas: accounting (11), office (6), project management (9), games and entertainment (9), knowledge management (6), development (5), and scientific and engineering (16).

The authors completed different phases to collect and analyze data and report the findings. In the first step, the study reviewed current literature and ISO 9241-11:2018 by the International Organization for Standardization (ISO) (2018) to generate a tentative list of UX maturity influencing factors for stakeholder interviews and expert discussions. In the second phase, the authors interviewed fifteen (15) FOSS stakeholders from the selected projects who actively discussed UX issues in projects' mailing lists or other discussion forums (see adopted questionnaire in **Appendix I**). Personalized e-mails were sent to selected stakeholders' e-mails extracted from discussion forums explaining the study's purpose and seeking participation consent. The recruitment process of stakeholders took place between mid-July 2021 and November 2022, with an initial response rate of 32.29% (31/96). Finally, fifteen (15) FOSS stakeholders agreed to participate and signed consent forms.

Nevertheless, the authors explored stakeholders' understanding of UX maturity-related concepts in the same phase. They adopted the thematic analysis method that Braun and Clarke (2006) proposed to identify themes from the collected data to streamline the interpretation and discussion of the results. The method contained six steps: (1) Familiarizing with data, (2) Generating initial codes, (3) Searching for themes, (4) Reviewing themes, (5) Defining and naming themes, and (6) Producing the report.

In the third phase, the study adopted the Fuzzy Delphi Method (FDM) for screening to obtain experts' consensus on UX maturity influencing factors established from a completed literature review, stakeholder interviews, and brainstorming with UX experts. The FDM finally assisted in the defuzzification and ranking of UX maturity influencing factors. The execution of the FDM usually begins with formulating triangular fuzzy numbers (TFNs) that use binary ratings ranging between 0 and 1 inclusively (see **Figure 4.3**).

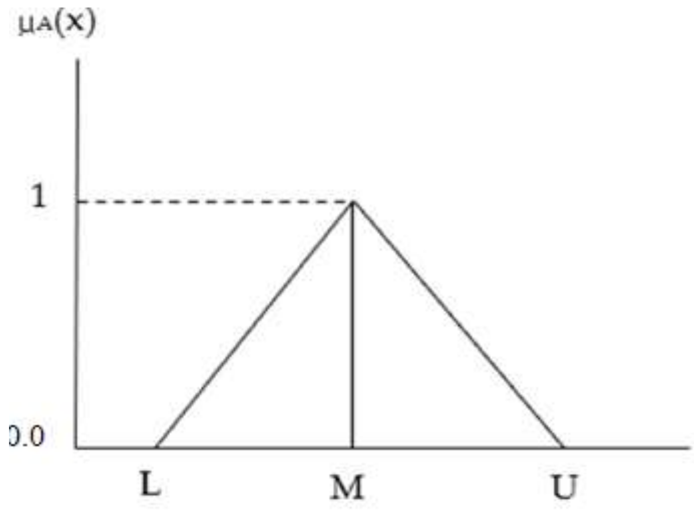

Figure 4.3: Triangular Fuzzy Number
SOURCE: Adapted from Lee et al.(2021)

TFNs often create fuzzy scales similar to Likert scales that translate linguistic variables into fuzzy numbers (see **Table 4.13**). Fuzzy numbers are a particular type of fuzzy sets that are normalized, convex, and have bounded sets. Unlike classical logic, each number in fuzzy logic has an approximate value. As a result, using fuzzy sets aligns more with human linguistic and occasionally ambiguous descriptions (Habibi et al., 2015).

Table 4.13: Triangular Fuzzy Scoring and 7-Points Likert Scale

Likert Scale Scoring	Linguistic variable	Fuzzy Scale scoring
1	Extremely unimportant	(0.00, 0.00, 0.10)
2	Very unimportant	(0.00, 0.10, 0.30)
3	Unimportant	(0.10, 0.30, 0.50)
4	Moderately important	(0.30, 0.50, 0.75)
5	Important	(0.50, 0.75, 0.90)
6	Very important	(0.75, 0.90, 1.00)
7	Extremely important	(0.90, 1.00, 1.00)

SOURCE: Habibi et al. (2015)

TFNs are often displayed with three real numbers constructed using the formula $\mu_A(x) = (L, M, U)$. L is the left threshold denoting minimum values of $\mu_A(x)$, U is the suitable threshold denoting the maximum value of $\mu_A(x)$, and M is the median value denoting the most likely value of a fuzzy number. The total values (T) of the TFN's defuzzification are often established according to the fuzzy set, deffuzify, proposed by Chen and Hwang (1992). The equation below explains the TFNs' membership function (MF).

$$MF(x) = \begin{cases} \frac{x-L}{M-L} L < x < M \\ \frac{U-x}{U-M} M < x < U \\ 0 otherwise \end{cases}$$

The FDM is a more advanced technique than the traditional Delphi Method (DM) developed by Dalkey and Helmer (1963) since it combines the classical Delphi Method (DM) and Fuzzy Theories (Kadir, Abdullah, & Mustapha, 2019; Saffie, Shukor, & Rasmani, 2016). It quickly determines the distance between the consensus levels among experts (Ishikawa et al., 1993) and saves time, effort, and cost in running questionnaires due to the extreme reduction in survey rounds (Jarana- Díaz, Romero- Martín, Ponce- Blandón, & Jiménez- Picón, 2021; Lee et al., 2021). This argument is supported by the single-round study performed by Habibi et al. (2015). The distance between Fuzzy numbers or threshold value (d) is calculated using the formula

$$d(m,n)=\sqrt{\frac{1}{3}\left[(m_1-M_1)^2+(m_2-M_2)^2+(m_3-M_3)^2\right]}$$

where, **(M_1, M_2, M_3)** are the fuzzy values of each item and **(m1, m2, m3)** are the corresponding fuzzy average values. According to Chen (2000) and Cheng and Lin (2002), if each screened variable's threshold value (d) does not exceed 0.2, the variables fulfill one condition among the three conditions for reaching experts' agreement. The second requirement is that the total percentages of items with d ≤ 2 be 75% or above (Chu & Hwang, 2008; Murry & Hammons, 1995).

Several investigations, such as Alharbi and Khalifa (2021), have offered efforts to create FDM variants. However, stakeholders have not agreed on the variations (2015). According to Alghawli et al.(2022) and Habibi et al. (2015), the FDM in its classic form provides sophisticated ways of handling ambiguity resulting from experts' aptitude and subjectivity in getting the UX maturity influencing factors that could later be used to construct UXCMM tailored to the FOSS community.

The FDM algorithm for screening consists of four main stages (see **Table 4.14**), and each step is tied to a specific goal(s) that must be accomplished (Habibi et al., 2015).

Table 4.14: Stages of the Fuzzy Delphi Method Algorithm for Screening

Sn.	Step	Goal(s)
1	Use or develop common fuzzy spectra.	Identify/develop an appropriate fuzzy spectrum for the fuzzification of linguistic expressions.
2	Collect and fuzzify experts‘ opinions	Fuzzy aggregation of experts' opinions
3	Defuzzification	Defuzzification
4	Testing the acceptability of each item	The authors decided which items to accept or reject based on the comparison between the threshold of 0.6 the study opted for and the items' crisp numbers.

In the first step, an appropriate fuzzy spectrum is selected or developed, for example, a triangular fuzzy spectrum of three, five, or seven fuzzy scales. Once the fuzzy spectrum is selected, execution goes to the second step, where different methods may be applied. However, if experts' opinions in the first step are displayed as triangular, calculating the fuzzy average numbers is the most viable option to complete this step. The formula shown below and explained by Manakandan et al. (2017) has been used in this study.

$$\mu(x)AVE = \frac{\sum_{i=1}^{n} L_i}{n}, \frac{\sum_{i=1}^{n} M_i}{n}, \frac{\sum_{i=1}^{n} U_i}{n}$$

However, other experimental formulas for aggregating experts' opinions may also be adopted. For example,

$$\mu(x)AGGR = \left(\min\{L\}, \left\{ \frac{\sum_{i=1}^{n} M_i}{n} \right\}, \max\{U\} \right)$$ and

$$\mu(x)AGGR = \left(\min\{L\}, \prod\{M\}, \max\{U\} \right)$$

$$\mu_A(x)AGGR = (min\{L\}, \prod\{M\}, max\{U\}).$$

Studies may also adopt different methods to accomplish the third stage. This step turns the final fuzzy values into crisp values (Z^*) to check if the experts' agreement's third and final condition is fulfilled. However, the study has used a method of defuzzification that uses formulas.

$$x_M^1 = \left(\frac{l+m+u}{3}\right), x_M^2 = \left(\frac{l+2m+u}{4}\right), x_M^3 = \left(\frac{l+4m+u}{6}\right) \text{where}$$

$$l = \frac{\sum L}{n}, m = \frac{\sum M}{n}, u = \frac{\sum U}{n}$$

$$Z^* = \max\left(x_M^1, x_M^2, x_M^3\right)$$

The calculated crisp values are then compared to the α-cut value, which is the median value for TFNs “0” and “1”, calculated as α-cut = (0+1)/2 = 0.5. If the value of Z^* is less than the α-cut value, the item is rejected because experts dispute it; nevertheless, if the value of Z^* exceeds the α-cut value, the item is accepted because experts have agreed (Mohamed Yusoff et al., 2021). Nevertheless, the Center Of Gravity (COG), the Center Of Area (COA), and the mean of maxima are other methods that might be adopted for the defuzzification step. However, the most straightforward and preferred method is the COA calculated using the formula.

$$\mathbf{DF_{ij}} = \frac{[(\mathbf{U_{ij}} - \mathbf{L_{ij}}) + (\mathbf{M_{ij}} - \mathbf{L_{ij}})]}{\mathbf{3} + \mathbf{L_{ij}}}$$

In the fourth and final step, the acceptability of all factors is tested. A threshold is first decided based on the researcher’s opinion; in this study, “0.6” was preferred to denote low-ranked items. The threshold (0.6) is critical, particularly when executing an additional FDM round is not an option. It helps eliminate or retain items based on the crisp values of defuzzification of aggregated expert opinions. When the item’s crisp value exceeds the threshold, the item is accepted, and if it falls below the threshold, the item is rejected (Alghawli et al., 2022).

Likewise, Okoli and Powlowski’s (2004) principles were used to establish the final set of UX assessment experts who were purposively selected. Their inclusion/exclusion criteria demanded those who have published, read, or reviewed scholarly publications on UX/usability maturity assessment. They must, however, actively participate in evaluating FOSS UX/usability maturity or have seen the progress of FOSS projects' UX maturity evaluation over the last ten years. Authors sent personalized e-mails to UX experts whose initial list of names and contacts was proposed by both authors. The list was continually updated using snowballing, and the recruitment process took place between mid-July 2021 and November 2022, with an initial response rate of 40.20% (41/102). Finally, twelve (12) UX experts agreed to participate and signed consent forms (see **Table 4.15**).

Table 4.15: UX Evaluation Experts' Demographic Information

Gender	Experience (years)	Age	Country	Title	Research Area
Male	18	61	Tanzania	Professor	eLearning, HCI, and information systems success
Male	10	67	Finland	Professor	UX and UX design
Male	11	58	Germany	Professor	Human-centered artificial intelligence, UX, HCI, ubiquitous computing, and mobile interaction
Male	22	55	USA	Professor	UX, HCI, Interaction Design, and Persuasive Technology
Female	25	55	Finland	Professor	UX design, UX, and human-automation interaction
Male	19	49	China	Associate Professor	HCI, Usability, UX, conversational Agents (Chatbots), and affective Computing
Male	16	48	France	Associate Professor	Cognitive science, emotion, HCI, and computational modeling
Male	23	65	Brazil	Professor	HCI and accessibility
Male	15	54	Italy	Professor	E-learning, Accessibility, HCI, UX, and Digital Health
Male	20	44	Spain	Professor	UX and social connectedness
Male	11	42	Tanzania	Lecturer	FOSS, HCI, and ICT4D
Female	10	47	Canada	Professor	eHealth, UCD, UX, and Requirements Engineering

4.4.4 Results

4.4.4.1 Initial FOSS UX Maturity Influencing Factors

After completing the first research phase, the study discovered twenty-one (21) UX maturity influencing factors that may fit in the FOSS community (see **Table 4.16**).

Table 4.16: Initial List of UX Maturity Affecting Factors Appealed to Apply in FOSS

Sn.	Proposed FOSS UX maturity influencing factors	Source(s)
1	Allocating adequate financial resources for UX-related works	Van Tyne (2009), MacDonald et al.(2022), Chapman and Plewes(2014), Rukonić et al.(2019), Peres et al.(2014), Sauro et al.(2017), and Young et al.(2020)
2	Dedicate UX staff with specialized UX skills.	Van Tyne (2009), MacDonald et al.(2022), Chapman and Plewes(2014), Rukonić et al.(2019), Peres et al.(2014), Sauro et al.(2017), and Young et al.(2020)
3	UX leadership	MacDonald et al.(2022), Chapman and Plewes(2014), Peres et al.(2014), Sauro et al.(2017), and Young et al.(2020).
4	Include UX professionals as decision-makers	Raza et al.(2012a) and Terry et al. (2010)
5	Adopt UCD techniques	Young et al.(2020) and Rukonić et al.(2019).
6	Consider UX as a professional	Rukonić et al.(2019), Raza et al.(2012a)**,** and Terry et al. (2010)
7	Buy-in of UX ideas	Rukonić et al.(2019), Peres et al.(2014), Sauro et al.(2017), and Young et al.(2020).
8	Clear UX maturity roles for each stakeholder	Chapman and Plewes(2014), and Rukonić et al.(2019).
9	Timing to UX techniques in the development process	Chapman and Plewes(2014) and Peres et al.(2014)
10	Regular and systematic use of	Van Tyne (2009), Pernice et al. (2021),

Sn.	Proposed FOSS UX maturity influencing factors	Source(s)
	UX research methods	Anchahua et al.(2018)and Rukonić et al.(2019),
11	Formalize UX design standards.	MacDonald et al.(2022), Chapman and Plewes(2014), Peres et al.(2014), Sauro et al.(2017), and Young et al.(2020).
12	Participation of stakeholders beyond the UX team during UX design	Young et al.(2020), Rukonić et al.(2019) and Pernice et al (2021)
13	Regular UX training	Peres et al.(2014) and MacDonald et al.(2022).
14	Esteeming UX skill sets similar to technical skills.	Raza et al. (2012a), Peres et al.(2014), and Sauro et al.(2017).
15	Organizational flexibility	Sauro et al.(2017) and Rukonić et al.(2019)
16	UCD goals predicting UX metrics.	Rukonić et al.(2019) and Jokela and Abrahamsson (2000)
17	Regular review and update of UX metrics	Jokela and Abrahamsson (2000), Pernice et al. (2021), and Chapman and Plewes(2014).
18	UX impacts all project stakeholders.	Pernice et al. (2021), Rukonić et al.(2019) and MacDonald et al.(2022).
19	constantly monitoring UX quality	Namayala et al. (2022), Pernice et al. (2021) and Rukonić et al.(2019),
20	Users' needs	Peres et al.(2014), Chapman and Plewes(2014), and Van Tyne (2009).
21	Consistently tracking UX-related tasks.	Van Tyne (2009)and Sauro et al.(2017).

4.4.4.2 Reliability and Validity of the Study

All instructions and questionnaires adopted by the study were pre-tested with the author, the co-authors in the published papers, and two independent postgraduate students who aided in various research-related activities. As a result of the offered recommendations, minor tool adjustments were made. Involved stakeholders were selected from FOSS projects with activity levels of at least 90% calculated from user ratings. Nevertheless, predetermined standards for quality and inclusion/exclusion criteria decided which document and scholars to include or exclude in the survey. Finally, several initiatives were made to secure communications with respondents to ensure the integrity of the collected data.

4.4.4.3 FOSS Stakeholders' Knowledge of UX Maturity and Linked Concepts in the FOSS Community

In response to research question one**," How do the FOSS community's stakeholders understand UX maturity and linked terms?"** The findings show that among fifteen (15) FOSS stakeholders, 26.67% were UX designers, 13.33% were product designers, 6.67% were visual designers, 26.67% were UX researchers, 6.67% were content strategists, and 6.67% emanated from other specialties. Nevertheless, 13.33% of selected respondents did not respond to the question regarding their profession. Regarding the respondents' working experience, findings show that 26.67% worked in the FOSS community between zero and five years, 46.67% between five and ten years, 20% between ten and fifteen years, and 6.67% had over fifteen years of working experience.

Since there has never been a consistent method for analyzing interview data (Roulston, 2014), this study performed a content analysis of the interview data using NVivo 12. The findings identified four themes: UX experts' understanding of UX maturity, how stakeholders' descriptions relate to what Nielsen and Norman Group already described, acquired experience in assessing UX maturity, and factors influencing FOSS UX maturity. The presentation and discussion style of the findings mixed deductive and inductive strategies. It adopted the top-down approach that started with developing research theory and progressed to tentative hypothesis testing, observing data patterns and ended with a conclusion by observations. The deductive approach is different from the inductive approach. The former tries to develop a theory, while the latter aims to test an existing idea (Symons, 2019).

Findings from the theme, "UX experts' understanding of UX maturity," show that although each stakeholder has a unique understanding of UX maturity, provided explanations indicate common features. For example, one UX professional said, "….*UX maturity usually describes how an organization transforms from principle-based to UCD.*" Another one said,*"…UX maturity involves users at every stage of organizational product development processes instead of only getting business and technology working*." We find similarities between the provided definitions in these two sampled descriptions. Nevertheless, stakeholders often used similar words when defining UX maturity and related ideas (see **Figure 4.4)**

Figure 4.4: Common Terms Used to Define UX Maturity

Findings from the second theme, "How stakeholders' descriptions relate to what Nielsen and Norman Group already described," demonstrate how each stakeholder remarked on the similarities and differences between the given descriptions and Nielsen and the Norman Group's well-known explanation. For example, one stakeholder said, "…*My description and what the Nielsen and Norman group said do not differ noticeably."* The second said, *"….Every academic has adopted the definition the Nielsen and Norman group offers, and I am likewise at ease."* One more stakeholder communicated, "…*Although my definition and the Nielsen and Norman group definition contextually present the same thing, however, the two definitions differ significantly*."

Findings from the third theme, "The acquired experience in assessing UX maturity," show that FOSS stakeholders learn and develop new techniques from experience acquired by assessing UX maturity. However, acquired experiences vary among stakeholders based on their specialties. For example, one stakeholder said, "….*Yes, I have been engaged in UX maturity-related tasks for over six years, particularly designing evaluations that utilize currently accessible UX maturity models. Through years of utilizing the six levels of the UX maturity model, I have improved my knowledge of UX maturity assessment. However, assessing UX maturity has not always been simple, and it has occasionally been difficult to distinguish between levels*." A different stakeholder said, *"….I have more than ten years of working experience and have been directly or indirectly engaged in assessing UX maturity; however, I have encountered several difficulties. For example, many companies use the only available UX expert to produce logos, banners, or fliers and believe they are their responsibility. Another challenging aspect is the misunderstanding of UX functions. Finally, getting resources for UX-related tasks is challenging in several FOSS projects."* One more stakeholder said,*"…. Over ten years, I have gathered several UX and Customer Experience (CX) models. However, the task has been disappointing because many available models are designed for UX teams to assist them in monitoring development progress. They are expensive, use complex languages, and very few are understandable enough for those not knowledgeable in UX."* Finally, one more stakeholder said, *"….Surprisingly, little research is present on the elements that make up a 'mature' UX. Moreover, it has been challenging to explain the link between UX maturity and organizational success and what is the optimal maturity level for a given organization."*

Findings from the final theme, "The factors influencing FOSS UX maturity," show that the FOSS stakeholders have enough knowledge of what factors may positively impact the UX maturity of FOSS projects. For example, one stakeholder said, *"….User-centred design, which I believe is the driving force behind UX maturity, is influenced by many factors, including UX teams, UX research, a defined UX strategy, organization actions aligning with user expectations, and quantitative UX metrics. Others may include adopting artificial intelligence to examine the enormous amount of users' feedback, quality of user interfaces, and simplicity in submitting* *users' complaints."* Another stakeholder said, *"... As a result of several empirical tests, various unique factors may affect the UX maturity of the FOSS community, including meeting users' requirements, sophisticated ways of collecting and handling users' feedback, UX learning, adoption of UCD methodology, projects' understandability, learnability, operationality, attractiveness, simplicity of bug reporting procedures, UX/usability testing, and proper documentation."* Another stakeholder said, "*….Numerous elements influence UX maturity. including budgets, improved tools, thoughtful design methodology, high-quality user research, and continuity."* Finally, one stakeholder remarked*, "….UX maturity is affected by several key indicators, such as ease of use, speed of use, learnability, consistency, content, accessibility, and flexibility. Others may include aesthetics, recovery from errors, help, brand recall, persuasiveness, differentiation, and the greater good."*

4.4.4.4 Applying Fuzzy Delphi Method to Analysis of UX Experts' Agreement on Established UXMIF

The thirty-six most important UX maturity influencing factors to the FOSS community were established after the literature review, stakeholders' interviews, and consultation with UX experts (see **Table 4.17**). The factors were grouped into six themes: the FOSS UX culture, the FOSS UX processes, the FOSS UX outcomes, the FOSS UX strategy, FOSS-adopted technologies, and FOSS usability. The factors were given to the UX experts in response to research question two**," What UX maturity influencing factors affect the UX maturity of the FOSS community?"**

The FOSS UX culture describes a unique software design and development approach emphasizing collaboration, user-centered design, and openness. The FOSS UX processes describe methods and activities UX designers use to understand and improve the UX of users interacting with a product or service. They are typically iterative and involve activities designed to uncover user needs, preferences, and behaviors to develop and test solutions that meet those needs. The FOSS UX outcomes refer to the design process results and how they impact the overall UX of a product or service. They measure the success of a UX design based on how well it meets the needs and expectations of users and how it achieves the goals of the product or service. The FOSS UX strategy provides a framework for making design decisions and evaluating the design's success over time. It involves defining the target audience, understanding their needs and behaviors, and creating a plan for delivering an optimal UX. FOSS-adopted technologies refer to technologies that enhance UX by making products more engaging, interactive, and effective. They may include artificial intelligence (AI), machine learning (ML), virtual reality (VR), augmented reality (AR), the Internet of Things (IoT), motion design, and mobile and responsive design. Finally, FOSS usability refers to the ease with which users can interact with a product or service to achieve their goals. Usability is a crucial aspect of UX design and involves designing intuitive, efficient, and effective products.

Table 4.17: Items for Free and Open-Source Software User Experience Maturity Influencing Factors

Id.	Items
Usability (U)	
U1	Documentations and support
U2	Ease of use
U3	Speed of use
U4	learnability
U5	Users' feedback
U6	Bugs reporting procedure
U7	UX/usability testing and quality assurance
U8	Meeting users' requirements
U9	User-centered Design Methodology
U10	Operability
U11	Accessibility
U12	Attractiveness
The FOSS Adopted Technologies (T)	
T1	Artificial Intelligence (AI) (
T2	Adopted development methods
T3	Technical infrastructures, such as software development tools, libraries, and frameworks
The FOSS UX Strategy (S)	
S1	UX leadership

S2	Planning of UX-related activities
S3	Allocating adequate financial resources for UX-related works
S4	Dedicating UX staff with specialized UX skills
The FOSS UX Culture (C)	
C1	Developers' UX skill
C2	Regular training on UX-related activities
C3	Esteeming UX skill sets similar to technical skills.
C4	The organizational flexibility
C5	Considering UX as a professional
C6	Executive understanding and buy-in of UX ideas
C7	Clear UX maturity roles for each stakeholder
C8	UX professionals are part of the decision-makers
The FOSS UX Process integration (P)	
P1	Regular and systematic use of UX research methods.
P2	Formalize standards for UX design.
P3	Timing to include UX methods, practices, principles, and tools
P4	Consistently tracking UX-related tasks.
P5	Monitor UX quality constantly.
The FOSS UX Outcome (O)	
O1	UCD goals predicting UX metrics
O2	Regular review/update of UX metrics
O3	Early engagement of all stakeholders
O4	Number of errors made by users

Collected data were analyzed as prescribed in the algorithm for implementing the FDM for screening (see **Table 4.14**). Based on the findings from the completed FDM analysis (see **Table 4.18**), all threshold values are less than or equal to 0.2, and according to Cheng and Lin (2002), experts' consensus has been reached. The total percentage of agreement is above 75%, and all defuzzification crisp values exceed the α-cut = 0.5. All items are ranked based on priority, where users' feedback and technologies adopted in developing FOSS projects are ranked first, while the speed of use and learnability are ranked last. However, the importance and impacts of UX maturity influencing factors may vary between projects and user base, making definitive ranking highly challenging.

Table 4.18: Findings of UX Experts' Consensus on FOSS UX Maturity Influencing Factors

Items	Triangular Fuzzy Numbers			Defuzzification		
	Average of Fuzzy Numbers	**Threshold Value (d) by item**	**Percentage of UX experts' opinions with d≤ 2**	Crisp Values (Z*)	Ranking	Status
U1	(0.47, 0.71, 0.88)	0.133	83%	0.696	30	Accepted
U2	(0.81, 0.94, 1.00)	0.100	100%	0.930	5	Accepted
U3	(0.38, 0.60, 0.81)	0.142	100%	0.602	35	Accepted
U4	(0.38, 0.60, 0.81)	0.142	100%	0.602	35	Accepted
U5	(0.86, 0.98, 1.00)	0.025	100%	0.960	1	Accepted
U6	(0.49, 0.71, 0.88)	0.125	58%	0.703	29	Accepted
U7	(0.46, 0.67, 0.85)	0.175	83%	0.666	33	Accepted
U8	(0.79, 0.93, 0.99)	0.067	92%	0.917	7	Accepted
U9	(0.64, 0.80, 0.94)	0.175	75%	0.796	17	Accepted
U1	(0.47, 0.71, 0.88)	0.133	83%	0.696	30	Accepted
U2	(0.74, 0.90, 0.97)	0.117	100%	0.885	10	Accepted
U3	(0.56, 0.73, 0.90)	0.242	58%	0.732	27	Accepted
T1	(0.76, 0.91, 0.98)	0.067	100%	0.896	9	Accepted
T2	(0.86, 0.98, 1.00)	0.025	100%	0.960	1	Accepted
T3	(0.84, 0.96, 1.00)	0.100	100%	0.945	4	Accepted
S1	(0.60, 0.78, 0.91)	0.242	58%	0.772	20	Accepted
S2	(0.58, 0.76, 0.91)	0.175	75%	0.756	23	Accepted
S3	(0.80, 0.94, 0.99)	0.075	92%	0.924	6	Accepted
S4	(0.78, 0.93, 0.98)	0.100	83%	0.911	8	Accepted
C1	(0.60, 0.78, 0.91)	0.242	58%	0.772	20	Accepted
C2	(0.70, 0.86, 0.97)	0.150	92%	0.854	13	Accepted
C3	(0.68, 0.83, 0.96)	0.150	83%	0.828	14	Accepted

C4	(0.55, 0.75, 0.90)	0.158	75%	0.743	26	Accepted
C5	(0.58, 0.76, 0.91)	0.175	75%	0.756	25	Accepted
C6	(0.71, 0.88, 0.96)	0.183	100%	0.864	12	Accepted
C7	(0.59, 0.78, 0.93)	0.200	83%	0.775	19	Accepted
C8	(0.60, 0.77, 0.92)	0.267	67%	0.764	22	Accepted
P1	(0.59, 0.78, 0.93)	0.200	83%	0.775	18	Accepted
P2	(0.85,0.97, 1.00)	0.033	100%	0.953	3	Accepted
P3	(0.60, 0.81, 0.94)	0.100	100%	0.799	16	Accepted
P4	(0.58, 0.76, 0.91)	0.175	75%	0.756	23	Accepted
P5	(0.53, 0.74, 0.90)	0.142	75%	0.730	28	Accepted
O1	(0.44, 0.65, 0.84)	0.183	83%	0.647	34	Accepted
O2	(0.50, 0.70, 0.87)	0.250	25%	0.692	32	Accepted
O3	(0.63, 0.83, 0.95)	0.100	100%	0.813	15	Accepted
O4	(0.75, 0.89, 0.98)	0.067	92%	0.883	11	Accepted
Total construct		0.143	84%	N/a	N/a	N/a

4.4.5 Discussion of the Results

The findings reveal that all thirty-six UX maturity influencing factors identified have obtained expert consensus (84%) and meet all FDM analysis standards. As a result, they can all be considered when creating new UXCMMs for the FOSS community. The adopted procedure, i.e., the FDM, is reliable and legitimate for information systems research (IS). It has directly asked knowledgeable academics in the area questions about the factors that may impact UX maturity evaluation and iteratively perform ranking and validations.

4.4.5.1 Interpreting Results

The factors "Users' feedback" and "Adopted FOSS development methods" are at the top of the final ranking list. The participant who initially identified and challenged "User feedback" argued, "*…The quality of user feedback can impact UX maturity. FOSS projects that actively seek and incorporate user feedback tend to have more mature and user-friendly interfaces.*" Likewise, the participant who initially identified and challenged the factor "Adopted FOSS development methods" argued that *"...The FOSS Projects that use modern and flexible infrastructure tend to have more mature UX."*

Ranking "user feedback" and "adopted development technologies" at the top is unsurprising. The use of modern and flexible technologies in reporting, retrieving, and analyzing user feedback has been the current highly emphasized research area in the community for improving overall UX design (Duffy, 2018). However, current practices in the FOSS community offer less attention to user interfaces and related issues (Raza et al., 2010). They frequently address the requirements of a smaller, more niche audience (Swarts, 2019) and, as a result, cause several UX-related problems. For example, reporting and managing users' feedback has been challenging (Nichols & Twidale, 2005; L. Zhao & Deek, 2005). Among other causes, the FOSS community still faces difficulties developing products with desirable UX because of communication barriers, its nature, limited resources, and lack of standardized reporting channels and incentives. This finding is also confirmed by Yusop et al. (2017). The "user feedback" belongs to the theme "usability." Usability has been critical in developing products with desirable UX. Findings from other studies, such as Hassenzahl (2003), Roto et al.(2010), and Sauer et al. (2020), also confirm the argument. Usability affects the product's reputation because users are more inclined to share negative feedback and discourage others from using the software when they struggle with it or have a bad experience.

Table 4.19 details the justifications of UX scholars and other FOSS stakeholders when proposing UX maturity influencing factors in the usability theme.

Table 4.19: UX Experts' Justifications for Proposed FOSS UX Maturity Influencing Factors - Usability Theme

Factor	Justification(s)
Documentation	*"…May help FOSS develop a devoted user base and a robust community by making the program easier to understand and use, resolving problems, fostering community involvement, and making FOSS available to a wider audience."*
Ease of use	"…May *improve user pleasure, efficiency, accessibility, and overall acceptance, resulting in a good UX that satisfies users' needs*."
Speed of use	*"…FOSS that responds slowly can frustrate users and have a bad UX. While a quick and responsive system might result in a good UX and higher user satisfaction."*
Learnability	*"…A FOSS that is easy to learn and understand for new users will provide a satisfying UX."*
bugs reporting procedures	*"…A FOSS with a robust and efficient bug reporting process can lead to a positive and greater UX maturity by impacting how quickly and effectively developers can identify and address issues that affect users."*
Usability/UX testing	*"…May help find problems and areas where FOSS design needs to be improved, resulting in a better UX overall. UX testing entails assessing FOSS from the viewpoint of its users to comprehend how they interact with it and spot any problems or potential improvement areas."*
User requirements	*"..provide critical insights into the needs and preferences of the users who will be using FOSS and therefore promote UCD, users' satisfaction, and continuous improvement."*
UCD practices	*"…It entails creating FOSS while keeping the user's wants, preferences, and habits in mind. It encourages cooperation, openness, and iterative design, as well as participation from the community."*

operability	*"...It refers to the ease with which users can install, configure, and maintain FOSS. It includes aspects such as system compatibility, installation process, ease of updates and upgrades, and the availability of documentation and support. If users face difficulties installing, configuring, and updating or upgrading the software, they are more likely to abandon it, resulting in a poor UX."*
Accessibility	*"...It refers to the ability of users to access and use software applications regardless of their abilities, disabilities, or impairments. It helps make the FOSS inclusive, compliant, innovative, reputable, and satisfying to all users."*
Attractiveness	*"....It affects the first impression, engagement, brand image, usability, and competitive advantage of FOSS, which may help create visually appealing FOSS, which enhances the UX, increase user engagement, and promote the brand image and their reputation ."*

The technologies adopted by the FOSS community during the development of its projects belong to the theme "FOSS technology." Adopted technologies being ranked high is not a surprise. The FOSS community is suggested to adopt artificial intelligence (AI) technology due to the massive data increase and embedded complexity when submitting and addressing users' complaints (Zhou, Pan, Wang, & Vasilakos, 2017). Nevertheless, AI has been an option for quick code production, automatic user behavior analysis, and personalized recommendations (Batarseh, Mohod, Kumar, & Bui, 2020; Murphy-Hill et al., 2021). Finally, AI tools, such as "Copilot/Codex," primarily contribute to source codes (Pudari & Ernst, 2023).

Table 4.20 offers participants convincing arguments for each factor in the theme "Adopted technologies."

Table 4.20: UX Experts' Justifications for Proposed FOSS UX Maturity Influencing Factors - Adopted Technology Theme

Factor	**Argument**
Artificial Intelligence (AI)	"*…May make FOSS more personalized, predictive, efficient, and user-friendly through understanding the user's behavior and preferences, predictive analytics, Natural Language Processing (NLP), automation and sentiment analysis.*"
Technical infrastructures, such as software development tools, libraries, and framework	*"…A well-designed technical infrastructure can facilitate the development of high-quality, user-friendly software that meets users' needs."*

The theme "The FOSS UX Process Integration" contains five factors argued to influence the UX maturity of the FOSS community. However, the "Formalize standards for UX design" factor has emerged as significant, ranked third among all UX maturity influencing factors. This factor may influence the FOSS community's UX maturity by providing a consistent experience to all users across FOSS projects, platforms, and devices, improving users' efficiency in learning and accomplishing their tasks, accessibility, and usability, and encouraging innovations. This conclusion confirms the findings of Kashfi et al. (2019).

Table 4.21 provides participants' arguments regarding the importance of proposed FOSS UX maturity influencing factors in the theme "The FOSS UX Process Integration."

Table 4.21: UX Experts' Justifications for Proposed FOSS UX Maturity Influencing Factors - FOSS Process Integration Theme

Factor	Provided arguments
Regular and systematic use of UX research methods	"*…May involve collecting data about users' needs, behaviors, preferences, and pain points to inform the design and development of FOSS., leading to better FOSS UX that meets the needs and expectations of users.*"
Formalize UX design standards.	*"…May help to create consistency, clarity, and usability across different FOSS projects, creating projects that provide users with a better UX and more likely usable."*
Timing to include UX methods, practices, principles, and tools	*"…May help increase the likelihood of delivering FOSS projects with desirable UX."*
Consistently tracking UX-related tasks.	*"…May help improve UX maturity in FOSS by identifying areas for improvement, measuring progress, encouraging collaboration, enabling data-driven decision-making, and facilitating continuous improvement."*
Monitor UX quality constantly.	*"…May help FOSS developers identify issues and pain points, measure user satisfaction, monitor user engagement, gather user feedback, and encourage continuous improvement that matures to improved UX."*

In "The FOSS UX strategy" theme, the factor "FOSS allocate adequate financial resources for UX-related works" is ranked fifth. According to Levy (2015), UX strategy lies at the nexus of business strategy and UX design. When "the FOSS UX strategy" is empirically created and adopted, it may offer a significantly better probability of creating a successful digital product than simply developing some wireframes and writing many source codes.

Table *4.22* provides arguments about the UX maturity influencing factors in the theme "FOSS UX Strategy."

Table 4.22: UX Experts' Justifications for Proposed FOSS UX Maturity Influencing Factors - UX Strategy Theme

Factor	**Arguments**
FOSS UX leadership	*"…May prioritize UX design, create a culture of UX maturity, allocate resources, encourage collaboration, and hold teams accountable for UX design to ensure FOSS projects are designed with the user in mind and provide a better UX."*
Planning of FOSS UX-related activities	*"…May help FOSS developers set goals and objectives, conduct user research, plan for user testing, adopt an iterative design process, and foster collaboration to create projects designed with their users' needs in mind and provide a better UX."*
FOSS allocates adequate financial resources for UX-related works	*"…May help FOSS projects hire UX professionals, conduct user research, invest in design tools, conduct user testing, and support the community to create projects with the needs of its users in mind, better UX and improved UX maturity."*
FOSS dedicated UX staff with specialized UX skills	*"…May improve the UX maturity of the open-source software community by ensuring UCD design, design consistency, accessibility, user testing, design documentation, and collaboration."*

The theme "The FOSS UX culture" is equally important in the overall FOSS UX maturity. The group ranks "Regular training on UX-related activities" high at the 13th position. Regarding this study, the theme "FOSS UX culture" often refers to the values, beliefs, and practices a FOSS community adopts to prioritize and improve its products or services. It involves creating an environment that encourages and supports the development of UCD practices, user testing, and ongoing UX improvement (Bargas-Avila & Hornbæk, 2011) and must consider the cultural context in which its projects will be used (Paay et al., 2018). A strong UX culture can improve user satisfaction, increase loyalty, and give a competitive advantage in the marketplace. The study's explanation adapts the explanation provided by Hassenzahl and Tractinsky (2006) and Law et al.(2014).

Table *4.23* provides initial justifications for UX Maturity Influencing Factors in the FOSS UX culture theme.

Table 4.23: UX Experts' Justifications for Proposed FOSS UX Maturity Influencing Factors - UX Culture Theme

Factor	**Justification(s)**
FOSS developers' UX skill	*"…May help improve UX maturity in the FOSS community by incorporating design thinking, conducting usability testing, ensuring accessibility, enhancing the visual design, and collaborating more effectively."*
FOSS Regular training on UX-related activities	*"..Can benefit the FOSS community by improving team members' UX skills, keeping them up-to-date with the latest trends and best practices, ensuring consistency, promoting collaboration, and incorporating user feedback."*
FOSS Esteeming UX skill sets similar to technical skills	*"..Can help improve the UX maturity of the FOSS community by increasing recognition of UX expertise, encouraging collaboration, incorporating UX earlier in the development process, improving software usability, and attracting more UX professionals to FOSS projects."*
The organizational level of flexibility in FOSS	*"..Can improve FOSS UX maturity by allowing for quick adaptation to changing user needs, fostering experimentation and innovation, enabling collaboration with diverse contributors, leveraging existing tools and resources, and enabling customization of FOSS projects to meet specific user needs."*

FOSS considering UX as a professional	*"…May help the FOSS community to appreciate the importance of UX better, attract more dedicated UX resources, improve collaboration, establish design standards, and drive innovation."*
Buy-in of UX ideas	*"…May help improve UX maturity in the FOSS community by promoting a UCD approach, facilitating collaboration, encouraging user research and testing, promoting design consistency, and encouraging continuous improvement."*
Clear UX maturity roles for each FOSS stakeholder	*"…May help improve user experience maturity of the FOSS community by encouraging shared ownership, reducing overlap and duplication, promoting accountability, increasing efficiency, and helping to identify gaps and areas for improvement."*
FOSS includes UX professionals as decision-makers	*"….Can improve the UX maturity of the FOSS community by bringing user-centric perspectives, ensuring design consistency, providing expert knowledge, facilitating collaboration, and enhancing user satisfaction and engagement."*

Finally, the theme "The FOSS UX Outcome" is equally essential because by focusing on it, organizations can make data-driven decisions to improve the UX and, ultimately, drive business success. Current knowledge ties "the FOSS UX outcomes" to specific goals for a product or service, such as increasing user engagement, improving user satisfaction, or reducing user errors users make when using products (Hassenzahl & Tractinsky, 2006). The theme measures the impact of the UX design on the user and the business (Hassenzahl, 2010). It involves collecting data on user behavior, attitudes, and perceptions through various methods, such as surveys, interviews, usability testing, and analytics (William & Thomas, 2013). Among all the factors in the theme, "Number of errors made by users" is ranked 11th by this study.

Table 4.24 lists the factors under the theme "The FOSS UX outcome " along with initial justifications from those who proposed them.

Table 4.24: UX Experts' Justifications for Proposed FOSS UX Maturity Influencing Factors – the UX Outcome Theme

Factor	Initial arguments
UCD goals predicting FOSS UX metrics	*"..can help improve the UX maturity of the FOSS community by ensuring that design solutions are focused on meeting the users' needs and are more effective in achieving their intended outcomes."*
Regular review/update of FOSS UX metrics	*"... May contribute to the UX maturity of the FOSS community by providing data-driven insights into the UX, identifying areas for improvement, and promoting accountability and transparency in the design process."*
FOSS stakeholders' engagements	*"... May ensure that FOSS design solutions meet the needs of their users and stakeholders, build trust and collaboration, and ultimately, improve the overall user experience."*
Number of errors made by users	*"...When users encounter errors while using FOSS, it can lead to frustration, confusion, and a negative experience, making users less likely to use the application in the future. It can even lead to negative reviews and word-of-mouth feedback that can harm the FOSS's reputation."*

4.4.6 Study's Contribution and Implications to Research and Practice

While there has been some research on UX in FOSS projects, limited studies identify factors influencing the UX maturity of the projects (Cheng & Guo, 2018). As a result, the FOSS community still lacks a definitive list of factors influencing its UX maturity, making it challenging to create the right UXCMMs to measure its UX maturity and suggest how to improve the status quo. Generically developed UXCMMs may not apply in the FOS community because FOSS projects have varied priorities, goals, and user groups that accumulate into different dynamics than other software-developing communities.

As a significant contribution to the body of knowledge, this study highlights potential dangers to the accuracy and legitimacy of the FOSS community's UX maturity assessment by drawing a picture showing the absence of UX maturity influencing factors that account for the community's dynamics. Drawing this picture can help identify areas for improvement and provide insights into promoting better UX practices and the broader adoption of FOSS projects. As some of the researchers involved in the panel told us during the study, this discussion is urgent to increase user engagement, improve user satisfaction, reduce user errors, or promote the development of more user-friendly FOSS projects.

The study has used FDM to express the UX experts' consensus and justify the validity of all thirty-six UX maturity influencing factors relevant to the FOSS community. The method has also assisted in ranking the factors based on their priorities (see **Table 4.18**). However, since the metamorphosis of the FOSS development methods has not ceased and will keep advancing, evaluating whether other UX maturity influencing factors emerge periodically is crucial. As a result, the study has pointed out three topics that require more research. First, it has recommended additional empirical research using alternative sampling techniques and other strategies to upsurge analyzed factors and reach a relatively large population of stakeholders and FOSS projects. These studies may help test variables in the broader space and generalize the gathered findings to a large population. They may also uncover variables and other issues not noticed in the completed study. As a different initiative, the study recommends that future studies consult FOSS experts, practitioners, developers, users, and the industry to get their points of view regarding UX maturity influencing factors and compare the results for a broader overview.

Second, the study recommends a rigorous exploration of the IS literature on FOSS in search of specific instances of UX maturity influencing factors to allow quantifying the extent to which the factors have painted current FOSS research. Finally, the completed study has indicated the need to revise existing UX maturity models and other UX practices to suit the dynamics of the FOSS community, as was also suggested by Terry et al. (2010). This suggestion results from the reality that none of the current UX maturity models was designed for the FOSS community or sufficiently accounted for its UX characteristics and maturity influencing factors (Namayala et al., 2022).

4.4.7 Conclusion

Like many other software-developing communities, the FOSS community must correctly and consistently assess its UX maturity. According to the information corpus, practical UX maturity evaluation approaches must consider organizational dynamics that may vary from one organization or environment to another. To address organizational UX dynamics, it is frequently advantageous to understand the elements that influence UX maturity in a specific organization. However, despite ad hoc interventions, it has been determined that the FOSS community has made little to no effort to analyze the aspects that influence the UX maturity of its projects. Nevertheless, current initiatives lack methodological rigor, are inconsistent, and only cover a few influencing factors. This study aims to fill this gap by conducting an empirical study to assess which UX maturity influencing factors influence UX maturity in the FOSS community.

The UX maturity influencing factors were established from an exhausted literature review, interviews with FOSS stakeholders, and consultations with other UX assessment experts. It then adopted the FDM to establish factors that may apply in the FOSS community. The results from the experts' consensus accepted all thirty-six factors with a total percentage agreement of 84%. Agreed UX maturity influencing factors are supported with justifications on how they are critical in influencing the UX maturity of the FOSS community. This study is the first of its own in the FOSS community.

The discussion of the findings clarified that understanding the UX maturity influencing factors is associated with many advantages to the FOSS community. Findings from the completed study may have three immediate impacts or contributions to the body of knowledge. (1) They may provide a scientific way of creating UX maturity influencing factors for the FOSS community. (2) They may help the FOSS community's stakeholder plan and adjust course, culture, and policies toward an ideal assessment of UX maturity. (3) They may motivate future researchers to actively discuss UX maturity influencing factors specific to the FOSS community to improve its UX maturity assessment.

Finally, like any other study, this study has several limitations. For example, it relies solely on the opinions of UX experts, although highly qualified. It has also used a relatively small sample size and may accommodate unfamiliar participants. Nevertheless, the author believes that when carrying out this investigation, they may have missed certain crucial factors that significantly influence the UX maturity of the FOSS projects.

4.5 Developing the Free and Open Source Software User Experience Maturity Model (FOSS-UXMM)

4.5.1 Introduction

This subsection aims to create a model, FOSS-UXMM, as an artifact that solely evaluates the UX maturity of desktop-based FOSS projects. It explains the structure of FOSS-UXMM, including the rating system and a set of process attributes as measurable characteristics. Generally, the subsection intends to address the fourth specific objective, **"To develop the UX Maturity Model (FOSS-UXMM) for assessing the UX maturity of desktop-based FOSS projects using established UX maturity influencing factors."**

The components of the FOSS-UXMM, which builds its novelty, were derived from completed empirical research and exhausted literature reviews. The FOSS-UXMM model is both descriptive and prescriptive, which means it helps FOSS projects identify their strengths and weaknesses in implementing UX-related activities and recommends what strategies should be implemented to improve the status quo and progress to the next level of UX maturity.

4.5.2 Background Information

The primary goal of this subsection is to systematically create a model for evaluating the UX maturity of desktop-based projects. Adopting the model can help the project developers, managers, and owners assess the organizational ability to develop projects with desirable UX, which could better fit user needs and expectations (CMMI Product Team, 2010; Staggers et al., 2015). Ideally, we cannot improve anything unless it is measured accurately. Therefore, unless FOSS project decision-makers can evaluate organizational UX maturity, they may be unable to enhance their organizational capacity to build projects with excellent UX design.(Raza et al., 2012a).

Models do not have universally agreed definitions, mainly due to their abstract nature. Although several scholars annotate them differently, this study adopted the definition of Tomhave (2005), which recognizes models as "*abstract, conceptual constructs that represent processes, variables, and relationships without providing specific guidance on, or practices for, implementation*." A model might be conceptual (resides in minds) or a tangible thing simulating the authentic artifact's use case. Theoretically, comprehending the components of a model aids in understanding its conceptualization. According to Lethbridge and Laganiere (2004), A typical model consists of four features: elements of the participant variable, their relationships, processes, and precise instructions for its utilizations and applications.

From the author's knowledge accumulated from the examined literature, the FOSS community does not have a model specific for assessing the UX maturity of its projects. Nevertheless, using the generically developed UXCMMs to assess community-specific issues has been unrealistic because they do not perfectly fit the environments (Terry et al., 2010). The domain or context in which a software product is utilized impacts the UX significantly (Eriksson & Ferwerda, 2021). Therefore, developing the FOSS-UXMM model may be ideal for evaluating the FOSS community's UX maturity. FOSS-UXMM details the relationship between FOSS projects' UX maturity and influencing factors at five different maturity levels. Each UX maturity level usually reaches maturity in a critical subset of UX-related processes(Chrissis, Konrad, & Shrum, 2011).

4.5.3 Architecture and Operations of FOSS-UXMM

Compared to other software quality aspects, "UX" is exceptionally complicated, subjective, and varies significantly among users. It has been challenging for decades to create products that satisfy and enthrall all categories of users. For example, what one user may perceive as a product with acceptable UX may also be perceived as a product with terrible UX by another user. Although UX is highly subjective, the UX assessment community must handle it objectively to allow improvement. However, objective treatment of UX mandates alternative UX evaluation methods. As a result, this study introduces a novel UX maturity evaluation methodology for the FOSS community, FOSS-UXMM (see **Figure 4.5Error! Reference source not found.**), developed by adopting steps highlighted in the hybrid framework for developing UXCMMs.

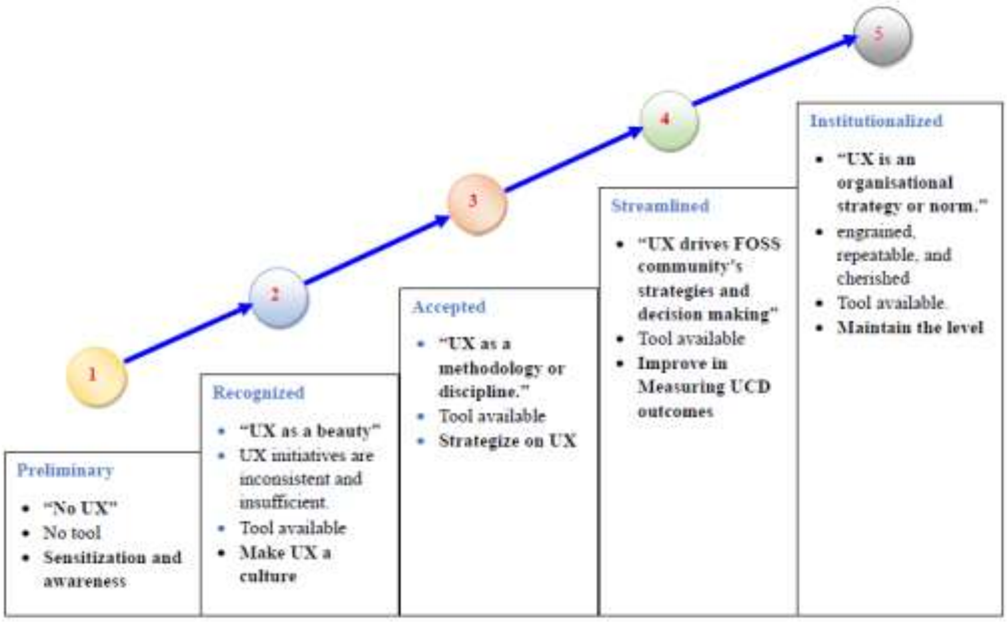

Figure 4.5: Free and Open Source Software Maturity Model

The assessment methodology of the FOSS-UXMM is based on the self-administered questionnaires that collect developers', managers', and owners' opinions regarding factors influencing FOSS projects' UX maturity. The questions in the questionnaires explain how various UX-related tasks are carried out and organized during the SDLC of FOSS projects. The UX maturity level reached by a particular FOSS project significantly depends on how developers, managers, and owners agree with the questions in the questionnaire.

The configurations of the FOSS-UXMM contain the thirty-six (36) UX maturity-impacting factors organized into six themes: UX strategy, UX process, UX culture, usability, the technology chosen, and UX outcomes established from finished empirical studies (see **Table 4.25**). Despite several shortcomings, the FOSS-UXMM significantly contributes to the methodology for assessing the UX design of the

FOSS community by tackling critical problems of current UXCMMs, such as influencing factors being developed from owners' years of working experience.

Table 4.25: Configuration of the FOSS UX Maturity Evaluation Methodology

Theme	Practice #	FOSS UX Maturity influencing factor
Usability	1	Documentations and support
	2	Ease of use
	3	Speed of use
	4	learnability
	5	Users' feedback
	6	Bugs reporting procedure
	7	usability testing
	8	Meeting users' requirements
	9	User-centered Design (UCD) Methodology
	10	Operability
	11	Accessibility
	12	Attractiveness
Adopted Technologies	13	Artificial Intelligence (AI) (
	14	Adopted development methods
	15	Technical infrastructures, such as software development tools, libraries, and frameworks
UX Strategy	16	UX leadership
	17	Planning of UX-related activities
	18	Allocating adequate financial resources for UX-related works
	19	Dedicating UX staff with specialized UX skills
UX Culture	20	Developers' UX skill
	21	Regular training on UX-related activities
	22	Esteeming UX skill sets similar to technical skills.
	23	The organizational flexibility
	24	Considering UX as a professional

Theme	Practice #	FOSS UX Maturity influencing factor
	25	Executive understanding and buy-in of UX ideas
	26	Clear UX maturity roles for each stakeholder
	27	UX professionals are part of the decision-makers
UX Process integration	28	Regular and systematic use of UX research methods.
	29	Formalize standards for UX design.
	30	Timing to include UX methods, practices, principles, and tools
	31	Consistently tracking UX-related tasks.
	32	Monitor UX quality constantly.
UX Outcome	33	UCD goals predicting UX metrics
	34	Regular review/update of UX metrics
	35	Early engagement of all stakeholders
	36	Number of errors made by users

Usability

Describes the extent to which a product, system, or service can be used by its intended users to achieve specific goals effectively, efficiently, and satisfactorily. In other words, it is about making things easy to use and ensuring they meet users' needs and expectations. In the configuration of FOSS-UXMM, usability contains documentation and support, ease of use, speed of service, learnability, users' feedback, and bug reporting procedure. It also entails usability testing, meeting users' requirements, UCD methodology, operability, accessibility, and attractiveness.

Adopted Technologies

The FOSS-adopted Technologies describe technological factors that may be adopted to enhance FOSS UX maturity. As technology evolves, expectations of developing new tools and techniques to improve UX in the FOSS community arise. With present knowledge, several tools that support the improvement of UX in various specialties exist. These tools can efficiently perform user analytics, research, design collaboration, and prototyping. They can also increase accessibility. In configuring FOSS-UXMM, adopted technologies entail adopting Artificial Intelligence and machine learning techniques in developing FOSS projects, techniques for improving development methods and used technical infrastructures, such as software development tools, libraries, and frameworks.

UX Strategy

The UX strategy is a plan that outlines how an organization will deliver a UCD experience to its users. It defines the organization's approach to UX design and how it will continuously incorporate user feedback to improve its products and services. It helps to ensure that user needs are at the forefront of product development and that the product or service meets those needs meaningfully. A UX strategy typically includes user research, user personas, user journeys, design principles, metrics, and cross-functional collaboration. In the configuration of FOSS-UXMM, the UX strategy consists of UX leadership, planning of UX-related activities, allocating adequate financial resources for UX-related works, and dedicated UX staff with specialized UX skills.

UX Culture

The UX culture refers to the shared beliefs, values, attitudes, and practices of UX design within an organization. It encompasses how an organization approaches UX design and integrates it into its overall strategy, processes, and culture. A strong UX culture prioritizes user needs and advocates for the user throughout product development. It encourages cross-functional collaboration, with designers, developers, product managers, and other stakeholders working together to deliver the UCD experience. It also values continuous improvement and incorporates user feedback into the design process. It continually embraces experimentation, testing, and iteration to refine and improve the UX. In the configuration of FOSS-UXMM, the UX culture contains the developers' UX skillset, regular training on UX-related activities, esteeming UX skill sets similar to technical skills, and the organizational flexibility level. Others include considering UX as a professional, buy-in of UX ideas or executive understanding, clear UX maturity roles for each stakeholder, and including UX professionals as decision-makers.

The UX Process Integration

The UX process integration refers to integrating UX design processes into an organization's overall product development process. It involves aligning UX design activities with the organization's goals, processes, and culture to ensure user needs are at the forefront of product development. It typically entails understanding the organization's goals and user needs, identifying the UX design process, aligning UX design with product development, collaborating across teams, and measuring success. In the configuration of FOSS-UXMM, the UX process integration entails regular and systematic use of UX research methods and formalizing UX design standards. Others include timing to include UX methods, practices, principles, and tools, tracking UX-related tasks consistently, and constantly monitoring UX quality.

The UX Outcome

The UX outcome refers to the result of UX design efforts, and it is the measurable impact or benefit of delivering a UCD product or service. We can measure the outcome of UX design in various ways, such as increased user satisfaction, improved user engagement, higher conversion rates, increased customer loyalty, and improved business performance. It is essential to establish clear goals and objectives for the UX design effort and to define metrics used to evaluate success to measure UX design outcomes. These requirements may involve conducting user research, defining key performance indicators (KPIs), and setting benchmarks to track progress over time. In configuring FOSS-UXMM, the UX outcome entails UCD goals predicting UX metrics, regular review/update of UX metrics, stakeholders' engagements, and the number of errors users make.

Unlike other UXCMMs, which are relatively complex, expensive, and not easily accessible, FOSS-UXMM is free and straightforward for average users. With FOSS-UXMM, a single respondent or a group of respondents can self-assess the UX maturity of the FOSS projects. However, the study encourages engaging multiple respondents who must agree on their ratings to reduce biased results.

The FOSS-UXMM is a linear model (see **Figure 4.5**) that does not allow skipping any stage. Its assessment results are based on the agreement among project owners, managers, core developers, and contributors on each questionnaire statement. The questionnaires' statements represent UX maturity influencing factors relevant to each UX maturity level, and the number of statements varies between levels because of the varied requirements. As a result, the study created four UX maturity assessment questionnaires, one for each maturity level, with an assessment tool to help demonstrate how FOSS-UXMM works.

Throughout the remaining parts, the study will use the following set of abbreviations to identify themes and the thirty-six UX maturity influencing factors identified: Usability (UBI), Adopted technologies (ATE), UX strategy (UXS), UX culture (UXC), UX process integration (UXPI) and UX outcome (UXO). Nevertheless, the questionnaires for the proposed maturity model contain the following symbols and abbreviations:

UXMF = User Experience Maturity Factor
ML = Maturity Level (an integer)
S = Statement
UXN = User Experience Factor Number (an integer)
Each questionnaire is numbered as S.ML.UXMF.UXN

Regarding the adopted maturity scales, the study adopted the most popular maturity scale, a 5-level scale like the CMM by Paulk et al.(2011), because it is straightforward enough to identify differences between levels. The five UX maturity levels of the FOSS-UXMM include "Preliminary," "Recognised," "Accepted," "Streamlined," and "Institutionalized" (see **Table 4.26**).

Table 4.26: FOSS-UXMM Proposed UX Maturity Levels

Level #	Level name	Characteristics
1	Preliminary	Ideally, the characteristics of this level accumulate into a wake-up call. It is characterized by **“No UX,”** where the evaluated organization either does not know about UX or believes it is unnecessary. At this level, All UX maturity influencing factors are not considered. important
2	Acknowledged	The stage is characterized by **"UX as a beauty,"** where the FOSS community begins acknowledging UX's values. Developed products frequently show UX promises by satisfying functional needs. However, actions often come late, are inconsistent and insufficient, and may not always involve the appropriate parties or methods.
3	Accepted	This level is distinguished by **"UX as a methodology or discipline,"** in which the FOSS projects employ semisystematic UX-related methodology that is widely used but varies in effectiveness and efficiency.
4	Streamlined	UX work is now incredibly effective in achieving organizational goals, and it is thorough, efficient, and all-pervasive. Almost all teams within the company typically carry out UX-related tasks correctly and efficiently. Nevertheless, there is frequent innovation in UX techniques and procedures and contributions to the discipline. The performance metrics that matter most to the leaders are those that center on or are fueled by efforts connected to UX.
5	Institutionalized	The FOSS-UXMM distinguishes this level by the strategy for innovation and change, which creates a setting where everyone is fully aware of UCD

Level #	Level name	Characteristics
		techniques. It treats UX as the standard: engrained, repeatable, and cherished. In their day-to-day work, UX is a priority for leaders, teams, and employees. Understanding users' demands through research influences the organization's strategy and project priority. As a result, development processes are iterative and user-focused.

Level 1: Preliminary

Comparing this level to the levels of other maturity models, such as the UX maturity model by Pernice et al.(2021), the preliminary level combines Absent and Limited levels. It is the initial level of UX maturity for FOSS projects, indicating that UX receives no consideration. At this level, FOSS communities are either unaware of UX or think it is not essential, and developed projects lack consistent and well-organized processes for their implementation. Nevertheless, no evidence supports the availability of initiatives to implement UBI, ATE, UXS, UXC, UXPI, or UXO factors.

To advance to the next level of maturity, the FOSS community has to raise UX awareness and literacy among stakeholders. For the UX to take off, it is necessary to find suitable UX design-related organizational needs and inspire users and critical decision-makers. Motivating other stakeholders can be done in several ways, such as displaying success stories and case studies or forming alliances with UX advocates from within or outside the organization.

When using FOSS-UXMM, a project is usually branded at the preliminary level if it does not match the requirements of the higher levels. As a result, no instrument exists to assess UX maturity at the "Preliminary" level.

Level 2: Acknowledged

At this stage, project teams know how UX may contribute to the success of FOSS projects. Nevertheless, UX has become a hot topic often debated amongst project stakeholders; however, all stakeholders do not perceive its importance positively.

Project teams may demonstrate an interest in UX by developing the necessary infrastructures and assigning personnel to UX positions at this level. However, allocated personnel are insufficient or do not necessarily have the qualifications needed. They are also involved in the later phases. Deliberations of UX work rely on personal interpretations and opinions instead of organizational policies or strategies.

The organizational UX goals are extensive and extremely hard to quantify. Finally, projects lack UX-related champions or agents, so many UX activities are inconsistently implemented.

To advance from the “Acknowledged level” to the “Accepted level,” organizations at the “Acknowledged level” must emphasize creating a UX culture that prioritizes UX-related tasks and makes them essential to the decision-makers. One way of prioritizing UX-related tasks is by creating pilot projects with obvious links between UX design and organizational goals managed by UX professionals.

FOSS-UXMM has a measurement tool for the "Acknowledged" level, illustrated as follows.

UXMF.2.1	**Usability**					
#	**Statement**	**0**	**1**	**2**	**3**	**4**
S.2.1.1	Project designers and developers address users’ requirements as the only way to enhance UX maturity.					
S.2.1.2	Project managers collect and analyze feedback from the majority of project participants.					
S.2.1.3	Project designers and developers are acquiring knowledge about the domain of usability engineering.					
S.2.1.4	The project team recognizes the importance of the UCD methodology in UX maturity.					
S.2.1.5	Project designers and developers are learning ways to improve the understandability of their products among users.					

S.2.1.6	The project team realizes the need for the learnability of their product for their prospective users.
S.2.1.7	Project developers acknowledge that what works for them may not work for end users.
S.2.1.8	The project team gathers data on how to improve the attraction of their product.
S.2.1.9	The development team understands the challenges that users face when reporting usability-related defects.
S.2.1.10	The project team realizes the need for a usability assessment plan before and after the release of the software.
S.2.1.11	The project team recognizes the necessity of documenting but lacks a systematic and planned approach.
UXMF.2.2	**Adopted Technologies**
S.2.2.1	The project team understands adopting user analytics, research, design, collaboration, and prototyping tools enhances UX maturity.
S.2.2.2	The project uses technical infrastructures, such as software development tools, libraries, and frameworks, to enhance UX maturity.
S.2.2.3	The project team agrees different software development methodologies affect UX maturity differently.
UXMF.2.3	**UX Strategy**
S.2.3.1	The project team implements UX-related principles to help FOSS projects succeed.
S.2.3.2	The project leaders allocate dedicated human resources to perform UX-related activities; however, they are insufficient and do not necessarily have the required qualifications.

S.2.3.3	The project teams consult UX professionals to improve UX design; however, it does so in the late stages.
S.2.3.4	No funds are spent to conduct UX design-related activities.
S.2.3.5	No UX design goals exist; if there are any, they are general and hard to measure.
S.2.3.6	There is ad hoc management of UX design activities.
S.2.3.7	UX design is often not part of the project's strategy.
S.2.3.8	UX design-related activities are not perceived as necessary by stakeholders outside the project team because they do not directly impact them.
UXMF.2.4	**UX Culture**
S.2.4.1	The project team understands developers' UX skill set helps to improve UX maturity.
S.2.4.2	The project team understands that support from project leadership improves UX maturity, but the project's administration does not support UX works in any way.
S.2.4.3	Few or no formal UX roles exist. However, some individuals in the project team are self-motivated and combine UX work with their regular jobs.
S.2.4.4	Unorganized UX work is being completed, resulting in unpredictable product quality.
S.2.4.5	The project team's buy-in of UX ideas or the executive understanding improves UX maturity.
UXMF.2.5	**UX Processes Integration**
S.2.5.1	The project team understands the

	organization's goals and user needs, but they are extensive and challenging to quantify
S.2.5.2	Stakeholders outside the project team know UX but do not accept or support it.
S.2.5.3	The project team understands how to align UX design with the product development process.
S.2.5.4	The project integrates some UX processes in UX design. However, the integration is inconsistent.
S.2.5.5	The team starts defining key performance indicators (KPIs) for measuring success; however, they rely on opinions rather than policies and strategies.
UXMF.2.6	**UX Outcome**
S.2.6.1	20% of quantitative UX metrics for assessing the created designs' excellence are gathered and debated.
S.2.6.2	The developers understand using UCD to predict UX metrics improves UX maturity.
S.2.6.3	Final projects engage at least 20% of stakeholders.
S.2.6.4	20% of users can accomplish their goals effectively and efficiently while using the project

Level 3: Accepted

This stage recognizes UX as a methodology or discipline where UX is becoming critical to FOSS design. The infrastructures for successfully implementing UX-related activities are also in place and continually improved. The project team can use these infrastructures to collect requirements for implementing UX-related activities. However, the team leaders still informally bring outside UX experts or researchers due to existing limitations. Nevertheless, only a few stakeholders are irregularly consulted for feedback, but not in the right ways and timing. Despite UX goals becoming apparent to some projects and occasionally measured, delegated budgets for their implementations are limited and may be misallocated or spent on non-UX-related tasks based on organizational politics.

Although stakeholders other than the project team, such as marketing and quality assurance, implement UX-related activities at this level to sustain requirements for the "Acknowledged level" and the "Preliminary" phases, there is a lack of senior leadership and consistent implementation of roles across projects and teams. To advance from the "Accepted level" to the "Streamlined level," the FOSS project teams must improve strategies for implementing UX-related activities and overcoming hidden obstacles. These obstacles, however not exhaustive, may include unclear team goals and unaccountable UX experts. The project participants must feel free and welcome to contribute to the UX outcomes. Thus, there must be a clear definition of UX roles, promoting UX-related activities and encouraging owners, developers, managers, and contributors to release budgets and offer increased support.

Below is an illustration of the measurement tool for the FOSS project's UX maturity at the "Accepted" level.

UXMF.3.1	**Usability**					
#	**Statement**	**0**	**1**	**2**	**3**	**4**
S.3.1.1	Identifying target users and collecting their requirements are essential for achieving project success.					
S.3.1.2	A system for gathering user feedback has been devised.					
S.3.1.3	The project team is dedicated to learning about usability engineering.					
S.3.1.4	UCD is critical for the project's ability to satisfy its target users.					
S.3.1.5	The FOSS team has devised a systematic strategy to improve the understandability of					

	their software product.
S.3.1.6	To improve the learnability of their product, the project team used a well-designed process.
S.3.1.7	Weak points of the project's operability are discovered, and corrective actions are taken.
S.3.1.8	A strategic plan is developed to boost the project's desirability.
S.3.1.9	At least 30% of project developers have learned the technical skills to provide users with convenient usability issue reporting options.
S.3.1.10	The project team has devised a strategy for testing the software's usability before deployment.
S.3.1.11	Formal user requirements, design, and testing documentation are available to the project team.
UXMF.3.2	**Adopted Technologies**
S.3.2.1	The project team adopts user analytics, user research tools, design collaboration tools, prototyping tools, accessibility tools, artificial intelligence, and machine learning to improve UX maturity.
S.3.2.2	The project team implements UX-related technical infrastructures, such as software development tools, libraries, and frameworks, to enhance UX maturity.
S.3.2.3	The project team adopts formal software development methodologies to improve UX.
UXMF.3.3	**UX Strategy**
S.3.3.1	The project has short-term and long-term UX design goals

S.3.3.2	The project team often includes UX work in project planning schedules.
S.3.3.3	The project has some staff with UX skills or experiences exclusively for UX-related activities, but UX is not their primary job.
S.3.3.4	The project team funds some UX work, but the allocated budget is inadequate and not ring-fenced; it may be used for something else when needed.
S.3.3.5	The team leaders informally bring in outside UX professionals to help with the project UX works.
S.3.3.6	UX is critical and acknowledged as a methodology, discipline, or professional.
S.3.3.7	The project team frequently contacts users to explain UX design choices. The timing or manner, though, is not always ideal.
UXMF.3.4	**UX Culture**
S.3.4.1	30% of the project team informally developed UX skill sets to help improve UX maturity.
S.3.4.2	The projects lack leadership, and few or no formal UX roles exist.
S.3.4.3	The UX works are being completed carelessly, and little to no efforts are made to assess or enhance them.
S.3.4.4	30% of the project team buy-in of UX ideas to improve UX maturity
UXMF.3.5	**UX Processes Integration**
S.3.5.1	The project team accepted the organization's goals and user needs but made trade-offs for their implementation.
S.3.5.2	Stakeholders outside the project team accept

	and support UX-related activities.
S.3.5.3	The project team informally aligns UX design with the product development process.
S.3.5.4	The project integrates 30% of UX processes into the design.
S.3.5.5	The team occasionally uses key performance indicators (KPIs) to measure project success.
UXMF.3.6	**UX Outcome**
S.3.6.1	40% of quantitative UX metrics for assessing the created designs' excellence are gathered and debated.
S.3.6.2	30% of developers use UCD to predict UX metrics to improve UX maturity.
S.3.6.3	Final projects engage at least 40% of stakeholders.
S.3.6.4	40% of users can accomplish their goals effectively and efficiently while using the project

Level 4: Streamlined

At this level, the project team has amassed adequate resources to suit the needs of its users by smoothly implementing various UX-related activities. Moreover, UX drives FOSS community strategies and decision-making, and UX design is no longer a conversation topic since it is all-inclusive, pervasive, and omnipresent. UX goals are precise, quantifiable, and integrated into the project mindsets. UX is a focus of or perhaps the driving force behind the project's crucial success metrics that leaders care about most. To be precise, UX methods, procedures, and contributions are frequently improved, which makes projects display UX design excellence as a differentiator and selling proposition.

At this level, the product quality is consistent because UX is always under control, and the relationship between UX ROI and business objectives is fully understood.

FOSS stakeholders are structurally involved, and team members receive and discuss user feedback to propose ideas and innovative approaches. Nevertheless, project managers, owners, and UX-accountable senior management use qualitative and quantitative UX data to inform decisions. Teams with different disciplines assume distinct roles in projects, and members from one team know the functions of members from another team. Appropriate UX design practices, processes, and guidelines are consistently employed in research, which may include creating and evaluating several prototypes.

To advance from the "Streamlined maturity level" to the " Institutionalised UX maturity level," the FOSS Community must develop new tactics that might center on UCD outcome measures beyond projects' UX to acquire a big picture. For example, it has to make UX-related activities solve users' and the community's goals.

Below is an illustration of the measurement tool for the FOSS project's UX maturity at the "Streamlined" level.

UXMF.4.1	**Usability**					
#	**Statement**	**0**	**1**	**2**	**3**	**4**
S.4.1.1	The project has sufficient resources and capabilities to gather user requirements and meet user expectations.					
S.4.1.2	The project team learns from user feedback and avoids making the same mistakes twice.					
S.4.1.3	For usability learning, project managers employ both formal and informal methods.					
S.3.1.4	The project team has sufficient resources and expertise to implement the UCD methodology.					
S.4.1.5	User feedback is used to monitor user understandability regularly.					
S.4.1.6	The ability of new users to understand the					

	software is regularly checked through their reactions and feedback.
S.4.1.7	The established strategic strategy is implemented to improve the software project’s operability.
S.4.1.8	A strategic plan is put in place to make the project more appealing.
S.4.1.9	The project development team has created a service protocol to allow end users to report usability flaws and errors easily.
S.4.1.10	Project testers learn from previous experience and test findings to avoid making the same mistakes.
S.4.1.11	The project documentation contains a clear set of principles for dealing with usability difficulties.
UXMF.4.2	**Adopted Technologies**
S.4.2.1	The project team has adequate resources and skills to adopt user analytics, user research tools, design collaboration tools, prototyping tools, accessibility tools, artificial intelligence, and machine learning to improve UX maturity.
S.4.2.2	To enhance UX maturity, the project team has sufficient resources to implement technical infrastructures, such as software development tools, libraries, and frameworks.
S.4.2.3	The project team has enough resources to implement formal software development methodologies to improve UX.
UXMF.4.3	**UX Strategy**

S.4.3.1	The project has resources to implement and evaluate short-term and long-term UX design goals.
S.4.3.2	The project team consistently includes UX work in project planning schedules.
S.4.3.3	The project has enough staff with UX skills or experiences whose UX is their primary job.
S.4.3.4	All stakeholders are regularly involved, consulted, and systematically receive and discuss user feedback to offer opinions and innovative UX design ideas.
S.4.3.5	Senior management leadership is accountable for designing UX-related tasks and values them equally as meeting the system's functionality.
S.4.3.6	UX is critical in design and acknowledged as a methodology, discipline, or professional.
S.4.3.7	The project team frequently contacts users to explain UX design choices. The timing or manner, though, is not always ideal.
UXMF.4.4	**UX Culture**
S.4.4.1	The project team formally develops UX skill sets to help improve UX maturity.
S.4.4.2	The project has UX leadership, and formal UX roles exist.
S.4.4.3	The project team buy-in of UX ideas to improve UX maturity
S.4.4.4	UX design and its ROI are consistently described and understood across the entire organization, but the UX design is seen as being limited to interface design only.

S.4.4.5	Project managers and owners make decisions using all stakeholders' qualitative and quantitative UX data.
S.4.4.6	The project leadership mostly accepts the needs for UX design and works towards improving them, but there may still be some skeptical leaders who are not convinced or do not provide enough support.
UXMF.4.5	**UX Processes Integration**
S.4.5.1	The project team has enough resources to implement the organization's goals and user needs.
S.4.5.2	The majority of non-UX roles embrace and support UX design. However, levels of acceptance differ within the organization.
S.4.5.3	The project team formally aligns UX design with the product development process.
S.4.5.4	Various appropriate UX design methods are consistently used within and across projects.
S.4.5.5	The team formally and consistently uses key performance indicators (KPIs) to measure project success.
S.4.5.6	Project quality is consistently maintained across projects, and UX design is always under control.
S.4.5.7	The project's functional areas are clear on their roles in the UX design process, and they know one another's duties.
S.4.5.8	The decision-making process for the project acknowledges that the UX design is

	considered at the product family or portfolio levels.
UXMF.4.6	**UX Outcome**
S.4.6.1	80% of quantitative UX metrics for assessing the created designs' excellence are gathered and debated.
S.4.6.2	60% and above of developers use UCD to predict UX metrics to improve UX maturity.
S.4.6.3	Final projects engage at least 70% of stakeholders.
S.4.6.4	70% of users can accomplish their goals effectively and efficiently while using the project
S.4.6.5	UX drives the project's strategies and decision-making process.
S.4.6.6	UX design works strongly impact project design.

Level 5: Institutionalised

The highest level of FOSS project UX maturity is called "Institutionalised." The UX characterizes this level as part of the global strategy or norm across the FOSS project or the FOSS community. The UX design is tightly integrated into all facets of a project to achieve and maintain the goals of users and projects. Every interaction with the project is constantly tracked and evaluated to help continuously improve UX processes, establish the UX culture, and formulate a UCD UX leadership. As Chapman and Plewes (2014) recommended, UX design is not implemented by UX designers alone but by the entire ecosystem's contribution.

Nevertheless, the UX leadership guides in implementing the strategic roadmap and development activities that assist all stakeholders in selecting viable UX processes. UX design goals are strongly linked to project objectives, and qualitative and quantitative data are used to inform the organization of the path to be taken. As a routine procedure, managers and other project leaders consult the UX designers, users, and other teams before making decisions.

The reputation gained by the FOSS community, which is significant in achieving project goals and other competitive advantages, must be sustainable and maintained at any cost. To maintain this maturity level, instituted UX-related initiatives must be consistent, realistic, and long-term.

Below is an illustration of the measurement tool for the FOSS project's UX maturity at the "Institutionalised" level.

UXMF.5.1	**Usability**					
#	**Statement**	**0**	**1**	**2**	**3**	**4**
S.5.1.1	The project successfully answers user requirements and meets users' expectations.					
S.5.1.2	Regularly collect users' feedback to improve the project quality, especially usability.					
S.5.1.3	The project team is dedicated to usability learning and advancing knowledge in HCI and usability.					
S.5.1.4	The project team regards the UCD approach as a critical strategic asset.					
S.5.1.5	User feedback demonstrates their pleasure and ability to understand the FOSS features quickly.					
S.5.1.6	End users can get interactive guidance to improve their understanding of the software project.					
S.5.1.7	Advanced features are being incorporated to make the FOSS more operational and give users more control.					
S.5.1.8	The project combines standardized usability principles with creative ways to make the					

	user interface beautiful and appealing.
S.5.1.9	The project team regularly monitors, maintains, and enhances the existing service protocol for practical and straightforward reporting of usability defects.
S.5.1.10	The project team continuously experiments with new approaches to improve usability testing.
S.5.1.11	The project documentation is periodically updated to reflect any changes.
UXMF.5.2	**Adopted Technologies**
S.5.2.1	The project team continually adopts user analytics, user research tools, design collaboration tools, prototyping tools, accessibility tools, artificial intelligence, or machine learning.
S.5.2.2	The project team continuously implements technical infrastructures, such as software development tools, libraries, and frameworks, to improve UX consistently.
S.5.2.3	The project team continually implements formal software development methodologies.
UXMF.5.3	**UX Strategy**
S.5.3.1	The project has a sufficient, nearly sufficient, or significant UX budget.
S.5.3.2	The project has a roadmap or plans to achieve UX design goals linked to its objectives.
S.5.3.3	Early project planning always includes UX in design work.
S.5.3.4	The project has enough UX specialists and the expertise to meet all UX design work

	needs
S.5.3.5	Every interaction of stakeholders with the project is constantly tracked and evaluated.
S.5.3.6	All stakeholders are regularly involved, consulted, and systematically receive and discuss user feedback to offer opinions and innovative UX design ideas.
UXMF.5.4	**UX Culture**
S.5.4.1	UX is understood and applied beyond interfaces by including systems and processes.
S.5.4.2	The highest level of the organization's leadership supports UX in the design.
S.5.4.3	Plans are in place to repeat and improve UX design work across the project on all teams.
S.5.4.4	Detailed UX job profiles and career paths for standard UX responsibilities and meta-UX tasks exist.
S.5.4.5	The project team continually develops UX skill sets to help improve UX maturity.
S.5.4.6	The project leadership continually updates needs for UX design towards improving them.
UXMF.5.5	**UX Processes Integration**
S.5.5.1	UX design techniques are utilized outside traditional fields, such as strategy, supply chains, and customer service.
S.5.5.2	Stakeholders without UX roles highly esteem UX and collaborate with those who do UX-related tasks to implement them.
S.5.5.3	UX works are implemented systematically, providing well-established, effective frameworks shared, maintained, and

	improved throughout the business.
S.5.5.4	Before making decisions, project executives consult the UX design teams and other stakeholders.
S.5.5.5	The project team continually and formally aligns UX design with product development.
S.5.5.6	Various appropriate UX design methods are continually improved and consistently used within and across projects.
S.5.5.7	Projects continually improve quality across projects and UX design.
UXMF.5.6	**UX Outcome**
S.5.6.1	Quantified UX metrics always influence decision-making.
S.5.6.2	UX works strongly impact the design quality of the finished project and may always lead to industry and design standards.
S.5.6.3	UX design culture is built and integrated into the project's overall strategy.
S.5.6.4	Final projects continually improve to engage more stakeholders.
S.5.6.5	Users can accomplish their goals effectively and efficiently without support when using the project.

Table 4.27 hereunder lists the total number of assessment statements found in FOSS-UXMM for each theme of UX maturity-influencing factors across all five maturity levels.

Table 4.27: The FOSS-UXMM Framework

UX maturity levels	Number of statements on each theme of UX maturity influencing factors in the FOSS-UXMM						
	UBI	**ATE**	**UXS**	**UXC**	**UXPI**	**UXO**	**Total**
Preliminary				None			0
Acknowledged	11	3	8	5	5	4	36
Accepted	11	3	7	4	5	4	34
Streamlined	11	3	7	6	8	6	41
Institutionalized	11	3	6	6	7	5	38

4.5.4 FOSS-UXMM Performance Scale and Rating Methodology

4.5.4.1 Performance Scale

Each project's performance is rated on a five-point scale: *"Fulfilled," "Largely Fulfilled," "Partially Fulfilled," "Not Fulfilled,"* and *"Unknown"* to ascertain its UX maturity level (see **Table 4.28**). Compared to rating scales adopted by BOOTSTRAP and OS-UMM, the rating scale "Unknown" had been added to boost the FOSS-UXMM flexibility when the fulfillment status of conditions/statements is unknown.

Table 4.28: **Performance Scales and Rating Thresholds**

Rating Scale	Performance Scales			Rating Threshold (%)
	FOSS-UXMM	**OS-UMM**	**BOOTSTRAP**	
4	Fulfilled	Fulfilled	Completely satisfied	≥ 80
3	Largely fulfilled	Largely Fulfilled	Largely satisfied	66.7 – 79.9
2	Partially fulfilled	Partially Fulfilled	Partially satisfied	33.3 – 66.6
1	Not fulfilled	Not fulfilled	Absent/poor	≤ 33.2
0	Unknown	Not Applicable		-

To preserve consistency with already established and proven popular scales, the rating scales and rating thresholds of FOSS-UXMM are the derivatives of the OS-UMM by Raza et al. (2012a) and the BOOTSTRAP by Wang and King (2000). However, the language terms have been somewhat altered following the designed questionnaires. Overall, based on their level of agreement with the statements in the questionnaires, owners, and developers can assess the UX maturity of the FOSS projects using the study's rating methodology. In this sense, the study employs the self-assessment technique.

4.5.4.2 Rating Methodology

The FOSS-UXMM's rating methodology was partially derived from the OSS-UMM by Raza et al. (2012a) and the BOOTSTRAP algorithm by Wang and King (2000). It has adopted terms like UX Maturity Rating (UXMRt), Number of Fulfilled Statements (NFS), Passing Threshold (PT), and UX Maturity Level (UXML).

Let UXMRt[i,j] represent a rating of the j^{th} maturity level in the i^{th} UX maturity impacting factor. Consequently, it can be summed up as follows

UXMRt [i, j] = 4 if the fulfillment of the condition/statement is at least 80%
= 3 if the fulfillment of the condition/statement is from 66.7 to 79.9%
= 2 if the fulfillment of the condition/statement is from 33.3 to 66.6%
= 1 if the fulfillment of the condition/statement is less than 33.3%
= 0 if the fulfillment of the condition/statement is unknown.

An i^{th} condition/statement at the j^{th} maturity level is deemed fulfilled if UXMRt [i, j] ≥ 3 or UXMRt [i, j] is 0. The number of conditions/statements fulfilled at the j^{th} maturity level is defined as:

NFS [j] = Number of {UXMRt [i, j] | Fulfilled}
= Number of {UXMRt [i, j] | UXMRt [i, j] ≥ 3 or UXMRt [i, j] = 0}

If 80% of the conditions or statements in the questionnaire are fulfilled, UX maturity is deemed to have been attained. Therefore, the passing threshold at the j^{th} maturity level is defined as PT [j] = TN [j] * 80% if TN [j] is the total number of statements at the j^{th} maturity level. See **Table 4.29** for the total number of questions for the passing threshold of 80% at each UX maturity, with values rounded to the nearest tenth.

Table 4.29: **Rating Threshold for the FOSS UX Maturity Assessment**

UX Maturity Level	Total Questions	Passing Thresholds at 80%
Preliminary	0	Unknown
Acknowledged	36	29
Accepted	34	27
Streamlined	41	33
Institutionalized	38	30

The highest maturity level at which the number of satisfied conditions or statements is more than or equal to the passing threshold PT [j] is the UX Maturity Level (UXML). Thus, UXML = max{ j | NFS [j] ≥ PT [j]}.

4.5.5 Reliability and Validity of FOSS-UXMM

Before the actual deployment of FOSS-UXMM, the study conducted a pilot study to analyze the validity and reliability of the constructs. Thirty-three (33) FOSS projects from FOSS repositories, particularly SourceForge.net, with an activity level of 90% and above and recently updated, were involved in this study. The projects were grouped into seven categories: accounting (4), office (3), project management (5), games and entertainment (4), knowledge management (3), development (2), and scientific and engineering (12).

Subsequently, the study administered the questionnaire by contacting project managers, developers, and owners to determine how much they agreed with each contained statement. Each participant received a personalized e-mail outlining the goals and scope of the study along with the consent request to participate in the survey. The recruitment process occurred between mid-July 2021 and November 2022, with an initial response rate of 25.53% (24/94). Finally, 18 FOSS stakeholders agreed to participate and signed consent forms.

Internal consistency analysis was adopted using the coefficient alpha by Cronbach (1951) to test the reliability and validity of constructs. In the final analysis, the coefficient alpha ranged from 0.702 to 0.923. (see **Table 4.30**), which confirms an acceptable internal consistency level based on Nunnally et al.(1994) and Taber's (2018) endorsements. Therefore, referencing the principles for reliability, all the items developed for this investigation are reliable.

Table 4.30: Reliability Analysis Using Coefficient Alpha

UX maturity levels	Reliability coefficients of each category of UX maturity influencing factors in the assessment questionnaire					
	UBI	**ATE**	**UXS**	**UXC**	**UXPI**	**UXO**
Preliminary	Unknown					
Acknowledged	0.702	0.894	0.748	0.886	0.762	0.879
Accepted	0.746	0.860	0.791	0.749	0.870	0.895
Streamlined	0.754	0.773	0.860	0.796	0.855	0.923
Institutionalized	0.730	0.801	0.841	0.920	0.771	0.712

Nevertheless, the study performed the Principal Components Analysis (PCA) by (Comrey & Lee, 2013) for all thirty-six UX maturity factors in each maturity level and reported the findings. The performed analysis intended to reveal the elements' convergence validity to assist in deciding what variables to keep and what to eliminate. Convergent validity always occurs when scaling items correlate and move in the same direction (Campbell & Fiske, 1959). The Eigenvalue one-criterion, also called the Kaiser Criterion (Kaiser, 1960), was used to examine construct validity as a reference point in the study. Based on recommendations, studies always retain components with an Eigenvalue larger than one. From the completed analysis, items in the questionnaires form entirely a single factor with Eigenvalues greater than one (see **Table 4.31**). As a result, the convergent validity of the FOSS-UXMM tools can be deemed sufficient.

Table 4.31: Convergent Validity of Factors Using Eigenvalue

UX maturity levels	Eigen Value of each group of UX maturity influencing factors in the assessment questionnaire					
	UBI	ATE	UXS	UXC	UXPI	UXO
Preliminary	Unknown					
Acknowledged	2.058	1.833	1.574	1.248	1.110	1.021
Accepted	2.912	2.111	2.052	1.656	1.387	1.011
Streamlined	2.054	1.949	1.688	1.596	1.348	1.025
Institutionalized	2.391	2.002	1.716	1.612	1.213	1.032

4.5.6 Discussions of FOSS-UXMM

In software engineering, UX maturity models gather detailed information about the strengths and weaknesses in implementing various UX-related processes and activities. Organizations, including the FOSS community, can use data from UX maturity assessment to identify their status quo and propose strategies for improvement. In addition to examining the relevancy of present UXCMMs in the FOSS community, this study has established the FOSS community's UX maturity influencing factors, the analytical lenses of determining the level of preparation to achieve higher UX maturity levels, develop FOSS-UXMM and validate it by completing two case studies in actual FOSS projects. The assessment framework (PRM) of FOSS-UXMM was derived from the thirty-six factors grouped on six themes collected from literature review and consultations with experts and other FOSS stakeholders.

The FOSS-UXMM is distinct from all other UXCMMs, and it has been designed exclusively for the FOSS community while accounting for the surrounding dynamics. It has been created experimentally with the help of a novel model derived from several models, guidelines, and checklists, including de Bruin et al.(2005), Namayala et al. (2022), Hevner et al. (2004), Pefers (2007), Becker et al. (2010) and Dresch et al. (2015). The methodologies adopted for evaluating each UX maturity level are simple to comprehend and apply. Unlike other UXCMMs, FOSS-UXMM, to a large extent, accurately captures the dynamics of the FOSS community.

Moreover, the FOSS-UXMM has been tested in the real world using case studies. It is integrated with each level's UX maturity assessment tools, derived from well-known rating scales and performance thresholds such as BOOTSTRAP. As a result, anyone can utilize it, even those who are not UX assessment professionals, which bypasses several problems, including those identified by Hanson (2017) and Traynor (2022).

4.6 Examining the Practical Applicability of the FOSS-UXMM in the FOSS Community

4.6.1 Overview

A critical challenge of the present UXCMMs is the lack of practical applicability in natural settings, particularly within the FOSS community (Namayala et al., 2022; Raza et al., 2012a; Terry et al., 2010). Existing UXCMMs do not adequately outline how to address the FOSS community's unique dynamics, characteristics, and culture. Other challenges include the complexity of their assessment procedures, and they use jargon and complicated language that non-UX professionals may not understand well (Hanson, 2017; Traynor, 2022).

This study has instituted efforts to test how well the proposed UX maturity assessment methodology suits the FOSS community to avoid replicating similar challenges repeatedly by completing two case studies in two desktop-based FOSS projects (Project I and II). The completed case studies tested the questionnaire's applicability, deepened understanding of the assessment method, and determined the generalizability of FOSS-UXMM to every desktop-based FOSS project and stakeholder. The inclusions/exclusions criteria for the selected projects include projects that were conscious of and realized the importance of UX and its maturity assessment.

4.6.2 Completed Case Studies

4.6.2.1 Introduction

Although no appropriate definitions for case studies exist, they refer to a thorough investigation into a person, a group of people, or a unit to generalize findings over several teams (Gustafsson, 2017). It is a rigorous, systematic analysis of a single person, group, community, or other units in which the researcher looks at extensive data about several variables (Whitfield, 1995). The case studies also explore complicated events in a natural context to deepen understanding (McCorcle & Bell, 1986).

4.6.2.2 Assessment Methodology

Delivering products with desirable UX is an organizational endeavor (Chapman & Plewes, 2014; Cheng & Guo, 2018). Every product, intentionally or unintentionally, exhibits some UX features (Kashfi et al., 2017). Projects included in this study were carefully chosen from the FOSS source codes hosting facilities.

In addition, participants were informed the assessment was part of a PhD study and that neither their names nor projects would be revealed in the thesis or any other subsequent publications. The study used the FOSS-UXMM questionnaires to assess the UX maturity of each selected project. Finally, following the standard practice in the FOSS community, the study communicated with participants via email, the mailing list, online forums, and other survey links. Each project had multiple responses; the average score was computed as the overall response after analyzing the inter-rater agreement among the raters using Kendall’s coefficient of concordance (W). As a result, there was minimal bias in arriving at the conclusive results.

4.6.2.3 The Case Study in Project I

Project one deals with interactive visualizations of structural and temporal patterns and trends in a scientific topic and is under the education category in SourceForge.net. It had more than 3,961 weekly downloads, with an activity level of 90%. Its mailing list had over 10,000 subscribers discussing user-related issues and offering assistance and support. The degree to which “project I" respondents agreed or disagreed with the assertions made in the prescribed questionnaires at each level for evaluating UX maturity is provided in **Table 4.32**.

Table 4.32: Assessment Details of Case Study in Project I

Acknowledged Level -2		Accepted Level -3		Streamlined Level -4		Institutionalized Level -5	
Q#.	**Value**	**Q#.**	**Value**	**Q#.**	**Value**	**Q#.**	**Value**
S.2.1.1	3	S.3.1.1	2	S.4.1.1	2	S.5.1.1	4
S.2.1.2	4	S.3.1.2	3	S.4.1.2	3	S.5.1.2	3
S.2.1.3	3	S.3.1.3	2	S.4.1.3	3	S.5.1.3	2
S.2.1.4	3	S.3.1.4	3	S.4.1.4	3	S.5.1.4	3
S.2.1.5	3	S.3.1.5	3	S.4.1.5	2	S.5.1.5	3
S.2.1.6	3	S.3.1.6	3	S.4.1.6	3	S.5.1.6	3
S.2.1.7	3	S.3.1.7	2	S.4.1.7	4	S.5.1.7	3
S.2.1.8	4	S.3.1.8	4	S.4.1.8	4	S.5.1.8	2
S.2.1.9	4	S.3.1.9	2	S.4.1.9	2	S.5.1.9	4
S.2.1.10	2	S.3.1.10	2	S.4.1.10	2	S.5.1.10	2
S.2.1.11	3	S.3.1.11	2	S.4.1.11	2	S.5.1.11	3
S.2.2.1	3	S.3.2.1	3	S.4.2.1	2	S.5.2.1	3
S.2.2.2	3	S.3.2.2	2	S.4.2.2	2	S.5.2.2	2
S.2.2.3	3	S.3.2.3	4	S.4.2.3	3	S.5.2.3	3
S.2.3.1	3	S.3.3.1	2	S.4.3.1	3	S.5.3.1	3
S.2.3.2	2	S.3.3.2	2	S.4.3.2	2	S.5.3.2	3
S.2.3.3	3	S.3.3.3	3	S.4.3.3	3	S.5.3.3	2
S.2.3.4	3	S.3.3.4	3	S.4.3.4	2	S.5.3.4	4
S.2.3.5	3	S.3.3.5	3	S.4.3.5	3	S.5.3.5	4
S.2.3.6	3	S.3.3.6	3	S.4.3.6	2	S.5.3.6	2
S.2.3.7	3	S.3.3.7	2	S.4.3.7	2	S.5.4.1	2
S.2.3.8	4	S.3.4.1	3	S.4.4.1	1	S.5.4.2	3
S.2.4.1	3	S.3.4.2	1	S.4.4.2	3	S.5.4.3	2
S.2.4.2	2	S.3.4.3	2	S.4.4.3	4	S.5.4.4	1
S.2.4.3	3	S.3.4.4	3	S.4.4.4	3	S.5.4.5	3
S.2.4.4	4	S.3.5.1	3	S.4.4.5	2	S.5.4.6	1
S.2.4.5	4	S.3.5.2	2	S.4.4.6	3	S.5.5.1	2
S.2.5.1	3	S.3.5.3	3	S.4.5.1	3	S.5.5.2	2

Acknowledged Level -2		Accepted Level -3		Streamlined Level -4		Institutionalized Level -5	
Q#.	**Value**	**Q#.**	**Value**	**Q#.**	**Value**	**Q#.**	**Value**
S.2.5.2	3	S.3.5.4	3	S.4.5.2	3	S.5.5.3	3
S.2.5.3	4	S.3.5.5	2	S.4.5.3	2	S.5.5.4	3
S.2.5.4	4	S.3.6.1	2	S.4.5.4	3	S.5.5.5	3
S.2.5.5	4	S.3.6.2	4	S.4.5.5	2	S.5.5.6	3
S.2.6.1	2	S.3.6.3	3	S.4.5.6	2	S.5.5.7	2
S.2.6.2	3	S.3.6.4	3	S.4.5.7	4	S.5.6.1	4
S.2.6.3	2			S.4.5.8	2	S.5.6.2	2
S.2.6.4	3			S.4.6.1	2	S.5.6.3	3
				S.4.6.2	3	S.5.6.4	3
				S.4.6.3	3	S.5.6.5	2
				S.4.6.4	2		
				S.4.6.5	4		
				S.4.6.6	2		
NFS	31	NFS	19	NFS	21	NFS	23

With the adopted rating system, the study regards statements as agreed upon if respondents assign them with a value greater than or equal to 3 or 0. The study has determined each level's Number of Fulfilled Statements (NFS). As a result, NFS is 31 at Level 2, 19 at Level 3, 21 at Level 4, and 23 at Level 5. According to the rating scale and passing thresholds, NFS at Level 2 has a pass threshold of 80%. Consequently, this project is at Level 2, **"Acknowledged,"** of the FOSS-UXMM.

4.6.2.4 The Case Study in Project II

The second project, "Project II," belongs to the database class that offers users SQL syntax hints and highlights by providing code formatting and syntactic mistakes. The project has more than 242 users download each week, and all downloads give it positive reviews. Its user mailing list has over 500 subscribers who actively engage in various topics and discuss project improvements.

After collecting and analyzing responses from involved stakeholders, the extent to which each developer, manager, and contributor rated each statement in the tool is well described in **Table 4.33**.

Table 4.33: **Assessment Details of Case Study in Project II**

Acknowledged Level -2		**Accepted Level -3**		**Streamlined Level -4**		**Institutionalized Level -5**	
Q#.	**Value**	**Q#.**	**Value**	**Q#.**	**Value**	**Q#.**	**Value**
S.2.1.1	3	S.3.1.1	2	S.4.1.1	3	S.5.1.1	2
S.2.1.2	4	S.3.1.2	3	S.4.1.2	3	S.5.1.2	2
S.2.1.3	3	S.3.1.3	4	S.4.1.3	3	S.5.1.3	2
S.2.1.4	1	S.3.1.4	3	S.4.1.4	3	S.5.1.4	4
S.2.1.5	3	S.3.1.5	4	S.4.1.5	4	S.5.1.5	1
S.2.1.6	3	S.3.1.6	2	S.4.1.6	4	S.5.1.6	4
S.2.1.7	4	S.3.1.7	2	S.4.1.7	4	S.5.1.7	1
S.2.1.8	2	S.3.1.8	2	S.4.1.8	4	S.5.1.8	2
S.2.1.9	1	S.3.1.9	3	S.4.1.9	3	S.5.1.9	2
S.2.1.10	1	S.3.1.10	4	S.4.1.10	4	S.5.1.10	2
S.2.1.11	2	S.3.1.11	4	S.4.1.11	2	S.5.1.11	3
S.2.2.1	4	S.3.2.1	2	S.4.2.1	4	S.5.2.1	1
S.2.2.2	3	S.3.2.2	1	S.4.2.2	3	S.5.2.2	3
S.2.2.3	3	S.3.2.3	2	S.4.2.3	2	S.5.2.3	3
S.2.3.1	3	S.3.3.1	3	S.4.3.1	1	S.5.3.1	3
S.2.3.2	4	S.3.3.2	2	S.4.3.2	2	S.5.3.2	1
S.2.3.3	1	S.3.3.3	2	S.4.3.3	4	S.5.3.3	3
S.2.3.4	4	S.3.3.4	1	S.4.3.4	2	S.5.3.4	2
S.2.3.5	3	S.3.3.5	2	S.4.3.5	1	S.5.3.5	4
S.2.3.6	2	S.3.3.6	4	S.4.3.6	3	S.5.3.6	2
S.2.3.7	3	S.3.3.7	4	S.4.3.7	4	S.5.4.1	2
S.2.3.8	2	S.3.4.1	1	S.4.4.1	1	S.5.4.2	1
S.2.4.1	2	S.3.4.2	3	S.4.4.2	2	S.5.4.3	1
S.2.4.2	3	S.3.4.3	3	S.4.4.3	3	S.5.4.4	3
S.2.4.3	4	S.3.4.4	2	S.4.4.4	4	S.5.4.5	4

Acknowledged Level -2		Accepted Level -3		Streamlined Level -4		Institutionalized Level -5	
Q#.	**Value**	**Q#.**	**Value**	**Q#.**	**Value**	**Q#.**	**Value**
S.2.4.4	4	S.3.5.1	4	S.4.4.5	3	S.5.4.6	4
S.2.4.5	2	S.3.5.2	2	S.4.4.6	1	S.5.5.1	3
S.2.5.1	1	S.3.5.3	4	S.4.5.1	2	S.5.5.2	1
S.2.5.2	2	S.3.5.4	2	S.4.5.2	2	S.5.5.3	1
S.2.5.3	2	S.3.5.5	3	S.4.5.3	2	S.5.5.4	1
S.2.5.4	2	S.3.6.1	4	S.4.5.4	3	S.5.5.5	1
S.2.5.5	3	S.3.6.2	1	S.4.5.5	1	S.5.5.6	1
S.2.6.1	2	S.3.6.3	3	S.4.5.6	1	S.5.5.7	3
S.2.6.2	2	S.3.6.4	2	S.4.5.7	3	S.5.6.1	4
S.2.6.3	3			S.4.5.8	2	S.5.6.2	1
S.2.6.4	2			S.4.6.1	1	S.5.6.3	3
				S.4.6.2	1	S.5.6.4	2
				S.4.6.3	3	S.5.6.5	3
				S.4.6.4	1		
				S.4.6.5	4		
				S.4.6.6	2		
NFS	**19**	**NFS**	**17**	**NFS**	**22**	**NFS**	**16**

The numerical values indicate how the developers, contributors, and project owners have rated Project II using the questionnaires for each maturity level. From the analysis of the collected ratings, NFS at Level 2 is 19, Level 3 is 17, Level 4 is 22, and Level 5 is 16. According to the adopted rating thresholds, the observed values of NFS indicate that the project has not passed the requirements of any of the assessed levels: Level 2, Level 3, Level 4, and Level 5. Therefore, Project II is at Level 1 of maturity, the **"Preliminary."**

The summary below (see **Table 4.34**) for completed case studies rigorously explains how the author interpreted the results.

Table 4.34: Summary of Case Studies

Level	Total Questions	# of questions for passing the 80% threshold	Project-I NFS	Project-II NFS
Preliminary	0	Unknown	Unknown	Unknown
Acknowledged	36	29	31	19
Accepted	34	27	19	17
Streamlined	41	33	21	22
Institutionalized	38	30	23	16

4.6.2.5 The Inter-Rater Agreement Analysis (IRA)

Although the FOSS-UXMM can produce compelling results with just one rater, the findings are more efficient when numerous raters are used. The ideal number of raters for optimal outcomes is currently open for debate. When engaging multiple raters, like in the completed case studies, assessing the level of agreement between raters (The Inter-Rater Agreement Analysis) is mandatory before presenting the final results. The Inter-Rater Agreement Analysis (IRA) refers to the degree to which various raters rate each object with an identical value (Gisev, Bell, & Chen, 2013). In reality, multiple respondents from a single sample may have different perspectives about a similar project's aspects (El Emam, 1999).

The statistical literature disagrees on the uses and suitability of various IRA and Inter-Rater Reliability Analysis (IRR) indices and their derivatives (Gisev et al., 2013). Some studies specifically used to quantify IRA, such as Bland-Altman plots, while others, like Cohen's kappa, are used to measure both IRA and IRR. However, the final figures are not considerably different. According to Gisev et al.(2013), four critical issues must be considered when selecting appropriate measures for IRR and IRA. They include the goal of the analysis, the significance of absolute value or trend in ratings, the type of the variable examined, and the number of raters engaged.

According to the nature, the number of raters, the type of data collected in the study's case studies, and the framework proposed by Gisev et al.(2013), the Intraclass Correlation Coefficient (ICC) or Kendall coefficient of concordance are the more effective measures of the IRA. Therefore, this work has carried out an IRA using the Kendall coefficient of concordance (W). For example, the completed case studies involved 15 and 10 respondents from Projects I and II, respectively. **Table 4.35** displays the Kendall statistics for Project I, whereas **Table 4.36** lists the Kendall statistics for Project II.

The W values can range from 0 to 1, with 0 denoting utter discord and 1 denoting perfect agreement (Landis & Koch, 1977). The higher values of W indicate a stronger association among raters. According to Schmidt (1997), other values of W can be interpreted as: 0.1 denotes very weak agreement (none), 0.3 denotes a weak agreement (low), 0.5 denotes a moderate agreement (fair), 0.7 denotes a strong agreement (high), and 0.9 denotes a robust agreement (very high)

Table 4.35: **The Inter-Rater Agreement Analysis of Project I**

UX Maturity Level	**Kendall's Coefficient of Concordance**	
	Coefficient	**X^2**
Acknowledged	0.915	247.108*
Accepted	0.957	244.084*
Streamlined	0.927	250.379*
Institutionalized	0.937	224.825*

* P < 0.001; ** P < 0.005

From the W values calculated in completed case studies and results presented in **Table 4.35** and **Table 4.36**, the agreement among raters of both **"Project I"** and **"Project II"** is substantial.

Table 4.36: **The Inter-Rater Agreement Analysis of Project II**

UX Maturity Level	**Kendall's Coefficient of Concordance**	
	Coefficient.	**X^2**
Acknowledged	0.918	165.292*
Accepted	0.901	135.134*
Streamlined	0.913	182.583*
Institutionalized	0.941	150.633*

* P < 0.001; ** P < 0.005

4.7 The Comparison of FOSS-UXMM with Other UXCMMs

4.7.1 Overview

Although comparative studies of the existing UXCMMs are tedious and complicated, their absence creates a methodological gap that makes it more difficult to qualify them (Jokela et al., 2006). Comparative studies have several advantages. For example, they may assist in avoiding the need to reinvent the wheel, prevent practitioners from sifting through the “methodological jungle” to find an acceptable model, and bridge a research gap. They may also help to bring together previously distinct lines of work that may provide more insights to support future improvements.

Comparative studies would result in favorable outcomes. For example, they might help uncover answers to several questions, such as: Are the FOSS-UXMM and other UXCMMs derived from the same or different research backgrounds? Are there any notable distinctions among them? Does the FOSS-UXMM have unique or similar Key Performance Indicators (KPIs) compared to other UXCMMs? Does FOSS-UXMM employ different or similar assessment techniques compared to other UXCMMs? Finally, are there direct connections between the FOSS-UXMM and other UXCMMs? Answers derived from the above questions may be instructive because they may provide students with alternatives when choosing UXCMMs by giving adequate information to choose preferred models independently. As a result, they prevent instructors from infringing on students’ autonomy by planting specific UXCMMs and ignoring alternative models. Comparison is always performed on like things. The author clearly understands models from various perspectives will never be similar; therefore, they might not be equivalent. As a result, only UXCMS that depict the same views will be compared to account for this crucial factor.

4.7.2 The Methodology for Comparative Analysis

The study analyzed several articles that provide analytical frameworks (analytical lenses) and guidelines on systematically comparing IS to build a practical approach for analyzing UXCMMs. For example, a framework by Farooq et al.(2014) compares the analysis of programming language to locate a suitable language that users can learn as the First Programming Language (FPL). A Paradigmatic Analysis

Contrasting Information Systems Development Approaches and Methodologies by Iivari, Hirschheim, and Klein (1998), and Dimensions of information systems design: A Framework for a long-range research Program by Iivari (1986).

Other analytical frameworks analyzed include A Social Action Perspective of Information Systems Development by Lyytinen, Boland, and Hirschheim (1987), Different Perspectives on Information Systems: Problems and Solutions by Lyytinen(1987), Criteria for Measuring and Comparing Information Systems by Palmius (2007), New directions on agile methods: A comparative analysis by Abrahamsson et al. (2003b), and From information behavior research to the design of information systems: the Cognitive Work Analysis framework by Fidel and Pejtersen (2004).

Theoretically, analytical lenses support the adequate analysis of different information systems, including models where UXCMMs are not exceptional and intend to provide a broad overview of the organizational characteristics one could investigate on each UXCMM (Coral & Bokelmann, 2017; Jokela et al., 2006). However, providing analytical lenses for effectively comparing contemporary UXCMMs is difficult since they lack common vocabulary, and establishing a link between them is difficult. This conclusion confirms the findings of Monteiro and Maciel (2020). However, this study has adopted lenses, such as approach, empirical evidence, and concrete guidance, commonly adopted by other studies like Namayala et al.(2022), Jokela et al.(2006), and Lacerda and von Wangenheim (2018). Abrahamsson et al. (2003b) and Avison and Fitzgerald (1995) refer to these lenses as “generic” and “applicable” to any comparative analysis of IS.

Deriving the final list of analytical lenses mandated the study to undergo several iterative tests. The tests objected to establishing the relevance of identified lenses in the comparative analysis of encountered UXCMMs. Finally, the analytical lenses listed in **Table 4.37** were helpful for this work in comparing the basic features of UXCMMs and FOSS-UXMM.

Table 4.37: Final Analytical Lenses

Lens	Descriptions
Identity	Examine models' unique identifiers, such as title, author(s), year of publication, publication type, theme (s) explored (capability, maturity, capability and maturity, and framework or UX strategy), citations, and the number of levels. It also reveals the motives behind model creation.
Approach	Examine the overall structure of building the model. It inspects whether a model is based on a well-known method or is a novel concept.
Inputs analysis	Examines how inputs to the model were acquired and processed. Is the process involved transparent? Which technique or software is used to analyze inputs? For example, does input analysis employ statistical or fuzzy mathematics?
FOSS Practical applicability	Examine the status of the actual application of UXCMMs in natural settings of the FOSS projects and how the models were verified and validated.
Empirical evidence	According to Lacerda and von Wangenheim (2018), empirical evidence examines models' practical research efforts and their benefits to organizations. Moreover, how well do models perform? Are they based on experimentation, observation, or specific behavior and patterns documentation?
Integrate assessment tool	It examines if the UXCMMs provide easy-to-use self-administered tools for assessing the UX maturity of each level.
Scope	Focuses on the models' objectives (descriptive, prescriptive, or comparative) and occasionally considers the pertinent domains and situations. It analyzes the sets of process areas and the rating scales.
Concrete guidance	Examines how UXCMMs provide practitioners with explicit and extensive guidance that is easy to understand and follow.
Operations	Examine if anybody can apply the models, even those not experts in UX maturity assessment. How complex are the models? Is the method of evaluation well communicated? Are the rating scale and performance thresholds established?

The vast array of UXCMMs in the corpus of knowledge prevents this study from conducting a side-by-side comparison with each one. Instead, the study has focused on demonstrating how the proposed model is superior to alternatives in assessing the UX maturity of the FOSS community.

4.7.3 What Differentiates FOSS-UXMM From Other UXCMMs

Identity

Unlike other UXCMMs, FOSS-UXMM has been named uniquely to mimic the domain and prescribed functions. Its name explains the assessment of UX maturity in the FOSS community by evaluating the UX maturity of individual projects. It does so by seeking an agreement to statements regarding implementing UX maturity-related activities. Like CMM by Paulk et al. (1993), the FOSS-UXMM has adopted the 5-scale maturity. Nevertheless, the FOSS-UXMM was developed in response to the problem of having a UX maturity model that cannot appropriately and experimentally account for the dynamism of the FOSS community, an issue that none of the existing UXCMMs has attempted to solve (Namayala et al., 2022).

Approach

The FOSS-UXMM's structure and assessment methodology are based on existing concepts. For example, it has adopted 5-levels of maturity like CMM by Paulk et al. (1993) and OS-UMM by Raza et al. (2012a). It has also adopted rating and passing thresholds from the BOOTSTRAP algorithm by Wang and King (2000). However, FOSS-UXMM has developed a unique PRM that contains influencing factors confirmed to be relevant for the FOSS community. Included factors are rigorously tested within the FOSS community and have received endorsements from UX experts.

Inputs analysis

Unlike other UXCMMs, inputs to the FOSS-UXMM were acquired and processed empirically. The study used the fuzzy Delphi method to establish inputs of the FOSS-UXMM. The technique was very transparent and involved UX assessment experts from multiple locations. The study adopted statistical analysis to analyze consolidated inputs; only statistically significant ones are included in the model. Many existing UXCMMs lack proper documentation explaining how their inputs were collected and analyzed, leaving the assumption they have been developed from owners' years of working experience.

FOSS Practical applicability

Unlike other UXCMMs, the FOSS-UXMM has been tested in the actual setting in the completed case studies that determined the UX maturity of two projects. The adopted assessment methodology perfectly fits in the FOSS community because it only demands inputs from developers, contributors, and other project owners who are versed in technical skills and have knowledge of UX maturity. Present UXCMMs have not been tested in the FOSS community (Namayala et al., 2022; Terry et al., 2010). As a result, it is unclear if they can perfectly fit the dynamics of the FOSS community.

Empirical evidence

Unlike many UXCMMs, the FOSS-UXMM has been developed from practical research efforts, and therefore, it has a bigger chance of benefiting the FOSS community. Except for the works of Sakhardande and Thanawala (2014), Hanson (2017), Rukonić et al.(2019), and Molich (2021), several other UXCMMs lack empirical evidence (Lacerda & von Wangenheim, 2018; Namayala et al., 2022). Although FOSS-UXMM still needs more work to be fine-tuned, it addresses the UX assessment of the FOSS community much better compared to other initiatives. It is less complex, requires no close monitoring to be effective, and can easily be used by people with primary knowledge of software development.

Integrate assessment tool

Unlike many other UXCMMs, the FOSS-UXMM provides an easy-to-use self-administered tool for assessing the UX maturity of each level. Regardless of the project size, anybody who wants to determine the UX maturity of FOSS projects can quickly adopt and use the tool without support. The tool provided by the FOSS-UXMM is fact-based rather than opinion-based.

Scope

The FOSS-UXMM is descriptive because it highlights the critical areas for improvement, prescriptive because it suggests measures for improvement, and has been exclusively developed for assessing the UX maturity of the FOSS community projects, particularly desktop-based projects. It precisely defines its PRM, rating scale, and development methodology. However, current UXCMMs promise to apply to any type and size of company, huge or tiny, sophisticated or straightforward, and closely regulated, such as health care. In practice, however, no UXCMM could fully and holistically capture these qualities. Although current UXCMMs share some commonalities, many provide little or no information on what constitutes a matured UX system, which matches findings from the Sauro et al.(2017) study. They do not explicitly state their PRMs, ordinal scales, development procedures, or essential process areas.

Concrete guidance

Unlike many other UXCMMs, the FOSS-UXMM provides practitioners with explicit and extensive guidance that is easy to understand and follow to self-assess their projects in the FOSS community. Concrete guidelines help assessors decrease the likelihood of confusion when using, adopting, and applying the FOSS-UXMM assessment methodology. As a result, the assessment outcomes are fact-based rather than the assessors' opinions. However, present UXCMMs have limited or no information regarding how they can efficiently be adopted, and those with little guidance lack training schemes. Therefore, the actual application of present UXCMMs necessitates much interpretation and does not always guarantee the results that the authors intended.

Operations

Unlike other UXCMMs, the FOSS-UXMM can be applied by anyone with average knowledge of UX and their assessments, and it does not require special skills from experts in UX maturity assessment. Nevertheless, FOSS-UXMM is not complex and communicates its evaluation method, rating scale, and performance thresholds to users. Adopting proven key UX maturity factors distinguishes FOSS-UXMM from other models in operations.

4.7.4 Discussion of the Findings from Comparative Analysis of the FOSS-UXMM

From the completed comparative analysis, the study has found a significant improvement in the model for assessing the UX maturity of the FOSS community. The proposed method, FOSS-UXMM, is more realistic in evaluating the UX maturity of FOSS projects by accounting for known dynamics as recommended by scholars. However, it is also subject to further improvement and broader deployments for generic applicability. Nevertheless, by introducing the FOSS-UXMM, the study has assisted the FOSS community by providing a consistent framework that may be adopted across projects to offer the harmonized process for determining UX maturity.

Nevertheless, the completed analysis has shed light on areas that need improvements and special attention from future researchers. However, to date, UXCMM's comparative analysis is missing, attracting more challenges in identifying if the UXCMMs are improving toward assessing the UX, particularly in the FOSS community. This is the first comprehensive study comparing UXCMMs, and its findings complement the earlier findings in the comparative analysis of UXCMMs.

4.8 Summary of the Chapter

This chapter has analyzed the data collected from the research questionnaires, field observations, documentary reviews, interviews, consultations with experts, and case studies to provide the results. From the findings, the chapter has presented analytical lenses to determine the preparation of desktop-based FOSS projects for UX maturity, established factors influencing UX maturity in desktop-based FOSS projects, develop the UX maturity model (FOSS-UXMM), and performed the comparative analysis between FOSS-UXMM and other UXCMMs.

All questionnaire variables were found significant after rigorously adopting the statistical tests, including Internal consistency analysis, convergent validity analysis, hypothesis testing, IRR, and IRA. The proposed FOSS-UXMM, which can assist the decision-makers by contributing the methodology to optimize the design for UX of desktop-based FOSS projects, was validated by experts and demonstrated using case studies to be a successful tool in evaluating the UX maturity in the FOSS community.

Chapter five is allocated for the conclusion, recommendations, and way forward resulting from the analysis and discussion of the findings presented in this chapter.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 An Overview

This work divides this chapter into six subsections, where subsection two summarises the study findings in connection to its research questions. Subsection three presents the study's implications for research and practice. Subsection four offers the study's contributions, subsection five demonstrates the study's conclusion, and finally, the study closes the chapter with recommendations for future studies.

5.2 Summary of the Findings

The exhausted literature review has generally found that researchers have relatively given inadequate attention to the FOSS projects' UX design and assessment. Generally, the FOSS community has not correctly addressed UX design-related issues and is still developing desktop-based FOSS projects with undesirable UX, making them not widely adopted. This study has developed a model as a contribution to the body of knowledge that may reasonably help the FOSS community self-evaluate the UX maturity of its projects**.** While developing this model, the study has answered six research questions formulated in the fifth section of Chapter One. The mapping between research questions and answers based on findings is as follows:

RQ-1 How applicable are the current UXCMMs in determining the UX maturity of desktop-based FOSS projects?

The findings from the SLR study of Namayala et al. (2022), which is part of this study, show the current UXCMMs are generically developed and may not be applicable in the FOSS community because the FOSS community differs from other software-developing communities and have unique cultures, dynamics, and variables that may need novel treatments. This finding confirms the findings of Raza et al. (2012a).

RQ 2 What analytical lenses may analyze the preparation of the desktop-based FOSS projects for UX maturity?

Although this study's findings confirm the presence of elementary initiatives for preparing the FOSS community to assess its UX maturity, such as adopting ad hoc UX strategies, examining factors influencing the adoption of desktop-based FOSS projects, and evaluating UX maturity using generically developed UXCMMs, the measures do not attest to a complete preparation. With the current setup, it is difficult to tell whether the community is prepared for UX maturity. The preparation status of the FOSS projects is perceived as infancy, inconsistent, results from personal views, and is based on opinions rather than facts.

Having strategies, such as analytical lenses, may consistently help in knowing the extent of preparation of the FOSS community for UX maturity, improve the design for UX, and consequently deliver desktop-based FOSS projects that may likely meet users' expectations and achieve higher levels of UX maturity. In turn, the analytical lenses may make the FOSS community well-informed about the prerequisites before it can be called a UX-mature community. However, little information is available in analytical lenses for assessing the preparation of desktop-based projects for UX maturity in the literature.

As a result, the FOSS community fosters many factors that make it an unpleasant place for UX/HCI designers and practices. For example, UX professionals are not accepted into executive teams. Many FOSS projects in the community have insufficient UX experts vested with executive decision-making authority. They unclearly allocate UX tasks for each stakeholder, which fails to assign separate responsibilities to stakeholders and track their progress. Nevertheless, current practices in the FOSS community unvalue UX maturity assessment-related activities in the same manner as core functions and source code aesthetics, leading to decision-makers not supporting their implementation. Finally, understanding UX maturity assessment concepts across the FOSS community, projects, and stakeholders is inconsistent, which brings in different views and priorities and hinders their implementation.

This study conducted empirical research reported in section three of Chapter Four to establish ten (10) analytical lenses that may consistently analyze the preparation of the FOSS projects for UX maturity to break the deadlock and respond to the second research question. These analytical lenses comprise the availability of UX strategic plans, accommodate UX professionals as decision-makers, clear UX roles for each stakeholder, value the UX-related tasks, adopting UCD practices in design, decision-makers buy-in and support UX assessment initiatives, availability of FOSS-exclusive UX maturity assessment tools, accommodate UX principles in FOSS design, uniform understanding of UX concepts, and quantify UX maturity assessment metrics.

The methodology adopted by this study to establish the mentioned lenses, the fuzzy Delphi method, involved multiple stakeholders. It started by reviewing available literature to establish a preliminary list of lenses, which was used to trigger a brainstorming session with eleven UX experts who finally created and ranked the lenses. Final analytical lenses were validated by seventy (70) FOSS community stakeholders using questionnaires that passed content and face validity tests.

RQ-3 What factors (metrics) influence the UX maturity of desktop-based FOSS projects?

The findings from the exhausted literature review confirm the absence of UX maturity influencing factors exclusive to the FOSS community. Like many other software developing communities, the FOSS community still uses generically established UX maturity influencing factors that have not thoroughly been tested in the actual projects. As a result, these factors may be susceptible to errors and overlook the community-specific dynamics.

This study has conducted an empirical study reported in section four of Chapter Four to break the deadlock and respond to the third research question. The outcome of the completed empirical study is the thirty-six (36) UX maturity influencing factors exclusive to the FOSS community. The study adopted a methodology that first reviewed existing literature to establish a preliminary list of the influencing factors. Then, it interviewed 15 purposely selected UX assessment stakeholders actively engaged in discussion forums of the chosen projects to get deeper insights into the factors. Finally, it established experts' agreement and ranked the factors based on their importance using the fuzzy Delphi Method.

The identified UX maturity influencing factors were classified into six themes: Usability, adopted technologies, UX strategy, UX culture, UX processes, and UX outcome, and were adopted to develop a new methodology for assessing the UX maturity of desktop-based FOSS projects, explained in the fourth research question.

Nevertheless, from interview inputs of highly experienced stakeholders with the necessary skills regarding UX maturity and linked terms, the study found that UX maturity and linked concepts are known concepts in the FOSS community. The community's stakeholders often adopt and adapt explanations from Nielsen and Norman's group and further agree that embracing UCD approaches may improve organizational capacity to involve users through user research, content design, and assessment. However, the FOSS community might have unique UX maturity influencing factors different from any other software-developing community. Therefore, currently used UX assessment tools and included elements may be irrelevant to the FOSS community and are still open for debate.

RQ-4 What tool can optimize the assessment of the UX maturity of desktop-based FOSS projects? How is it developed?

The findings show that the FOSS community lacks an adequate model to optimize the design for UX in the development process of desktop-based FOSS projects. Available models do not adequately account for the FOSS community's UX maturity dynamics, which may not be relevant in drawing the actual picture of what is happening within the community. This study developed the FOSS-UXMM in section five of Chapter Four using the UX maturity influencing factors established from the completed empirical analysis in section four to answer RQ-4. The FOSS-UXMM was validated and verified in section six of chapter four and compared to other models in section seven of chapter four.

The FOSS-UXMM contains five levels of UX maturity, just like Paulk et al. (1993): Preliminary, Acknowledged, Accepted, Streamlined, and Institutionalized, whose assessment methodologies are self-administered questionnaires. The distinction between adopted levels of UX maturity is more apparent, and each level of UX maturity is measured using different questionnaires to simplify their explanation regarding operations. However, the first level does not have a specific questionnaire. The project is positioned at the first level when it does not qualify for conditions set by all other assessment tools.

Unlike other UX maturity models, the FOSS-UXMM has proposed a new performance scale, “Unknown,” to boost flexibility when NFS is unknown. It has also adopted rating scales, methodology, and thresholds from BOOTSTRAP.

RQ-5 How applicable is the FOSS-UXMM in determining the UX maturity of FOSS projects compared to other generic UXCMMs?

The findings from the exhausted literature review show that available UXCMMs may not be applicable in the FOSS community because they have never been tested in actual FOSS projects. The study performed two different sets of activities to respond to RQ-5. It first completed two case studies reported in section six of Chapter Four to test the questionnaires’ applicability, deepen understanding of assessment methodology, and determine the generalization of tools to every FOSS project. It also performed a comparative analysis between FOSS-UXMM and other UXCMMs. Generally, the completed case studies and comparative analysis findings show that the FOSS-UXMM fits well in FOSS projects and performs relatively better than other UXCMMs.

The FOSS-UXMM is easy to use with less or no complexity because it uses simple language that does not demand particular specialized knowledge. The results derived when using FOSS-UXMM are free from biases because they can involve multiple raters who must agree on their rating values based on Kendall's Coefficient of Concordance (W). However, the optimal number of raters for optimal assessment results is still being discussed.

The findings established when FOSS-UXMM is compared with other UXMMs may help the FOSS community prevent practitioners, developers, and other stakeholders from re-inventing the wheely by bringing together existing lines of work that may provide inputs for future improvements. Nine lenses were adopted to complete the comparative analysis: identity, inputs analysis, applicability in the FOSS projects, empirical evidence, integrating assessment tool(s), scope, concrete guidance, and operations. However, the study could not implement a complete side-to-side comparison between FOSS-UXMM and other UXCMMs due to the immense number of UXCMMS.

RQ-6 How is the design for UX in desktop-based FOSS projects optimized?

The design for UX may be optimized for a fair amount by implementing this study because its findings may help developers, contributors, owners, and other FOSS community stakeholders stop investing their time and other resources in building projects that will not be used due to bad UX.

Nevertheless, the study has offered several possibilities to the FOSS community that may help optimize the UX design, including understanding the preparation of desktop-based projects for UX maturity. Preparing desktop-based FOSS projects helps designers and developers tailor their approach, allocate resources effectively, foster collaboration, and advocate for UX design within the community. Consequently, it enables a targeted and incremental optimization of UX design practices, resulting in developing projects that better meet user needs and enhance user satisfaction.

The study has also assisted in understanding the UX maturity influencing factors exclusive to the FOSS community, which may also help significantly optimize the UX design. Understanding the UX maturity influencing factors allows designers to tailor their strategies, identify barriers, implement focused interventions, engage the community, foster collaboration and knowledge sharing, and drive continuous improvement. Generally, it ensures that the design efforts are relevant, effective, and aligned with the unique characteristics of the FOSS community.

Finally, the FOSS-UXMM equips the FOSS community with the ability to identify its strengths to capitalize on to develop products with desirable UX and weaknesses to address. It provides desktop-based FOSS projects status quo and suggests ways to overcome all identified challenges before projects can advance in their UX maturity. Generally, utilizing the FOSS-UXMM can make FOSS projects systematically evaluate, plan, and implement UX improvements. The FOSS-UXMM serves as a guide for optimizing design processes, allocating resources, setting goals, and fostering a culture of continuous improvement. Ultimately, it enhances the UX quality of desktop-based FOSS projects and their impact on users.

5.3 The Implication of the Study to Research and Practice

Study implications refer to the broader consequences or effects that the findings of a study may have on the field of research, practice, or policy. They highlight the practical or theoretical significance of the research and how the results can influence future decisions or actions. Implications can address various aspects, such as filling gaps in knowledge, challenging existing theories, suggesting new approaches, or informing practical applications. They provide insights into the potential impact and relevance of the study beyond the immediate findings.

The research's findings and discussions have imposed several implications on the UX assessment and the FOSS community in general that are beyond the study’s scope. The repercussions could help the FOSS community re-strategize and re-establish reliable techniques to enhance the UX of desktop-based FOSS projects. For example, they may help formulate new strategies, standards, policies, and theories. The thesis has divided the study’s implications into four categories: research, practice, policy, and computer science.

5.3.1 Implications to Research

As a significant contribution to the UX assessment, this study highlights potential dangers to the accuracy and legitimacy of the FOSS community's UX maturity assessment by showing the absence of proper processes, methods, techniques, and tools to optimize the design for UX in desktop-based FOSS projects. Drawing this picture can provide more insights into the need to advance practices, theories, or policies. This study highlights potential areas that require improvement and research efforts to promote better UX practices and broader adoption of FOSS projects.

First, the research community can investigate and advocate for adopting UCD methods within the FOSS community. These investigations may entail researching current design methods in FOSS projects and identifying places where user demands and comments may be addressed more effectively. They can further show how UX considerations might be incorporated into the development cycle of FOSS projects. Second, research can focus on developing and adapting UX evaluation methods tailored explicitly for FOSS projects. This focus can involve creating metrics and frameworks to assess the usability, accessibility, and overall user satisfaction of FOSS software. By providing effective evaluation methods, research can help FOSS developers identify and address UX issues more systematically. Third, research can investigate strategies to enhance collaboration and community engagement in the context of UX design. This investigation may include exploring ways to involve users, designers, developers, and other stakeholders in the design process, promoting effective communication channels, and fostering a UCD culture within FOSS communities.

Fourth, research can focus on improving the accessibility and quality of documentation and learning resources related to UX design in FOSS. This focus can involve developing guidelines, best practices, and tutorials tailored explicitly to FOSS developers to help them create more intuitive and usable software interfaces. Fifth, research can explore methods to improve the accessibility features and support within FOSS projects. This focus can involve investigating assistive technologies, developing guidelines for accessibility implementation, and raising awareness about the importance of inclusivity in FOSS design. Finally, research can explore models and strategies for sustaining and funding FOSS projects prioritizing UX design. This initiative can involve studying successful case studies, identifying funding mechanisms, and advocating for the recognition and support of UX designers within FOSS communities.

5.3.2 Implications to Practice

The FOSS community lacks experimentally tested elements and consistent methods that may explain its willingness to evaluate its UX maturity. Through identifying lenses for assessing readiness, the study has implicated practice by establishing design patterns, style guides, and standards to promote consistency and coherence. The results have also contributed knowledge to the practice by demonstrating a consistent procedure for identifying and ranking thirty-six essential UX maturity influencing factors exclusive to the FOSS community.

Nevertheless, the findings also enabled a clearer understanding of the UX Maturity assessment domain, weaknesses, and improvement areas for successfully implementing UX-related activities by adopting an iterative design approach that involves continuous testing, refinement, and improvement based on user feedback. They also provide clear, comprehensive, easily accessible documentation and help resources. Nevertheless, they recognize the importance of design expertise within FOSS projects and encourage the active involvement of UX designers, collaboration, and participation of community members. The completed case studies support that it was relatively easy for UX practitioners to pinpoint the FOSS projects' UX Maturity level using FOSS-UXMM and the UX maturity influencing factors identified. The developed model, FOSS-UXMM, also encourages collaboration and involvement of community members, including users, designers, developers, and other stakeholders. It creates channels for open communication, feedback, and contributions to foster a UCD culture within the community.

5.3.3 Implications to Policy

The findings may have ten policy implications in the FOSS community. First, Policy initiatives can emphasize the importance of UX design in FOSS projects and promote its recognition within the community. These initiatives can highlight the value of UCD design principles and their positive impact on UX, adoption, and user satisfaction. Second, Policies can support funding opportunities for FOSS projects prioritizing UX design, including grants, sponsorships, or research funding allocated explicitly for enhancing the UX of FOSS software. Financial support can enable projects to allocate resources to hire UX designers or collaborate with external design experts. Third, Policy initiatives can promote education and training opportunities for UX design within the FOSS community, which can involve establishing programs, workshops, or courses that provide FOSS developers and contributors with the necessary skills and knowledge to integrate UX design principles effectively. Fourth, Policy efforts can advocate for incorporating accessibility regulations and standards into FOSS projects, which can involve aligning FOSS software with existing accessibility guidelines or developing specific accessibility requirements for FOSS projects, ensuring that software is usable by individuals with disabilities. Fifth, Policies can address user privacy and data protection concerns within FOSS projects, which may involve defining guidelines or regulations that promote transparent data collection and usage practices, secure data handling, and user consent mechanisms. Policy initiatives can encourage FOSS projects to prioritize user privacy and implement appropriate safeguards.

Sixth, Policies can foster an open design culture within the FOSS community, encouraging collaboration and knowledge sharing. This initiative can involve advocating for open design methodologies, open licensing, and promoting platforms for sharing design assets, patterns, and best practices. Seventh, policies can encourage FOSS projects to implement user feedback mechanisms as an integral part of the development process, which can involve creating channels for users to provide feedback, report issues, or suggest improvements. They can also emphasize the importance of addressing user feedback and ensuring it informs the design iterations. Eighth, policy initiatives can support developing and disseminating UX guidelines and best practices tailored explicitly to FOSS projects. These resources can provide practical guidance on incorporating UX design principles into FOSS software and help developers improve the overall usability of their projects. Ninth, policies can facilitate collaboration between FOSS projects and UX design experts, which can involve establishing networks, platforms, or partnerships that connect FOSS communities with UX professionals who can provide guidance, feedback, and expertise in improving the UX of FOSS software. Finally, policies can encourage evaluating and recognizing FOSS projects based on their UX design efforts, including incorporating UX metrics into evaluation criteria, acknowledging FOSS projects prioritizing UX in award programs or creating certifications or badges highlighting the software's UCD approach.

5.3.4 Implications to Computer Science

UX design arose as a generation from HCI, which deals with designing computer systems for human use. Likewise, HCI is a crucial subfield of computer science, informing the visual design and workflow of computer systems we use daily. As a result, whatever implications the study has brought in UX design have similar consequences on HCI and computer science. Computer science is incomplete without improved knowledge of UX, including UX maturity assessment. However, there is no explicit agreement on whether UI/UX is a separate field or computer science component.

5.4 Research Contributions

Research contribution refers to an investigation's impact on a particular field or area of knowledge. It shows how much the study advances our understanding of the world, improves our ability to solve problems, or creates new opportunities for innovation. However, research contribution is complex and can mean different things to different people (Gendron, 2013). This study has offered subsidies in six ways: advancing knowledge in assessing the UX maturity of the FOSS community, improving the FOSS community UX maturity quality, advancing theoretical knowledge in FOSS UX maturity assessment, Simplifies the selection of ideal UXCMMs among students, researchers, and practitioners, changes politicians and decision-makers mindsets and creating new opportunities for innovations to improve UX maturity assessment of FOSS projects for optimizing the design for UX.

5.4.1 Advancing Knowledge in Assessing the FOSS Community’s UX Maturity

By finding many prerequisites and suggesting a model for UX maturity assessment, this study has increased stakeholders' understanding in evaluating the FOSS community's UX maturity. The study has discovered that the FOSS community did not have a suitable model for assessing its UX maturity. Nevertheless, despite embedded advantages, a consistent method to identify the preparation of FOSS projects to achieve higher levels of UX maturity was also missing. As a result, the FOSS community could not have a consistent way to understand whether its projects had indicators for achieving higher levels of UX maturity, which caused challenges in determining required interventions. By completing this study, the FOSS community can now analyze the preparation status using the ten analytical lenses developed using the Fuzzy Delphi method. Finally, the FOSS community has established its own UX maturity influencing factors to account for its dynamics, which was impossible before. It has used them to develop its model to assess FOSS projects' UX maturity.

5.4.2 Improves the FOSS Community's UX Maturity Assessment Quality.

This study and several other studies, such as Namayala et al.(2022) and Raza et al. (2012a), contend that existing UXCMMs were not ideal for assessing the UX maturity of FOSS projects. This conclusion has been reached after examining their applicability in the actual projects in the FOSS community. In addition, it is uncertain if present UXCMMs measure UX maturity correctly and contain relevant influencing factors that account for the dynamics of the FOSS community. Nevertheless, the difficulty and expense of embedded assessment procedures limit the adoption of contemporary UXCMMs and other CMMs, such as CMMI, to many desktop-based projects.

This study created a new model from the thirty-six UX maturity influencing variables developed empirically using the fuzzy Delphi method as an endeavor to address identified gaps, simplify, and improve the quality of UX maturity evaluation in the FOSS community. It has created a new guideline that can be used to guide future developments of SPCMMs. However, developed FOSS-UXMM and guidelines to develop SPCMMs are open for improvement.

5.4.3 Advance the Theoretical knowledge in FOSS UX Maturity Assessment

This study has offered new insights, perspectives, and prerequisites for assessing the FOSS community's UX maturity assessment. The study has shown that although assessing UX maturity using existing and developed UXCMMs is essential, if the FOSS community is not prepared to achieve higher levels of UX maturity, these initiatives will not make a difference. It has also contended that the UX assessment community cannot continue implicating UX maturity to cost only. The community must also examine how UX maturity relates to its impacts on the success of projects or organizations.

In addition, the study has encouraged the UX assessment community to use alternative techniques instead of UXCMMs, such as UX maturity assessment frameworks, UX institutionalization strategies, and other techniques not reliant on context and application areas. It has also advanced knowledge in examining the applicability of contemporary UXCMMs in FOSS by advising analytical lenses, established analytical lenses to determine organizational preparation to achieve higher levels of UX maturity, identified FOSS community-specific UX maturity influencing factors, and performed a comparative analysis of contemporary UXCMMs. The study has also provided a new guideline to make the process for creating SPCMMs consistent. Generally, the knowledge from the study's findings may help refine or extend existing theories regarding UX maturity assessment in the FOSS community.

5.4.4 Simplifies the Selection of Relevant UXCMM

The UX assessment community, particularly in the FOSS community, faces a dilemma in identifying UXCMMs appropriate for assessing the UX maturity of its projects. However, by doing a comparative analysis, this study has provided an optimal method and simplified the selection of UXCMM that is appropriate for a specific application.

5.4.5 Changes Mindsets of Politicians and Other Decision-makers

Highlighting potential benefits developing countries may enjoy by optimizing the design for UX in the development process of desktop-based FOSS projects and enhancing its adoptions, the study may influence politicians and other decision-makers to shift their mindsets toward supporting FOSS adoption and implementation initiatives. Once the mindset of Tanzanian politicians and other decision-makers are changed, policies and other FOSS institutionalization indicators will be favored.

5.4.6 Offers New Opportunities for Innovations to Improve UX Maturity Assessment

In addition to requesting future researchers to undertake the additional examination of the relevancy of the contemporary UXCMMs in assessing the UX maturity of the FOSS community, improve analytical lenses for examining the preparation of the community's projects to achieve higher levels of UX maturity, conduct a supplementary examination of FOSS exclusive UX maturity influencing factors, improve the developed FOSS-UXMM, and perform side to side comparison of present UXCMMs, This study has proposed several other areas for future researchers as stipulated in the implications for research section to innovate and work on to improve the assessment of UX maturity in the FOSS community.

5.5 Conclusion

Based on the FOSS community's ongoing advancement, which requires FOSS projects with desirable UX and meeting users' expectations, the community's old culture must also change. For example, it must stop thinking that its projects are only for software engineers and are unaffected by UX-related issues. The FOSS community's user base has escalated beyond technically equipped, and its projects are currently used in critically based missions that demand no operation mistakes. This conclusion confirms the results from Dawood et al. (2019), Mtebe (2019), and Cheng and Guo (2018). Nevertheless, the FOSS community must also ensure all prerequisites enabling FOSS projects to achieve higher levels of UX maturity are in place and measurable.

Although desktop-based FOSS projects accommodate relatively low TCO and aim to fill the digital divide gap resulting from inadequate resources to procure proprietary software, end users do not widely adopt them. Many FOSS projects fall short of the expectations of practitioners and other stakeholders inside and outside the FOSS community, who predict their adoption to be higher. Many desktop computer users, particularly in developing countries, still use highly pirated proprietary software, warranting the failure of other ongoing interventions promoting adopting desktop-based FOSS projects, including TAM and UTAUT.

Contemporary issues in the literature confirm that the adoption misfortune of FOSS projects is mainly a result of poor UX demanding necessary initiatives to be taken by the FOSS community to enhance the UX of its projects. However, the FOSS community's researchers generally give UX research relatively little attention. As a result, limited tools, methods, and research to address UX-related problems, including UX evaluations, exist. UX is an organizational endeavor that cannot be improved by having few individuals knowledgeable on UX/HCI. Best practices encourage using UXCMMs to improve the UX of products. However, none has justified its suitability in the FOSS community (Namayala et al., 2022).

The main contribution of this study is the model to evaluate the UX maturity of the FOSS community (FOSS-UXMM). The model comprises assessment questionnaires, a performance scale, and a rating strategy and has been created to address organizational UX problems in the FOSS community. Before developing the FOSS-UXMM, the study implemented measures that ensured the availability of prerequisites for UX maturities, such as testing the practical applicability of contemporary UXCMMs in the FOSS community, understanding how prepared the FOSS projects are for UX maturity and establishing the factors affecting UX maturity specific to the FOSS community. Finally, the FOSS-UXMM was validated in the actual FOSS projects from the completed case studies.

Generally, this study has derived fivefold conclusions based on the collected findings and discussions. First, the FOSS community still lacks harmonized methods to assess desktop-based preparation for UX maturity. As a result, more analytical lenses can be investigated, tested, and validated to help draw a holistic picture of the preparedness of FOSS projects for UX maturity. Second, although thirty-six factors influencing UX maturity in the FOSS community have been identified using the fuzzy Delphi Method, the FOSS community still needs further exploration. The factors established and adopted are based on experts' opinions that might not reflect the overall picture for the community. As a result, the FOSS community must implement supplementary examinations that may involve other stakeholders' views to draw a holistic picture.

Third, although the validated FOSS-UXMM exhibits a strong fit with the data gathered and sufficient power to explain the relationship between influencing factors and the UX maturity of the FOSS projects, completed case studies are not enough for generalizing the applicability of the FOSS-UXMM. Thus, the FOSS community must perform more case studies that may involve more projects and stakeholders. Fourth, although a single rater can rate the UX maturity of a given FOSS project using the proposed model, multiple raters who agree in their rating help reduce biases in the derived results. The ideal number of raters for reliable assessment results has not been established. Thus, more interventions are required to investigate the correct number of involved raters for optimal results after adopting FOSS-UXMM.

Fifth, although from the completed comparative analysis between FOSS-UXMM and other UXCMMs, the study can spot better performance of FOSS-UXMM. The investigation is poorly conducted and does not explain the fundamental differences between FOSS-UXMM and other UXCMMs. As a result, more efforts are required to implement a side-to-side comparative analysis that may exhaust many lenses or features.

5.6 Recommendations

Generally, the completed study has focused on four research objectives and six research questions. These components may not have covered other significant goals and roles that might significantly improve the design for UX in the development cycle of desktop-based FOSS projects, particularly in developing countries where FOSS and UX are not institutionalized. Unlike European countries and other developed countries that have laid the foundations for FOSS and UX practices, several developing countries, including Tanzania, may not eventually benefit from this study, necessitating different measures. These initiatives may involve acknowledging UX and FOSS in strategic goals and ICT policies, raising FOSS and UX awareness, and discontinuing supporting proprietary software adoption in favor of readily available FOSS projects.

Several organizations in developing countries are informally adopting FOSS projects and UX practices. For example, in Tanzania, the University of Dodoma uses several FOSS projects for teaching and other tasks, including compilers, database servers, web servers, and student records management systems, amounting to approximately 30% of the adopted software. However, reviews of the primary documents, including ICT policies of the University of Dodoma (UDOM), confirm limited information regarding FOSS and UX. The situation is the same for other higher education institutions (HEIs), including the University of Dar es Salaam (UDSM), the Mbeya University of Science Technology (MUST), and the College of Business Education (CBE).

As a result, the study recommends sixfold areas discussed hereunder for future research to enhance the optimization of the design for UX in the development process of desktop-based FOSS projects. First, the proposed FOSS-UXMM is not fixed or unchanged. Therefore, it is open to continuous development and improvement. For example, the FOSS-UXMM and the developed scale could be a basis for forming a starting point for developing other instruments for evaluating the UX maturity of the FOSS community. As a result, more studies are required to extend or modify the FOSS-UXMM by adding additional dimensions or external variables.

Second, the completed self-assessment case studies involved the developer testing FOSS-UXMM questionnaires and communicating the findings. As a result, derived conclusions on the ease of use and the perfect fit of the FOSS-UXMM may not be confirmed unless independent reviewers are involved. The study, therefore, recommends enhancing the assessment methodology by introducing onsite assessments performed by independent assessors.

Third, rather than creating a more generic model that can assess the UX maturity of any desktop-based FOSS project, we may modify current UXCMMs, including FOSS-UXMM, to specific application domains such as communications, medicine, and the automobile industry. Exploratory research in this area will aid in developing more domain-oriented models.

Fourth, Although the FOSS-UXMM is easily applicable in the FOSS community, it still lacks a definition of how improvement plans will be produced and implemented after evaluation. A framework for moving up the maturity ladder from one maturity level to the next is also lacking. As a result, the study encourages future research into these two topics.

Fifth, although the study identified thirty-six UX maturity influencing factors to the FOSS community and ten lenses that aid in testing the FOSS community's readiness to assess its UX maturity, they are not rigorously tested to generalize their applicability. Furthermore, they may not be sufficient to explain the overall picture. As a result, the study suggests that future studies conduct additional testing on the components and lenses and, if possible, develop others.

Finally, the proposed guideline for developing SPCMMs presented in a published paper by Namayala et al. (2022) is not fixed, unchanged, and open to continuous development and improvement. For example, the validated procedures could be a basis for forming a starting point when developing standardized guidelines for developing SPCMMs. Therefore, more studies are required to extend or modify the presented approach by adding additional dimensions, steps, or variables.

REFERENCES

Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003a). Agile software development methods: A comparative review. *The International Conference on Software Engineering, May 3-5, 2003*. Portland,Oregon, USA.: IEEE.

Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003b). New directions on agile methods: A comparative analysis. *Proceedings - International Conference on Software Engineering*, 244–254. https://doi.org/10.1109/icse.2003.1201204

Abrahão, S., Juristo, N., Law, E. L. C., & Stage, J. (2010). Interplay between usability and software development. *Journal of Systems and Software*, *83*(11), 2015–2018. https://doi.org/10.1016/j.jss.2010.05.080

Abramova, V., Pires, F., & Bernardino, J. (2016). Open Source and Proprietary Project Management Tools for SMEs. *Journal of Information Systems Engineering & Management*, *1*(3), 177–186. https://doi.org/10.20897/lectito.201633

Abzalov, M. (2016). Exploratory Data Analysis. In *Modern Approaches in Solid Earth Sciences* (Vol. 12, pp. 207–219). https://doi.org/10.1007/978-3-319-39264-6_15

Adams, C. (2009). Eight is (not) enough. Retrieved June 14, 2020, from Printing Impressions website: https://articles.uie.com/eight_is_not_enough/

Adem, A., Mbugua, D. K., Gathuka, P., Mwogoi, T., & Sciences, H. (2021). *Response Rate, Incentives and Timing of Online Surveys: A study of Agriculture Researchers in Kenya *. 20*(1), 82–93.

Adewumi, A., Misra, S., & Omoregbe, N. (2013). A Review of Models for Evaluating Quality in Open Source Software. *IERI Procedia*, *4*, 88–92. https://doi.org/10.1016/j.ieri.2013.11.014

Adewumi, A., Misra, S., Omoregbe, N., Crawford, B., & Soto, R. (2016). A systematic literature review of open source software quality assessment models. *SpringerPlus, 5*(1), 1936. https://doi.org/10.1186/s40064-016-3612-4

Adhitya, C., Andreswari, R., & Alam, P. F. (2021). Analysis and Design of UI and UX Web-Based Application in Maiproyek Startup Using User Centered Design Method in Information System Program of Telkom University. *IOP Conference Series: Materials Science and Engineering*, *1077*(1), 012039. https://doi.org/10.1088/1757-899X/1077/1/012039

Agu, M., & Bakpo, F. (2011). Lessons and challenges from software quality assessment: The case of space systems software. *Global Journal of Pure and Applied Sciences*, *17*(1), 93-98–98. https://doi.org/10.4314/GJPAS.V17I1

Ahmad, W. N. W., & Ali, N. M. (2013). Engendering Trust through Emotion in Designing Persuasive Application. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*: *Vol. 8237 LNCS* (pp. 707–717). https://doi.org/10.1007/978-3-319-02958-0_64

Ahmady, G. A., Mehrpour, M., & Nikooravesh, A. (2016). Organizational Structure. *Procedia - Social and Behavioral Sciences*, *230*, 455–462. https://doi.org/10.1016/j.sbspro.2016.09.057

Al-Azzawi, A. (2014). *Experience with Technology* (1st Ed). London: Springer London. https://doi.org/10.1007/978-1-4471-5397-9

Al-nuiam, H., & Al-Harigy, L. (2015). User Interface Context of Use Guidelines for Mobile Apps. *International Journal of Human Computer Interaction(IJHCI)*, (3), 65–80.

Alba, J. W., & Hasher, L. (1983). Is memory schematic? *Psychological Bulletin*, *93*(2), 203–231. https://doi.org/10.1037/0033-2909.93.2.203

Alben, L. (1996). Quality of experience: defining the criteria for effective interaction design. *Interactions*, *3*(3), 11–15. Retrieved from http://portal.acm.org/citation.cfm?doid=332040.332433

Alfonzo, O., Dom, K., Rivas, L., P, M., Mendoza, L., & Ortega, M. (2008). Quality Measurement Model for Analysis and Design Tools Based on FLOSS. *19th Australian Conference on Software Engineering (Aswec 2008)*, 258–268. IEEE. https://doi.org/10.1109/ASWEC.2008.4483214

Alghawli, A. S. A., Al-khulaidi, A. A., Nasser, A. A., AL-Khulaidi, N. A., & Abass, F. A. (2022). Application of the Fuzzy Delphi Method to Identify and Prioritize the Social-Health Family Disintegration Indicators in Yemen. *International Journal of Advanced Computer Science and Applications*, *13*(5), 680–691. https://doi.org/10.14569/IJACSA.2022.0130579

Alharbi, M. G., & Khalifa, H. A. E. W. (2021). Enhanced Fuzzy Delphi Method in Forecasting and Decision-Making. *Advances in Fuzzy Systems*, *2021*, 1–6. https://doi.org/10.1155/2021/2459573

Ali, S., Hongqi, L., Khan, S. U., Zhongguo, Y., & Liping, Z. (2017). Success Factors for Software Outsourcing Partnership Management: An Exploratory Study Using Systematic Literature Review. *IEEE Access*, *5*, 23589–23612. https://doi.org/10.1109/ACCESS.2017.2764946

Ali, S., Li, H., Khan, S. U., Zhao, Y., & Li, L. (2018). Fuzzy multi attribute assessment model for software outsourcing partnership formation. *IEEE Access*, *6*, 55431–55461. https://doi.org/10.1109/ACCESS.2018.2871710

Allam, A. H., Che Hussin, A. R., & Dahlan, H. M. (2013). User Experience: Challenges and Opportunities. *Journal of information systems research and innovation*, *3*(1), 28–36.

AlMheiri, N. H., Rajan, A., & Akre, V. (2018). Framework for Open Source Software implementation in the Government Sector of Dubai. *2018 Fifth HCT Information Technology Trends (ITT)*, 71–76. IEEE. https://doi.org/10.1109/CTIT.2018.8649506

Alves, R., Valente, P., & Nunes, N. J. (2014). The state of user experience evaluation practice. *Proceedings of the NordiCHI 2014: The 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, 93–102. New York, New York, USA: ACM Press. https://doi.org/10.1145/2639189.2641208

Amega-Selorm, C., & Awotwi, J. (2010). Free and open source software (FOSS). https://doi.org/10.1145/1930321.1930342

Amrit, C., & Van Hillegersberg, J. (2010). Exploring the impact of socio-technical core-periphery structures in open source software development. *Journal of Information Technology*, *25*(2), 216–229. https://doi.org/10.1057/jit.2010.7

Amrollahi, A., Khansari, M., & Manian, A. (2014). Success of Open Source in Developing Countries. *International Journal of Open Source Software and Processes, 5*(1), 50–65. https://doi.org/10.4018/ijossp.2014010103

Anchahua, M. C., Garnique, L. V., & Tarazona, J. A. (2018). User Experience Maturity Model for Ecommerce Websites. *2018 Congreso Internacional de Innovación y Tendencias En Ingeniería (CONIITI)*, 1–6. IEEE. https://doi.org/10.1109/CONIITI.2018.8587080

Andrade, C. (2020). The Limitations of Online Surveys. *Indian Journal of Psychological Medicine*, *42*(6), 575–576. https://doi.org/10.1177/ 02537 17620957496

Andrade, C. (2021). The Inconvenient Truth About Convenience and Purposive Samples. *Indian Journal of Psychological Medicine*, *43*(1), 86–88. https://doi.org/10.1177/0253717620977000

Andreasen, M. S., Nielsen, H. V., Schrøder, S. O., & Stage, J. (2006). Usability in Open Source Software Development: Opinions and Practice. In *Information Technology And Control* (Vol. 35). https://doi.org/10.5755/j01.itc.35.3.11776

Androutsellis-Theotokis, S., Spinellis, D., Kechagia, M., & Gousios, G. (2010). Open Source Software: A Survey from 10,000 Feet. *Foundations and Trends® in Technology, Information and Operations Management*, *4*(3–4), 187–347. https://doi.org/10.1561/0200000026

Anitha, P., & Prabhu, B. (2012). Integrating requirements engineering and user experience design in Product life cycle Management. *2012 First International Workshop on Usability and Accessibility Focused Requirements Engineering (UsARE)*, 12–17. IEEE. https://doi.org/10.1109/UsARE.2012.6226784

Anthony, B., Boudnik, K., Adams, C., Shao, B., Lee, C., & Sasaki, K. (2016). User Experience. https://doi.org/10.1002/9781119281320.ch4

Apraiz Iriarte, A., & Ganix, L. E. (2020). Questionnaires for the evaluation of User Experience and Usability in industrial environments: A Bibliographic Review. *DYNA NEW TECHNOLOGIES*, *7*(1), [21 p.]-[21 p.]. https://doi.org/10.6036/NT9845

Araz, C. (2018). Why you should ditch your UX definition, and use this one instead. Retrieved December 31, 2020, from UX Collective website: https://uxdesign.cc/we-have-lost-track-of-what-ux-actually-means-8d55259dacb0

Ardito, C., Buono, P., Caivano, D., Costabile, M. F., & Lanzilotti, R. (2014). Investigating and promoting UX practice in industry: An experimental study. *International Journal of Human-Computer Studies*, *72*(6), 542–551. https://doi.org/10.1016/j.ijhcs.2013.10.004

Ardito, C., Buono, P., Caivano, D., Costabile, M. F., Lanzilotti, R., & Dittrich, Y. (2014). Human-Centered Design in Industry: Lessons from the Trenches. *Computer*, *47*(12), 86–89. https://doi.org/10.1109/MC.2014.355

Arhippainen, L. (2009). Studying user experience: issues and problems of mobile services:- Case ADAMOS: User experience (im)possible to catch? *Faculty of Science of the University of Oulu*, *42*(4), 1–9.

Arhippainen, L., & Tähti, M. (2003). Empirical evaluation of user experience in two adaptive mobile application prototypes. *Proceedings of the 2nd International Conference …*, 27–34. Retrieved from http://www.ep.liu.se/ecp/011/ 007/ecp011007.pdf

Aubakirov, M., & Nikulchev, E. (2016). Development of System Architecture for E-Government Cloud Platforms. *International Journal of Advanced Computer Science and Applications*, *7*(2). https://doi.org/10.14569/IJACSA.2016.070235

Aversano, L., & Tortorella, M. (2013). Quality evaluation of floss projects: Application to ERP systems. *Information and Software Technology*, *55*(7), 1260–1276. https://doi.org/10.1016/j.infsof.2013.01.007

Avison, D. D. E., & Fitzgerald, G. (1995). Information systems develpment: methodologies, techniques and tools. In *Information and Software Technology* (2nd ed., Vol. 31). Berkshire, UK: Paul & Company Publishers Consortium, Inc .\.

Axelsson, R., March, J. G., & Olsen, J. P. (1977). Ambiguity and Choice in Organizations. *The Scandinavian Journal of Economics*, *79*(1), 138. https://doi.org/10.2307/3439574

Ayala, C., Hauge, Ø., Conradi, R., Franch, X., & Li, J. (2011). Selection of third party software in Off-The-Shelf-based software development—An interview study with industrial practitioners. *Journal of Systems and Software*, *84*(4), 620–637. https://doi.org/10.1016/j.jss.2010.10.019

Bach, P. M., DeLine, R., & Carroll, J. M. (2009). Designers wanted: Participation and the User Experience in Open Source Software Development. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 985–994. New York, NY, USA: ACM. https://doi.org/10.1145/1518701.1518852

Bach, P. M., & Twidale, M. (2010). Social participation in open source. *Interactions*, *17*(3), 70–74. https://doi.org/10.1145/1744161.1744177

Badampudi, D., Wohlin, C., & Petersen, K. (2016). Software component decision-making: In-house, OSS, COTS or outsourcing - A systematic literature review. *Journal of Systems and Software*, *121*, 105–124. https://doi.org/10.1016/ j.jss.2016.07.027

Bahamdain, S. S. (2015). Open source software (OSS) quality assurance: A survey paper. *Procedia Computer Science*, *56*(1), 459–464. https://doi.org/10.1016/j.procs.2015.07.236

Baldwin, C. Y., & Clark, K. B. (2006). The architecture of participation: Does code architecture mitigate free riding in the open source development model? *Management Science*, *52*(7), 1116–1127. https://doi.org/10.1287/ mnsc. 1060.0546

Banerjee, A., & Chaudhury, S. (2010). Statistics without tears: Populations and samples. *Industrial Psychiatry Journal*, *19*(1), 60. https://doi.org/10.4103/0972-6748.77642

Barcellini, F., Détienne, F., & Burkhardt, J. M. (2014). A situated approach of roles and participation in open source software communities. *Human-Computer Interaction*, *29*(3), 205–255. https://doi.org/10.1080/07370024.2013.812409

Barchard, K. A., Freeman, A. J., Ochoa, E., & Stephens, A. K. (2020). Comparing the accuracy and speed of four data-checking methods. *Behavior Research Methods*, *52*(1), 97–115. https://doi.org/10.3758/s13428-019-01207-3

Barchard, K. A., & Pace, L. A. (2008). Meeting the challenge of high quality data entry: a free double-entry system. *International Journal of Services and Standards*, *4*(4), 359. https://doi.org/10.1504/IJSS.2008.020053

Barchard, K. A., & Pace, L. A. (2011). Preventing human error: The impact of data entry methods on data accuracy and statistical results. *Computers in Human Behavior*, *27*(5), 1834–1839. https://doi.org/10.1016/j.chb.2011.04.004

Barchard, K. A., & Verenikina, Y. (2013). Improving data accuracy: Selecting the best data checking technique. *Computers in Human Behavior*, *29*(5), 1917–1922. https://doi.org/10.1016/j.chb.2013.02.021

Bargas-Avila, J. A., & Hornbæk, K. (2011). *Bargas-Avila (2011) - Old Wine in New Bottles or Novel Challenge_A Critical Analysis of Empirical Studies of UX.pdf*. 2689–2698.

Bartlett, F. C., & Burt, C. (1933). Remembering: a study in experimental and social psychology. *British Journal of Educational Psychology*, *3*(2), 187–192. https://doi.org/10.1111/j.2044-8279.1933.tb02913.x

Baškarada, S. (2009). *Information Quality Management Capability Maturity Model*. Wiesbaden: Vieweg+Teubner. https://doi.org/10.1007/978-3-8348-9634-6

Baskerville, R., Baiyere, A., Gergor, S., Hevner, A., & Rossi, M. (2018). Design Science Research Contributions: Finding a Balance between Artifact and Theory. *Journal of the Association for Information Systems*, *19*(5), 358–376. https://doi.org/10.17705/1jais.00495

Batarseh, F. A., Mohod, R., Kumar, A., & Bui, J. (2020). The application of artificial intelligence in software engineering: A review challenging conventional wisdom. In *Data Democracy: At the Nexus of Artificial Intelligence, Software Development, and Knowledge Engineering* (pp. 179–232). https://doi.org/10.1016/B978- 0-12-818366-3.00010-1

Baum, D., Huff, L., Clingan, J., & Bostrom, P. (2013). *The Department of Defense (DoD) and Open Source Software, Oracle White Paper*.

Becker, J., Knackstedt, R., Lis, L., & Stein, A. (2010). Towards a maturity model for research portals. *18th European Conference on Information Systems, ECIS 2010*.

Becker, J., Knackstedt, R., & Pöppelbuß, J. (2009). Developing Maturity Models for IT Management. *Business & Information Systems Engineering*, *1*(3), 213–222. https://doi.org/10.1007/s12599-009-0044-5

Bedford, A. (2016). Trustworthiness in Web Design: 4 Credibility Factors. Retrieved from Nielsen Norman Group (NN/g) website: https://www.nngroup.com/articles/trustworthy-design

Begel, A., Herbsleb, J. D., & Storey, M. A. (2012). The future of collaborative software development. *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, 17–18. Seattle, WA, USA: ACM. https://doi.org/10.1145/2141512.2141522

Bell, K. (2014). longitudinal effect. Retrieved July 6, 2022, from Open Education Sociology Dictionary website: https://sociologydictionary.org/longitudinal-effect/

Benítez, I., Van de Vijver, F., & Padilla, J. L. (2022). A Mixed Methods Approach to the Analysis of Bias in Cross-cultural Studies. *Sociological Methods & Research*, *51*(1), 237–270. https://doi.org/10.1177/0049124119852390

Benson, C., Muller-Prove, M., & Mzourek, J. (2004). Professional usability in open source projects. *Extended Abstracts of the 2004 Conference on Human Factors and Computing Systems - CHI '04*, 1083. New York, New York, USA: ACM Press. https://doi.org/10.1145/985921.985991

Benson, T. (2016). Open Source Paradigm: A Synopsis of The Cathedral and the Bazaar for Health and Social Care. *Journal of Innovation in Health Informatics*, *23*(2), 488. https://doi.org/10.14236/jhi.v23i2.866

Berendes, S., Hilpert, S., Günther, S., Muschner, C., Candas, S., Hainsch, K., … Söthe, M. (2022). Evaluating the usability of open source frameworks in energy system modelling. *Renewable and Sustainable Energy Reviews*, *159*, 112174. https://doi.org/10.1016/j.rser.2022.112174

Berni, A., & Borgianni, Y. (2021). From the definition of user experience to a framework to classify its applications in design. *Proceedings of the Design Society*, *1*(August), 1627–1636. https://doi.org/10.1017/pds.2021.424

Bhalerao, S., & Kadam, P. (2010). Sample size calculation. *International Journal of Ayurveda Research*, *1*(1), 55. https://doi.org/10.4103/0974-7788.59946

Bhandari, P. (2022). Questionnaire Design | Methods, Question Types & Examples A. Retrieved June 9, 2022, from https://www.scribbr. com/methodology/ questionnaire/%0AQuestionnaire

Bian, P., & Zhang, N. (2012). Gestalt Psychology Theory Based Design of User's Experience on Online Trade Websites. In *Advances in Intelligent and Soft Computing*: *Vol. 140 AISC* (pp. 327–332). https://doi.org/10.1007/978-3-642-27945-4_52

Biberoglu, E., & Haddad, H. (2002). A survey of industrial experiences with CMM and the teaching of CMM practices. *Journal of Computing Sciences in Colleges*, *18*(2), 143–152.

Bilousova, L. I., Gryzun, L. E., & Zhytienova, N. V. (2021). Fundamentals of UI/UX design as a component of the pre-service specialist's curriculum. *SHS Web of Conferences*, *104*, 02015. https://doi.org/10.1051/shsconf/202110402015

Bodjanova, S. (2006). Median alpha-levels of a fuzzy number. *Fuzzy Sets and Systems*, *157*(7), 879–891. https://doi.org/10.1016/j.fss.2005.10.015

Bohus, D., & Horvitz, E. (2009). Dialog in the open world. *Proceedings of the 2009 International Conference on Multimodal Interfaces - ICMI-MLMI '09*, 31. New York, New York, USA: ACM Press. https://doi.org/10.1145/1647314.1647323

Boichuck, O. (2020). The Nine Principles of UX Design Psychology: Can You Predict the Behavior of Your Users? | UX Magazine. *UX Magazine*. Retrieved from https://uxmag.com/articles/the-nine-principles-of-ux-design-psychology-can-you-predict-the-behavior-of-your-users

Boivie, I., Åborg, C., Persson, J., & Löfberg, M. (2003). Why usability gets lost or usability in in-house software development. *Interacting with Computers*, *15*(4), 623–639. https://doi.org/10.1016/S0953-5438(03)00055-9

Bonacchi, S., & Czerniak, S. (2015). Introduction: Gestalt as structure principles in science, art and language. *Dialogue and Universalism*, *25*(4), 7–9. https://doi.org/10.5840/du201525480

Bongard-Blanchy, K., Bouchard, C., Bongard-Blanchy, K., Bouchard, C., Experience, U., Design, P., & Bongard-Blanchy, K. (2015). Dimensions of User Experience - from the Product Design Perspective. *Journal d'Interaction Personne-Système*, *3(1), 1-13*. https://doi.org/10.46298/j ips.1284

Booker, J. M., & McNamara, L. A. (2004). Solving black box computation problems using expert knowledge theory and methods. *Reliability Engineering & System Safety*, *85*(1–3), 331–340. https://doi.org/10.1016/j.ress.2004.03.021

Boote, D. N., & Beile, P. (2005). Scholars Before Researchers: On the Centrality of the Dissertation Literature Review in Research Preparation. *Educational Researcher*, *34*(6), 3–15. https://doi.org/10.3102/0013189X034006003

Bornoe, N., & Stage, J. (2014). Usability engineering in the wild: How do practitioners integrate usability engineering in software development? *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8742*, 199–216. https://doi.org/10.1007/978-3-662-44811-3_12

Bosley, J. J. (2013). Creating a Short Usability Metric for User Experience (UMUX) Scale. *Interacting with Computers*, *25*(4), 317–319. https://doi.org/10.1093/iwc/iwt007

Botha, A., Herselman, M., & Van Greunen, D. (2010). Mobile user experience in a mlearning environment. *ACM International Conference Proceeding Series*, 29–38. New York, New York, USA: ACM Press. https://doi.org/10.1145/1899503.1899507

Bouras, C., Kokkinos, V., & Tseliou, G. (2013). Methodology for Public Administrators for selecting between open source and proprietary software. *Telematics and Informatics*, *30*(2), 100–110. https://doi.org/10.1016/j.tele.2012.03.001

Boyce, C., & Neale, P. (2006). Conducting In-Depth Interviews: A Guide for Designing and Conducting In-Depth Interviews for Evaluation Input. *Pathfinder International*, *2*(May), 1–16.

Bretschneider, P. J., Cirilli, S., Jones, T., Lynch, S., & Wilson, N. A. (2017). *Document Review as a Qualitative Research Data Collection Method for Teacher Research*. 1 Oliver's Yard, 55 City Road, London EC1Y 1SP United Kingdom: SAGE Publications Ltd. https://doi.org/10.4135/9781473957435

Brooke, J. (1996). SUS: A "Quick and Dirty" Usability Scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & A. L. McClelland (Eds.), *Usability Evaluation In Industry* (Vol. 189, pp. 207–212). London: CRC Press. https://doi.org/10.1201/9781498710411-35

Brutus, S., Aguinis, H., & Wassmer, U. (2013). Self-Reported Limitations and Future Directions in Scholarly Reports. *Journal of Management*, *39*(1), 48–75. https://doi.org/10.1177/0149206312455245

Bruun, A., Larusdottir, M. K., Nielsen, L., Nielsen, P. A., & Persson, J. S. (2018). The role of UX professionals in agile development. *Proceedings of the 10th Nordic Conference on Human-Computer Interaction*, 352–363. New York, NY, USA: ACM. https://doi.org/10.1145/3240167.3240213

Bryan, P. (2012). 3 Keys to Aligning UX with Business Strategy :: UXmatters. Retrieved December 31, 2020, from http://www.uxmatters.com/mt/archives/2012/09/3-keys-to-aligning-ux-with-business-strategy.php

Budgen, D., & Brereton, P. (2006). Performing systematic literature reviews in software engineering. *Proceedings - International Conference on Software Engineering*, *2006*, 1051–1052. New York, New York, USA: ACM Press. https://doi.org/10.1145/1134285.1134500

Bufe, A. (2021). Gestalt Principles: The Complete Overview. *Bluespace*. Retrieved from https://uxcam.com/blog/gestalt-principles/

Buffart, H., & Jacobs, H. (2021). A Gestalt Theory Approach to Structure in Language. *Frontiers in Psychology*, *12*. https://doi.org/10.3389/ fpsyg. 2021.649384

Buis, Elaine, E. G., Ashby, Simone, S. R., & Kouwenberg, Kristel, K. P. A. (2023). Increasing the UX maturity level of clients: A study of best practices in an agile environment. *Information and Software Technology*, *154*, 107086. https://doi.org/10.1016/j.infsof.2022.107086

Buley, L. (2019). The New Design Frontier The widest-ranging report to date examining design's impact on business. *InVision Research*. Retrieved from https://www.invisionapp.com/design-better/design-maturity-model/

Burmester, M., Mast, M., Jäger, K., & Homans, H. (2010). Valence method for formative evaluation of user experience. *Proceedings of the 8th ACM Conference on Designing Interactive Systems - DIS '10*, 364. New York, New York, USA: ACM Press. https://doi.org/10.1145/1858171.1858239

Butler, S., Gamalielsson, J., Lundell, B., Brax, C., Mattsson, A., Gustavsson, T., … Lönroth, E. (2022). Considerations and challenges for the adoption of open source components in software-intensive businesses. *Journal of Systems and Software*, *186*, 111152. https://doi.org/10.1016/j.jss.2021.111152

Buxton, B. (2007). Sketching User Experiences. In *Sketching User Experiences: Getting the Design Right and the Right Design*. Elsevier. https://doi.org/10.1016/B978-0-12-374037-3.X5043-3

Bwalya, T., Akakandelwa, A., & Dobreva-McPherson, M. (2019). Adoption and Use of Free and Open Source Software (FOSS) Globally : An Overview and Analysis of Selected Countries. *Zambia Journal of Library & Information Science (ZAJLIS)*, *3*(1 & 2), 48–66.

Bygstad, B., Ghinea, G., & Brevik, E. (2008). Software development methods and usability: Perspectives from a survey in the software industry in Norway. *Interacting with Computers*, *20*(3), 375–385. https://doi.org/10.1016/j.intcom.2007.12.001

Cajander, Å., Gulliksen, J., & Boivie, I. (2006). Management perspectives on usability in a public authority. *Proceedings of the 4th Nordic Conference on Human-Computer Interaction: Changing Roles*, *189*, 38–47. New York, NY, USA: ACM. https://doi.org/10.1145/1182475.1182480

Campbell, D. T., & Fiske, D. W. D. W. (1959). Convergent and discriminant validation by the multitrait-multimethod matrix. *Psychological Bulletin*, *56*(2), 81–105. https://doi.org/10.1037/h0046016

Capan, P. (2020). 6 Gestalt Principles in UX Design. Retrieved from UX Hints website: https://uxhints.com/wp -content/uploads/gestalt _principles_ uxhints.com_.pdf

Carbon, R., Ciolkowski, M., Heidrich, J., John, I., & Muthig, D. (2007). Evaluating open source software through prototyping. In *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives* (pp. 269–281). IGI Global. https://doi.org/10.4018/978-1-59140-999-1.ch021

Carillo, K. D. A., Huff, S., & Chawner, B. (2014). It's Not Only about Writing Code: An Investigation of the Notion of Citizenship Behaviors in the Context of Free/Libre/Open Source Software Communities. *2014 47th Hawaii International Conference on System Sciences*, 3276–3285. IEEE. https://doi.org/10.1109/HICSS.2014.406

Carraro, J. M. (2014). How Mature is Your Organization when it Comes to UX ? Retrieved July 9, 2020, from UX Magazine website: https://uxmag.com/articles/how-mature-is-your-organization-when-it-comes-to-ux

Carsten, M., Jörn, H., Kindsmüller, M. C., H.A, M., Mohs, C., Hurtienne, J., … IUUI Research Group, D. (2006). IUUI – Intuitive Use of User Interfaces: Auf dem Weg zu einer wissenschaftlichen Basis für das Schlagwort "Intuitivität." *MMI-Interaktiv*, *11*(11), 75–84. Retrieved from https://www.researchgate.net/publication/26447903_IUUI_-_Intuitive_Use_of_User_Interfaces_Auf_dem_Weg_zu_einer_wissenschaftlichen_Basis_fur_das_Schlagwort_Intuitivitat

Carvajal, C. L., & Moreno, A. M. (2017). The maturity of usability maturity models. *Communications in Computer and Information Science*, *770*, 85–99. https://doi.org/10.1007/978-3-319-67383-7_7

Casteel, A., & Bridier, N. (2021). Describing Populations and Samples in Doctoral Student Research. *International Journal of Doctoral Studies*, *16*, 339–362. https://doi.org/10.28945/4766

Castillo-Vergara, M., Alvarez-Marin, A., & Placencio-Hidalgo, D. (2018). A bibliometric analysis of creativity in the field of business economics. *Journal of Business Research, 85*, 1–9. https://doi.org/10.1016/j.jbusres.2017.12.011

Caulfield, J. (2020). How to Do Thematic Analysis | A Step-by-Step Guide & Examples. Retrieved from Scribbr website: https://www.scribbr.com/ methodology/thematic-analysis/

Cenatic. (2010). Report on the International Status of Open Source Software 2010. *Source*. Retrieved from http://www.osepa.eu/site_pages/News/33/international OSS status.pdf

Çetin, G., Verzulli, D., & Frings, S. (2007). An analysis of involvement of HCI experts in distributed software development: Practical issues. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4564 LNCS*, 32–40. https://doi.org/10. 1007/97 8-3-540-73257-0_4

Chang, P.-L., Hsu, C.-W., & Chang, P.-C. (2011). Fuzzy Delphi method for evaluating hydrogen production technologies. *International Journal of Hydrogen Energy*, *36*(21), 14172–14179. https://doi.org/10.1016/ j.ijhydene. 2011.05.045

Chapman, C. (2020). Exploring the Gestalt Principles of Design. Retrieved from Toptal.Com website: https://www.toptal.com/designers/ui/gestalt-principles-of-design

Chapman, L., & Plewes, S. (2014). *A UX Maturity Model: Effective Introduction of UX into Organizations* (A. Marcus, Ed.). https://doi.org/10.1007/978-3-319-07638-6_2

Chełkowski, T., Jemielniak, D., & Macikowski, K. (2021). Free and Open Source Software organizations: A large-scale analysis of code, comments, and commits frequency. *PLOS ONE, 16*(9), e0257192. https://doi.org/10.1371/ journal. pone.0257192

Chen, C.-T. (2000). Extensions of the TOPSIS for group decision-making under fuzzy environment. *Fuzzy Sets and Systems*, *114*(1), 1–9. https://doi.org/10.1016/S0165-0114(97)00377-1

Chen, S.-J., & Hwang, C.-L. (1992). *Fuzzy Multiple Attribute Decision Making Methods*. https://doi.org/10.1007/978-3-642-46768-4_5

Cheng, C.-H., & Lin, Y. (2002). Evaluating the best main battle tank using fuzzy decision theory with linguistic criteria evaluation. *European Journal of Operational Research*, *142*(1), 174–186. https://doi.org/10.1016/S0377-2217(01)00280-6

Cheng, J., & Guo, J. L. C. (2018). How Do the Open Source Communities Address Usability and UX Issues? *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–6. New York, NY, USA: ACM. https://doi.org/10.1145/3170427.3188467

Choma, J., Guerra, E. M., Alvaro, A., Pereira, R., & Zaina, L. (2022). Influences of UX factors in the Agile UX context of software startups. *Information and Software Technology*, *152*. https://doi.org/10.1016/j.infsof.2022.107041

Chrissis, M. B., Konrad, M., & Shrum, S. (2011). *CMMI for Development®: Guidelines for Process Integration and Product Improvement* (3rd editio). Addison-Wesley Professional.

Christian, J., & Vu, A. N. (2021). Task-based structures in open source software: revisiting the onion model. *R and D Management*, *51*(1), 87–100. https://doi.org/10.1111/radm.12428

Chu, H., & Hwang, G. (2008). A Delphi-based approach to developing expert systems with the cooperation of multiple experts. *Expert Systems with Applications*, *34*(4), 2826–2840. https://doi.org/10.1016/j.eswa.2007.05.034

CMMI Institute. (2018). *CMMI® Institute Expands CMMI® V2.0 to Include Services and Supplier Management*. Retrieved from https://cmmiinstitute.com/news/press-releases/december-2018/cmmi®-institute-expands-cmmi®-v2-0-to-include-serv

CMMI Product Team. (2010). CMMI for Development, Version 1.3. *Software Engineering Process Management Program*, (November), 1–520. https://doi.org/CMU/SEI-2010-TR-033 ESC-TR-2010-033

Coelho, B., Andrade, R. M. C., & Darin, T. (2022). Not the Same Everywhere: Comparing the Scope and Definition of User Experience between the Brazilian and International Communities. *International Journal of Human–Computer Interaction*, *38*(7), 595–613. https://doi.org/10.1080/10447318.2021.1960727

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, *20*(1), 37–46. https://doi.org/10.1177/001316446002000104

Cohen, Y. (2022). How to tell if an organization is ready for a true UX culture? Retrieved February 3, 2022, from UX Collective website: https://uxdesign.cc/how-to-tell-if-an-organization-is-ready-for-a-true-ux-culture-4f04b4b476b9

Colford, S. (1986). Explaining Free and Open Source Software Open Source Software in Libraries. *Bulletin of the American Society for Information Science and Technology*, *35*(2), 10–14. Retrieved from http://onlinelibrary.wiley.com/doi/10.1002/bult.2008.1720350205/pdf

Colombo, M. G., Piva, E., & Rossi-Lamastra, C. (2014). Open innovation and within-industry diversification in small and medium enterprises: The case of open source software firms. *Research Policy*, *43*(5), 891–902. https://doi.org/10.1016/j.respol.2013.08.015

Comrey, A. L., & Lee, H. B. (2013). A First Course in Factor Analysis. In *A First Course in Factor Analysis*. Psychology Press. https://doi.org/10.4324/9781315827506

Coral, C., & Bokelmann, W. (2017). The role of analytical frameworks for systemic research design, explained in the analysis of drivers and dynamics of historic land-use changes. *Systems*, *5*(1). https://doi.org/10.3390/systems5010020

Coyle, J. R., & Thorson, E. (2001). The Effects of Progressive Levels of Interactivity and Vividness in Web Marketing Sites. *Journal of Advertising*, *30*(3), 65–77. https://doi.org/10.1080/00913367.2001.10673646

Crahmaliuc, R. (2021). Open-Source or Proprietary Software — What Is Best for Users? Retrieved April 27, 2022, from SIMSCALE Blog website: https://www.simscale.com/blog/2017/06/open-source-vs-proprietary-software/

Cresswell, J. W., Plano-Clark, V. L., Gutmann, M. L., & Hanson, W. E. (2003). Advanced mixed methods research designs. In A. Tashakkori & C. Teddlie (Eds.), *Handbook of Mixed Methods in Social and Behavioral Research* (pp. 209–240). Retrieved from http://www.sagepub.com/upm-data/19291_Chapter_7.pdf

Creswell, J. W., & Guetterman, T. C. (2019). Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research, 6th Edition. In *Pearson*. Retrieved from https://www.pearson.com/us/higher-education/program/Creswell-Educational-Research-Planning-Conducting-and-Evaluating-Quantitative-and-Qualitative-Research-plus-My-Lab-Education-with-Enhanced-Pearson-e-Text-Access-Card-Package-6th-Edition/PGM335066.html

Crilly, N., Moultrie, J., & Clarkson, P. J. (2004). Seeing things: consumer response to the visual domain in product design. *Design Studies*, *25*(6), 547–577. https://doi.org/10.1016/j.destud.2004.03.001

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, *16*(3), 297–334. https://doi.org/10.1007/BF02310555

Crosby, P. B. (1980). *Quality is free: The art of making quality certain*. Mentor; Cover Bent edition.

Crosby, P. B. (1995). Quality is still free: Making quality certain in uncertain times. In 2 (Ed.), *Organizational Dynamics* (Vol. 25). McGraw-Hill.

Crowston, K., & Howison, J. (2005). The social structure of free and open source software development. *First Monday*, *10*(2). https://doi.org/10.5210/fm.v0i0.1478

Crowston, K., Wei, K., Howison, J., & Wiggins, A. (2012). Free/Libre Open-Source Software development: What we know and what we do not know. *ACM Computing Surveys*, *44*(2). https://doi.org/10.1145/2089125.2089127

Cummings, S. R., Browner, W. S., & Hulley, S. B. (2001). Conceiving The Research Question. *Designing Clinical Research*, 17–26.

Curtis, B., Hefley, B., & Miller, S. (1995). People capability maturity model. *Software Engineering Institute*. Retrieved from http://www.sei.cmu.edu/reports/09tr003.pdf

Curtis, P. D., & Mehravari, N. (2015). Evaluating and improving cybersecurity capabilities of the energy critical infrastructure. *2015 IEEE International Symposium on Technologies for Homeland Security (HST)*, 1–6. IEEE. https://doi.org/10.1109/THS.2015.7225323

Cutis, B., Hefley, William, E., & Miller, Sally, A. (2001). People Capability Maturity Model (P-CMM), Version 2.0. *Carnegie Mellon Software Engineering Institute*, *2.0*(July), 10–99. Retrieved from http://www.sei.cmu.edu/reports/01mm001.pdf

Da Silva, T. S., Silveira, M. S., Maurer, F., & Silveira, F. F. (2018). The evolution of agile UXD. *Information and Software Technology*, *102*, 1–5. https://doi.org/10.1016/j.infsof.2018.04.008

Dalkey, N., & Helmer, O. (1963). An Experimental Application of the DELPHI Method to the Use of Experts. *Management Science*, *9*(3), 458–467. https://doi.org/10.1287/mnsc.9.3.458

Damsa, D. (2017). UX Maturity Model: A Tool To Improve Collaboration With The UX Team. Retrieved November 10, 2020, from https://uxstudioteam.com/ux-blog/ux-maturity-model/

Dana, W. (2023). Leveraging AI to Enhance UX Design: Personalization, Predictive Modeling, Natural Language Processing, and Efficiency. Retrieved May 10, 2023, from UX Planet website: https://uxplanet.org/leveraging-ai-to-enhance-ux-design-personalization-predictive-modeling-natural-language-a859f2be6577

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly: Management Information Systems*, *13*(3), 319–339. https://doi.org/10.2307/249008

Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). User Acceptance of Computer Technology: A comparison of two theoretical models. *Management Science*, *35*(8), 982–1003.

Dawood, K. A., Sharif, K. Y., Zaidan, A. A., Abd Ghani, A. A., Zulzalil, H. B., & Zaidan, B. B. (2019). Mapping and Analysis of Open Source Software (OSS) Usability for Sustainable OSS Product. *IEEE Access*, *7*, 65913–65933. https://doi.org/10.1109/ACCESS.2019.2914368

de Bruin, T., Rosemann, M., Freeze, R., & Kulkarni, U. (2005). Understanding the main phases of developing a maturity assessment model. *ACIS 2005 Proceedings - 16th Australasian Conference on Information Systems*.

De Silva, L. C. (2004). Audiovisual emotion recognition. *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, *1*, 649–654. IEEE. https://doi.org/10.1109/ICSMC.2004.1398373

Deprez, J. C., & Alexandre, S. (2008). Comparing Assessment Methodologies for Free/Open Source Software: OpenBRR and QSOS. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*: *Vol. 5089 LNCS* (pp. 189–203). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-69566-0_17

Desmet, P., & Hekkert, P. (2007). Framework of product experience. *International Journal of Design*, *1*(1), 57–66. Retrieved from http://www.ijdesign.org/ojs/index.php/IJDesign/article/viewFile/66/7

Despalatović, L. (2013). The Usability of Free/Libre/Open Source Projects. *Int. J. Comput. Inf. Technol*, *2*(05), 958–963.

Dhir, A., & Al-Kahtani, M. (2013). A case study on user experience (UX) evaluation of mobile augmented reality prototypes. *Journal of Universal Computer Science*, *19*(8), 1175–1196.

Dhir, S., & Dhir, S. (2017). Adoption of open-source software versus proprietary software: An exploratory study. *Strategic Change*, *26*(4), 363–371. https://doi.org/10.1002/jsc.2137

Donahole, S. (2021). How Artificial Intelligence Is Impacting UX Design. Retrieved February 13, 2023, from UXmatters website: https://www.uxmatters.com/mt/archives/2021/04/how-artificial-intelligence-is-impacting-ux-design.php

Dooley, K., Subra, A., & Anderson, J. (2001). Maturity and its impact on new product development project performance. *Research in Engineering Design - Theory, Applications, and Concurrent Engineering*, *13*(1), 23–29. https://doi.org/10.1007/s001630100003

Drahun, G. (2017). UX Maturity Model. Retrieved from https://pt .slideshare.net/Hienadz. Drahun/ux-maturity-models/36

Dresch, A., Lacerda, D. P., & Antunes, J. A. V. (2015). *Design Science Research*. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-07374-3

Duane, A., & O'Reilly, P. (2017). A conceptual stages-of-growth model for managing a social media business profile. *The Irish Journal of Management*, *36*(2), 78–98. https://doi.org/10.1515/ijm-2017-0015

Duffy, M. (2018). Designing a better user experience for open source software. Retrieved July 13, 2021, from https://www.redhat.com/en/blog/designing-better-user-experience-open-source-software

Dujinhouwer, F.-W., & Widdows, C. (2003). *Capgemini Expert Letter - Open Source Maturity Model*.

Durães Dourado, M. A., & Dias Canedo, E. (2018). Usability Heuristics for Mobile Applications - A Systematic Review. *Proceedings of the 20th International Conference on Enterprise Information Systems*, *2*, 483–494. SCITEPRESS - Science and Technology Publications. https://doi.org/10. 5220/00067 81404830494

Earthy, J. V., Bowler, Y., Forster, M., & Taylor, R. (1999). Human factors integration capability maturity model. *IEE Conference Publication*, (463), 320–326. https://doi.org/10.1049/cp:19990208

Ekka, P. M. (2021). *A review of observation method in data collection process*. *6*(12), 17–19.

El Emam, K. (1999). Benchmarking Kappa: interrater agreement in software process assessments. *Empirical Software Engineering*, *4*(2), 113–133. https://doi. org/10.1023/ A:1009820201126

Emam, K. El, & Goldenson, D. R. (2000). An empirical review of software process assessments. In *Advances in Computers* (Vol. 53, pp. 319–423). https://doi.org/10. 1016/S0065 -2458(00)80008-X

Engard, N. C. (2010). Practical Open Source Software for Libraries. In *Practical Open Source Software for Libraries* (Vol. 35). Chandos Publishing. https://doi.org/10.1533/9781780630434

Eriksson, M., & Ferwerda, B. (2021). Towards a User Experience Framework for Business Intelligence. *Journal of Computer Information Systems*, *61*(5), 428–437. https://doi.org/10.1080/08874417.2019.1693936

Etikan, I., Musa, S. A., & Alkassim, R. S. (2016). Comparison of Convenience Sampling and Purposive Sampling. In *American Journal of Theoretical and Applied Statistics* (Vol. 5). https://doi.org/10.11648/j.ajtas.20160501.11

Faber, J., & Fonseca, L. M. (2014). How sample size influences research outcomes. *Dental Press Journal of Orthodontics*, *19*(4), 27–29. https://doi.org/10.1590/2176-9451.19.4.027-029.ebo

Falessi, D., & Reichel, A. (2015). Towards an open-source tool for measuring and visualizing the interest of technical debt. *2015 IEEE 7th International Workshop on Managing Technical Debt (MTD)*, 1–8. IEEE. https://doi.org/ 10.1109/ MTD.2015.7332618

Farooq, M. S., Khan, S. A., Ahmad, F., Islam, S., & Abid, A. (2014). An Evaluation Framework and Comparative Analysis of the Widely Used First Programming Languages. *PLoS ONE*, *9*(2), e88941. https://doi.org/ 10.1371/journal. pone.0088941

Feijo, R. (2010). Planning Your UX Strategy. Retrieved July 9, 2020, from We Made It website: http://johnnyholland.org/2010/04/planning-your-ux-strategy/

Feller, J., & Fitzgerald, B. (2000). a Framework Analysis of the Open Source Software Development Paradigm. *Proceeding ICIS '00 Proceedings of the Twenty First International Conference on Information Systems*, 58–69.

Fendt, O., & Jaeger, M. C. (2019). Open Source for Open Source License Compliance. In *IFIP Advances in Information and Communication Technology* (Vol. 556, pp. 133–138). https://doi.org/10.1007/978-3-030-20883-7_12

Feng, L., & Wei, W. (2019). An empirical study on user experience evaluation and identification of critical UX issues. *Sustainability (Switzerland)*, *11*(8). https://doi.org/10.3390/su11082432

Fernberger, S. W. (1927). The language and thought of the child. *Psychological Bulletin*, *24*(8), 506–506. https://doi.org/10.1037/h0067537

Ferraz, I. N., & dos Santos, C. D. (2022). Transformation of Free and Open Source Software Development Projects: Governance Between the Cathedral and Bazaar. *RAE Revista de Administracao de Empresas*, *62*(1). https://doi.org/10.1590/S0034-759020220104

Ferre, X., Juriste, N., & Moreno, A. M. (2005). Framework for integrating usability practices into the software process. *Lecture Notes in Computer Science, 3547*, 202–215. https://doi.org/10.1007/11497455_17

Ferreira, J., Sharp, H., & Robinson, H. (2012). Agile development and user experience design integration as an ongoing achievement in practice. *Proceedings - 2012 Agile Conference, Agile 2012*, 11–20. https://doi.org/10.1109/Agile.2012.33

Fidel, R., & Pejtersen, A. M. (2004). From information behaviour research to the design of information systems: The Cognitive Work Analysis framework. *Information Research, 10*(1).

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, *76*(5), 378–382. https://doi.org/10.1037/h0031619

Fleiss, J. L., Levin, B., & Paik, M. C. (2003). Statistical Methods for Rates and Proportions. In *Statistical Methods for Rates and Proportions*. Wiley. https://doi.org/10.1002/0471445428

Fletcher, E. (2015). Interpreting qualitative data. *International Journal of Research & Method in Education*, *38*(4), 452–453. https://doi.org/10.1080/ 1743727X. 2015.1066173

Forlizzi, J., & Battarbee, K. (2004). Understanding experience in interactive systems. *DIS2004 - Designing Interactive Systems: Across the Spectrum*, 261–268. https://doi.org/10.1145/1013115.1013152

Forlizzi, J., & Ford, S. (2000). Building blocks of experience: An early framework for interaction designers. *Proceedings of the Conference on Designing Interactive Systems Processes, Practices, Methods, and Techniques - DIS '00*, 419–423. New York, New York, USA, New York, USA: ACM Press. https://doi.org/10.1145/347642.347800

Fornell, C., & Bookstein, F. L. (1982). Two Structural Equation Models: LISREL and PLS Applied to Consumer Exit-Voice Theory. *Journal of Marketing Research*, *19*(4), 440. https://doi.org/10.2307/3151718

Franch, X., Kenett, R., Mancinelli, F., Susi, A., Ameller, D., Annosi, M. C., … Siena, A. (2015). The RISCOSS Platform for Risk Management in Open Source Software Adoption. In *IFIP Advances in Information and Communication Technology* (Vol. 451, pp. 124–133). https://doi.org/10.1007/978-3-319-17837-0_12

Franco-Bedoya, O., Ameller, D., Costal, D., & Franch, X. (2017). Open source software ecosystems: A Systematic mapping. *Information and Software Technology*, *91*, 160–185. https://doi.org/10.1016/j.infsof.2017.07.007

Fraser, J., & Plewes, S. (2015). Applications of a UX Maturity Model to Influencing HF Best Practices in Technology Centric Companies – Lessons from Edison. *Procedia Manufacturing*, *3*(Ahfe), 626–631. https://doi.org/10.1016/j. promfg.2015.07.285

Fredricks, J. A., Blumenfeld, P. C., & Paris, A. H. (2004). School Engagement: Potential of the Concept, State of the Evidence. *Review of Educational Research*, *74*(1), 59–109. https://doi.org/10.3102/00346543074001059

Free Software Foundation. (2012). What is free software? Retrieved January 7, 2022, from Free Software Foundation website: http://www.gnu.org/philosophy/free-sw.html

FSF. (2011). Philosophy of the GNU project. Retrieved from http://www.gnu.org/ philosophy/philosophy.html

Galvis-Lista, E., & Sánchez-Torres, J. M. (2013). A critical review of knowledge management in software process reference models. *Journal of Information Systems and Technology Management*, *10*(2), 323–338. https://doi.org/10. 4301/S1807-17752013000200008

Ganglbauer, E., Schrammel, J., Deutsch, S., & Tscheligi, M. (2011). Applying Psychophysiological Methods for Measuring User Experience: Possibilities, Challenges and Feasibility. *Human-Computer Interaction. INTERACT 2011 (Lecture Notes in Computer Science)*, *6949*, 714–715. Retrieved from http://www.springerlink.com/content/f7380g7k2179484t/

Garcia, A., da Silva, T. S., & Silveira, M. S. (2019). Artifact-Facilitated Communication in Agile User-Centered Design. In *Lecture Notes in Business Information Processing* (Vol. 355, pp. 102–118). https://doi.org/10.1007/978-3-030-19034-7_7

Garzás, J., Pino, F. J., Piattini, M., & Fernández, C. M. (2013). A maturity model for the Spanish software industry based on ISO standards. *Computer Standards and Interfaces*, *35*(6), 616–628. https://doi.org/10.1016/j.csi.2013.04.002

Gay, L., Mills, G., & Airasian, P. (2006). Educational Research: Competencies for Analysis and Applications 8thEdition. In *Upper Saddler River, New Jersey: Pearson Education Inc.* Retrieved from https://eric.ed.gov/?id=ED594591

Gendron, Y. (2013). Rethinking the concept of research contribution. *Comptabilité – Contrôle – Audit*, *19*(2), 133–155.

Gerhardt- Powals, J. (1996). Cognitive engineering principles for enhancing human- computer performance. *International Journal of Human-Computer Interaction*, *8*(2), 189–211. https://doi.org/10.1080/10447319609526147

German, D. M., & Hassan, A. E. (2009). License integration patterns: Addressing license mismatches in component-based development. *2009 IEEE 31st International Conference on Software Engineering*, 188–198. IEEE. https://doi.org/10.1109/ICSE.2009.5070520

Ghasemi, A., & Zahediasl, S. (2012). Normality Tests for Statistical Analysis: A Guide for Non-Statisticians. *International Journal of Endocrinology and Metabolism*, *10*(2), 486–489. https://doi.org/10.5812/ijem.3505

Ghazi, A. N., Petersen, K., Reddy, S. S. V. R., & Nekkanti, H. (2019). Survey Research in Software Engineering: Problems and Mitigation Strategies. *IEEE Access*, *7*, 24703–24718. https://doi.org/10.1109/ACCESS.2018.2881041

Gherardi, S. (2012). How to Conduct a Practice-based Study. In *How to Conduct a Practice-based Study: Problems and Methods*. Edward Elgar Publishing. https://doi.org/10.4337/9780857933386

Gibbons, S. (2023). UX-Stakeholder Engagement 101. Retrieved February 20, 2023, from Nielsen Norman Group website: https://www.nngroup.com/ articles/ stakeholder-engagement/

Gil Urrutia, J. I., Brangier, E., Senderowicz, V., & Cessat, L. (2018). Beyond “Usability and User Experience”, Towards an Integrative Heuristic Inspection: From Accessibility to Persuasiveness in the UX Evaluation. In T. Ahram & C. Falcão (Eds.), *Advances in Intelligent Systems and Computing* (Vol. 607, pp. 460–470). Springer. https://doi.org/10.1007/978-3-319-60492-3_44

Girard, J. M., & Cohn, J. F. (2016). A Primer on Observational Measurement. *Assessment*, *23*(4), 404–413. https://doi.org/10.1177/1073191116635807

Gisev, N., Bell, J. S., & Chen, T. F. (2013). Interrater agreement and interrater reliability: Key concepts, approaches, and applications. *Research in Social and Administrative Pharmacy*, *9*(3), 330–338. https://doi.org/10.1016/j. sapharm. 2012.04.004

Gkogka, E. (2018). Gestalt principles in UI design. Retrieved from Muzli website: https://medium.muz.li/gestalt-principles-in-ui-design-6b75a41e9965

Glassman, M., & Kang, M. J. (2016). Teaching and learning through open source educative processes. *Teaching and Teacher Education*, *60*, 281–290. https://doi.org/10.1016/j.tate.2016.09.002

Goffman, E. (1992). Coding data and interpreting text:methods of analysis. In *Qualitative analysis: practice and innovation* (pp. 80–110). Australia: Allen & Unwin Publishers.

Goh, J. C.-L., & Karimi, F. (2014). Towards the Development of a 'User-Experience' Technology Adoption Model for the Interactive Mobile Technology. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*: *Vol. 8527 LNCS* (pp. 620–630). https://doi.org/10.1007/978-3-319-07293-7_60

Golden, B. (2008). Making Open Source Ready for the Enterprise : The Open Source Maturity Model. Retrieved October 5, 2020, from Open Source Business Resource website: http://timreview.ca/article/145

Gomez, P., Stahel, W. A., & Danuser, B. (2004). Respiratory responses during affective picture viewing. *Biological Psychology*, *67*(3), 359–373. https://doi.org/10.1016/j.biopsycho.2004.03.013

Goyal, A. (2017). Driving Business with User Experience – UX Collective. Retrieved from https://uxdesign.cc/driving-business-with-user-experience-d7f934a41b29

Grabis, J., Zdravkovic, J., & Stirna, J. (2018). Overview of Capability-Driven Development Methodology. In *Capability Management in Digital Enterprises* (pp. 59–84). Cham: Springer International Publishing. https://doi.org/10.1007 /978-3-319-90424-5_4

Grindrod, K. A., Li, M., & Gates, A. (2014). Evaluating user perceptions of mobile medication management applications with older adults: A usability study. *Journal of Medical Internet Research*, *16*(3), e11. https://doi.org/10. 2196/mhealth.3048

Güemes-Peña, D., López-Nozal, C., Marticorena-Sánchez, R., & Maudes-Raedo, J. (2018). Emerging topics in mining software repositories. *Progress in Artificial Intelligence*, *7*(3), 237–247. https://doi.org/10.1007/s13748-018-0147-7

Gunes, H., & Piccardi, M. (2006). A Bimodal Face and Body Gesture Database for Automatic Analysis of Human Nonverbal Affective Behavior. *18th International Conference on Pattern Recognition (ICPR'06)*, *1*, 1148–1153. IEEE. https://doi.org/10.1109/ICPR.2006.39

Guo, F. (2012). More Than Usability The Four Elements of User Experience. Retrieved December 18, 2020, from UXmatters website: http://www.uxmatters.com/mt/archives/2012/04/

Gustafsson, J. (2017). Single case studies vs. multiple case studies: A comparative study. *Academy of Business, Engineering and Science Halmstad University, Sweden*, 1–15. Retrieved from http://www.diva-portal.org/smash/record.jsf? pid=diva2:1064378%0Ahttp://www.diva-portal.org/smash/get/diva2: 1064378/FULLTEXT01.pdf

Guyan, K. (2018). *Analyzing qualitative data :A guide for University and College*. London: Practitioners Advance HE.

Gwebu, K. L., & Wang, J. (2011). Adoption of open source software: The role of social identification. *Decision Support Systems*, *51*(1), 220–229. https://doi.org/10.1016/j.dss.2010.12.010

Haaland, K., Groven, A., Regnesentral, N., Glott, R., & Tannenberg, A. (2010). Free/Libre Open Source Quality Models - a comparison between two approaches. *4th FLOS International Workshop on Free/Libre/Open Source Software*, *0*(180054), 1–2. Retrieved from http://maisqual.squoring.com/wiki/index.php/Free/Libre_Open_Source_Quality_Models

Habibi, A., Jahantigh, F. F., & Sarafrazi, A. (2015). Fuzzy Delphi Technique for Forecasting and Screening Items. *Asian Journal of Research in Business Economics and Management*, *5*(2), 130. https://doi.org/10.5958/2249-7307.2015.00036.5

Haddad, I., Warner, B., The Linux Foundation, & Haddad, Ibrahim, B. W. (2011). The Linux Foundation Understanding the Open Source Development Model. *A White Paper By The Linux Foundation*, (November).

Hanson, N. (2017). A New UX Maturity Model |. Retrieved February 12, 2021, from https://nataliehanson.com/2017/05/05/new-ux-maturity-model/

Harrington, H. J. (2006). The Breakthrough Strategy for Total Quality, Productivity and Competitiveness. In *Food Management* (Vol. 41).

Hartson, R., & Pyla, P. S. (2012). The UX Book: Process and Guidelines for Ensuring a Quality User Experience. In *The UX Book: Process and Guidelines for Ensuring a Quality User Experience* (1st ed.). Amsterdam, Netherlands: Elsevier. https://doi.org/10.1016/C2010-0-66326-7

Harutyunyan, N., Bauer, A., & Riehle, D. (2019). Industry requirements for FLOSS governance tools to facilitate the use of open source software in commercial products. *Journal of Systems and Software*, *158*, 110390. https://doi.org/10.1016/j. jss.2019.08.001

Hasan, L. A., & Al-Sarayreh, K. T. (2015). An Integrated Measurement Model for Evaluating Usability Attributes. *Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication*, 1–6. New York, NY, USA: ACM. https://doi.org/10.1145/2816839.2816861

Hassenzahl, M. (2001). The Effect of Perceived Hedonic Quality on Product Appealingness. *International Journal of Human-Computer Interaction*, *13*(4), 481–499. https://doi.org/10.1207/S15327590IJHC1304_07

Hassenzahl, M. (2003). *The Thing and I: Understanding the Relationship Between User and Product*. https://doi.org/10.1007/1-4020-2967-5_4

Hassenzahl, M. (2007). The Hedonic/Pragmatic Model of User Experience. *Towards a UX Manifesto. COST294-MAUSE Affiliated Workshop*, *2*(September 2007), 10–14. Retrieved from http://portal.acm.org/citation.cfm?id=1531468

Hassenzahl, M. (2010). Experience Design: Technology for All the Right Reasons. In *Synthesis Lectures on Human-Centered Informatics* (Vol. 3). https://doi.org/10.2200/s00261ed1v01y201003hci008

Hassenzahl, M. (2018). The Thing and I (Summer of '17 Remix). In *Heidegger Circle Proceedings* (Vol. 48, pp. 17–31). https://doi.org/10.1007/978-3-319-68213-6_2

Hassenzahl, M., Burmester, M., & Koller, F. (2003). *AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität*. https://doi.org/10.1007/978-3-322-80058-9_19

Hassenzahl, M., Burmester, M., & Koller, F. (2021). User Experience Is All There Is. *I-Com*, *20*(3), 197–213. https://doi.org/10.1515/icom-2021-0034

Hassenzahl, M., & Tractinsky, N. (2006). User experience - A research agenda. *Behaviour and Information Technology*, *25*(2), 91–97. https://doi.org/10.1080/01449290500330331

Hassenzahl, M., Wiklund-Engblom, A., Bengs, A., Hägglund, S., & Diefenbach, S. (2015). Experience-Oriented and Product-Oriented Evaluation: Psychological Need Fulfillment, Positive Affect, and Product Perception. *International Journal of Human-Computer Interaction, 31*(8), 530–544. https://doi.org/ 10.1080/ 10447318.2015.1064664

Hedberg, H., & Iivari, N. (2009). Integrating HCI specialists into open source software development projects. *IFIP Advances in Information and Communication Technology, 299*, 251–263. https://doi.org/10.1007/978-3-642-02032-2_22

Hedberg, H., Iivari, N., Rajanen, M., & Harjumaa, L. (2007). Assuring quality and usability in open source software development. *First International Workshop on Emerging Trends in FLOSS Research and Development, FLOSS'07*, 2–6. https://doi.org/10.1109/FLOSS.2007.2

Heeter, C. (2000). Interactivity in the Context of Designed Experiences. *Journal of Interactive Advertising, 1*(1), 3–14. https://doi.org/10.1080/ 15252019. 2000.10722040

Hellweger, S., & Wang, X. (2015). *What is User Experience Really: towards a UX Conceptual Framework*. Retrieved from http://arxiv.org/abs/1503.01850

Henriquez, V., Moreno, A. M., Calvo-Manzano, J. A., & Feliu, T. S. (2021). *Agile-CMMI Alignment: CMMI V2.0 Contributions and To-dos for Organizations*. Retrieved from http://arxiv.org/abs/2110.00399

Hess, W. (2009). 10 Most Common Misconceptions About User Experience Design. Retrieved June 18, 2022, from Mashable website: http://mashable.com/ 2009/01 /09/ user- experience-design/

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly, 28*(1), 75–105. https://doi. org/https:// doi.org/10.2307/25148625

Hiltunen, M., Laukka, M., & Luomala, J. (2002). *Professional Mobile User Experience*. Cromland.

Hinderks, A. (2015). User Experience Questionnaire. Retrieved from Procedia Computer Science website: http://www.ueq-online.org/

Hinderks, A., Domínguez Mayo, F. J., Thomaschewski, J., & Escalona, M. J. (2022). Approaches to manage the user experience process in Agile software development: A systematic literature review. *Information and Software Technology*, *150*, 106957. https://doi.org/10.1016/j.infsof.2022.106957

Hinderks, A., Schrepp, M., Domínguez Mayo, F. J., Escalona, M. J., & Thomaschewski, J. (2019). Developing a UX KPI based on the user experience questionnaire. *Computer Standards and Interfaces*, *65*(January), 38–44. https://doi.org/10.1016/j.csi.2019.01.007

Hirschheim, R., Klein, H., & Newman, M. (1987). A Social Action Perspective of Information Systems Development. *Proceedings of the 8th International Conference on Information Systems*, 45–56. Retrieved from http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1006&context=icis1987

Hoaglin, D. C., & Velleman, P. F. (1995). A Critical Look at Some Analyses of Major League Baseball Salaries. *The American Statistician*, *49*(3), 277–285. https://doi.org/10.1080/00031305.1995.10476165

Hogan, S. J., Soutar, G. N., McColl-Kennedy, J. R., & Sweeney, J. C. (2011). Reconceptualizing professional service firm innovation capability: Scale development. *Industrial Marketing Management*, *40*(8), 1264–1273. https://doi.org/10.1016/j.indmarman.2011.10.002

Hokkanen, L., & Väänänen-Vainio-Mattila, K. (2015). UX Work in Startups: Current Practices and Future Needs. In *Lecture Notes in Business Information Processing* (Vol. 212, pp. 81–92). https://doi.org/10.1007/978-3-319-18612-2_7

Hoover, A., & Stewart, K. (2020). Cybersecurity Maturity Model Certification (CMMC) Part 2: Process Maturity's Role in Cybersecurity. Retrieved from https://insights.sei.cmu.edu/blog/cybersecurity-maturity-model-certification-cmmc-part-2-process-maturitys-role-in-cybersecurity/

Hornbaek, K. (2010). Dogmas in the assessment of usability evaluation methods. *Behaviour and Information Technology*, *29*(1), 97–111. https://doi.org/10.1080/01449290801939400

Hornbæk, K., & Hertzum, M. (2017). Technology acceptance and user experience: A review of the experiential component in HCI. *ACM Transactions on Computer-Human Interaction*, *24*(5). https://doi.org/10.1145/3127358

Hou, L., Liu, Q., Saeed, K., Ali Haidery, S., Uddin, M. I., & Khattak, H. (2021). Enhancement of the Capability Maturity Model for Improving the Quality of Software Projects in Developing Countries. *Scientific Programming*, *2021*, 1–10. https://doi.org/10.1155/2021/9982227

Huber, M., Wolf, M., Meyer, W., Jokisch, O., & Nowack, K. (2018). Some design aspects of a cognitive user interface. *Online Journal of Applied Knowledge Management*, *6*(1), 15–29. https://doi.org/10.36965/OJAKM.2018.6(1)15-29

Huisman, G., & Van Hout, M. (2008). The development of a graphical emotion measurement instrument using caricatured expressions: the LEMtool. *Emotion in HCI-Designing for People: Proceedings of the 2008 International Workshop*, 5–7. Retrieved from http://www.emotion-in-hci.net/proceedings/ emotion_ workshop_ procs_2008.pdf#page=8

Humphrey, W. S., & Sweet, W. L. (1987). A Method for Assessing Capability of Contractors Capability of Contractors. *Software Engineering Institute*, (September), 1–46. Retrieved from https://apps.dtic.mil/sti/ pdfs/ ADA187230.pdf

Hussain, A., Mkpojiogu, E. O. C., & Husin, M. Z. (2021). Dimensioning UX Models for Design and Evaluation. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(3), 1878–1883. https://doi.org/10. 17762/ turcomat. v12i3.1018

Hussein, I., & Mahmud, M. (2012). User Experience Design (UXD) A survey of user interface development practices in Malaysia. *Proceedings of the Southeast Asian Network of Ergonomics Societies Conference*, 1–6.

Hussein, I., Mahmud, M., & Md Tap, A. O. (2015). A survey of user experience practice: A point of meet between academic and industry. *Proceedings - 2014 3rd International Conference on User Science and Engineering: Experience. Engineer. Engage, i-USEr 2014*, 62–67. https://doi.org/10.1109/ IUSER. 2014.7002678

Hustak, T., & Krejcar, O. (2016). Principles of Usability in Human-Computer Interaction. In *Lecture Notes in Electrical Engineering* (Vol. 354, pp. 51–57). https://doi.org/10.1007/978-3-662-47895-0_7

Hwang, W., & Salvendy, G. (2010). Number of people required for usability evaluation. *Communications of the ACM*, *53*(5), 130–133. https://doi.org/10. 1145/1735223.1735255

IDF. (2018). User Experience (UX) Design. Retrieved February 3, 2021, from Idf website: https://www.interaction-design.org/literature/topics/ux-design

Iivari, J. (1986). Dimensions of information systems design: A framework for a long-range research program. *Information Systems*, *11*(2), 185–197. https://doi.org/10.1016/0306-4379(86)90007-4

Iivari, J., Hirschheim, R., & Klein, H. K. (1998). A Paradigmatic Analysis Contrasting Information Systems Development Approaches and Methodologies. *Information Systems Research*, *9*(2), 164–193. https://doi.org/10. 1287/isre.9.2.164

Iivari, J., & Venable, J. (2009). Action research and design science research - Seemingly similar but decisively dissimilar. *17th European Conference on Information Systems, ECIS 2009*.

Iivari, N. (2006). Understanding the work of an HCI practitioner. *Proceedings of the 4th Nordic Conference on Human-Computer Interaction: Changing Roles*, *189*, 185–194. New York, NY, USA: ACM. https://doi.org/10. 1145/ 1182475.1182495

Iivari, N. (2010). Culturally compatible usability work: An interpretive case study on the relationship between usability work and its cultural context in software product development organizations. *Journal of Organizational and End User Computing*, *22*(3), 40–65. https://doi.org/10.4018/joeuc.2010070103

Iivari, N., Hedberg, H., & Kirves, T. (2008). Usability in Company Open Source Software Context - Initial Findings from an Empirical Case Study. In *IFIP International Federation for Information Processing* (Vol. 275, pp. 359–365). https://doi.org/10.1007/978-0-387-09684-1_33

Imenda, S. (2014). Is There a Conceptual Difference between Theoretical and Conceptual Frameworks? *Journal of Social Sciences*, *38*(2), 185–195. https://doi.org/10.1080/09718923.2014.11893249

In, J. (2017). Introduction of a pilot study. *Korean Journal of Anesthesiology*, *70*(6), 601. https://doi.org/10.4097/kjae.2017.70.6.601

Inal, Y., Clemmensen, T., Business, C., Rajanen, D., Iivari, N., Rizvanoglu, K., & Sivaji, A. (2020). Positive Developments but Challenges Still Ahead: A Survey Study on UX Professionals' Work Practices. *Journal of Usability Studies*, *15*(4), 210–246.

İnal, Y., & Güner, H. (2016). Understanding software developers' awareness and knowledge about user experience and usability. *Pamukkale University Journal of Engineering Sciences*, *22*(5), 384–389. https://doi.org/ 10.5505/ pajes.2016.83792

Inostroza, R., Rusu, C., Roncagliolo, S., Rusu, V., & Collazos, C. A. (2016). Developing SMASH: A set of SMArtphone's uSability Heuristics. *Computer Standards & Interfaces*, *43*, 40–52. https://doi.org/10.1016/j.csi.2015.08.007

International Organization for Standardisation. (2015). *ISO/IEC 33001:2015, Information technology — Process assessment — Concepts and terminology*. 19. Retrieved from https://www.iso.org/obp/ui/#iso:std:iso-iec:33001:ed-1:v1:en

International Organisation for Standardisation (ISO). (2018). Ergonomics of human-system interaction - Part 11: Usability: Definitions and Concepts. *Iso 9241-11:2018(E)*, 2. Retrieved from https://infostore.saiglobal. com/ preview/is/en/2018/i.s. eniso9241-11-2018. pdf?sku=1980667%0Ahttps: // www.sis. se/api/ document/preview/80003410/

Ishan. (2020). The psychology in UX design. *UX Collective*. Retrieved from https://uxdesign.cc/the-psychology-of-ux-design-859439bc8a32

Ishikawa, A., Amagasa, M., Shiga, T., Tomizawa, G., Tatsuta, R., & Mieno, H. (1993). The max-min Delphi method and fuzzy Delphi method via fuzzy integration. *Fuzzy Sets and Systems*, *55*(3), 241–253. https://doi.org/10.1016/0165-0114(93)90251-C

ISO/IEC. (2004). *Information technology - Process assessment - Part 1: Concepts and vocabulary*. 28. Retrieved from http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=38932

ISO/IEC. (2010). *Information technology- Software process assessment Part 2: A reference model for processes and process capability*. *1998*, 1–130.

ISO/IEC. (2011). *ISO/IEC 25010:2011 - Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models*. Retrieved from http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=35733

ISO/IEC. (2018). 9241-11. Ergonomic requirements for office work with visual display terminals (VDTs). In *The international organization for standardization* (Vol. 45).

ISO/IEC. (2019a). ISO/IEC 33020:2019 Information technology — Process assessment — Process measurement framework for assessment of process capability. *Software and Systems Engineering*, 42. Retrieved from https://www.iso.org/standard/78526.html

ISO/IEC. (2019b). ISO 9241-210:2019(en) Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems. *International Standard*, *2*, 1–33.

Isomursu, M., Sirotkin, A., Voltti, P., & Halonen, M. (2012). User Experience Design Goes Agile in Lean Transformation -- A Case Study. *2012 Agile Conference*, 1–10. IEEE. https://doi.org/10.1109/Agile.2012.10

Ivanyos, J. (2009). Implementing COBIT based Process Assessment Model for Evaluating IT Controls. *ECIIA–Yearbook of Internal Audit: 2009/2010: Best …*, 1–16. Retrieved from http://books.google.com/books? hl=en&lr=&id= VtMMzw4g8P4C&oi=fnd&pg=PA99&dq=Implementing+COBIT+based+Process+Assessment+Model+for+Evaluating+IT+Controls&ots=fGAoJSuZJg&sig= WKjVvPdt5jq1yKQqTxjaVdMwyEg

Jacobvitz, D., Curran, M., & Moller, N. (2002). Conducting in-depth interviews: A Guide for Designing and Conducting In-Depth Interviews for Evaluation Input. *Attachment and Human Development*, *4*(2), 207–215.

Jäkel, F., Singh, M., Wichmann, F. A., & Herzog, M. H. (2016). An overview of quantitative approaches in Gestalt perception. *Vision Research*, *126*, 3–8. https://doi.org/10.1016/j.visres.2016.06.004

Jarana- Díaz, T., Romero- Martín, M., Ponce- Blandón, J. A., & Jiménez- Picón, N. (2021). Integrative Review of Related Factors and Defining Characteristics of Lack of Family Integrity. *International Journal of Nursing Knowledge, 32*(1), 44–52. https://doi.org/10.1111/2047-3095.12289

Jiang, J.-Y., Cheng, P.-J., & Wang, W. (2017). Open Source Repository Recommendation in Social Coding. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1173–1176. New York, NY, USA: ACM. https://doi.org/10. 1145/30771 36.3080753

Jimenez, C., Rusu, C., Roncagliolo, S., Inostroza, R., & Rusu, V. (2012). Evaluating a Methodology to Establish Usability Heuristics. *2012 31st International Conference of the Chilean Computer Science Society*, 51–59. IEEE. https://doi.org/10.1109/SCCC.2012.14

Johnson, J. (2014). Designing with the Mind in Mind. In *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines, Second Edition*. Elsevier. https://doi.org/10.1016/C2012-0-07128-1

Johnston, K., Begg, S., & Tanner, M. (2013). Exploring the factors influencing the adoption of Open Source Software in Western Cape schools. *International Journal of Education and Development Using Information and Communication Technology*, *9*(2), 64. Retrieved from http://hec.summon.serialssolutions.com/ 2.0.0/link/0/eLvHCXMwzV07b8IwELaAqUvV0id96CaWiApIQsLQgdLSSoxF 6lZksANIEBAkqvrv6_MjMQU6d4rkvO0vvvPlu-8IgaDRpAx_KAlXoE79s F33xq1wFHBh3EKXYRi732_3nr3Oq_9aKBiZikmaKFLTw5T_nV7yjwc3Z86 h32hq5sx0wRHTTtlyZRxCJI447zJSLzZR8oWMr1n

Jokela, T., & Abrahamsson, P. (2000). Modelling Usability Capability – Introducing the Dimensions. In F. Bomarius & M. Oivo (Eds.), *Product Focused Software Process Improvemen* (pp. 73–87). Berlin Heidelberg. https://doi.org/10. 1007/978-3-540-45051-1_10

Jokela, T., Siponen, M., Hirasawa, N., & Earthy, J. (2006). A survey of usability capability maturity models: implications for practice and research. *Behaviour & Information Technology*, *25*(3), 263–282. https://doi.org/10. 1080/01 449290500168079

Jokonya, O., Kroeze, J. H., & van der Poll, J. A. (2013). A framework to analyze E-Government OSS adoption benefits. *19th Americas Conference on Information Systems, AMCIS 2013 - Hyperconnected World: Anything, Anywhere, Anytime*, *1*, 109–117.

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *374*(2065), 20150202. https://doi.org/10.1098/rsta.2015.0202

Jones, N. A., Ross, H., Lynam, T., Perez, P., & Leitch, A. (2011). Mental models: An interdisciplinary synthesis of theory and methods. *Ecology and Society*, *16*(1). https://doi.org/10.5751/ES-03802-160146

Jordan, P. (2000). Designing Pleasurable Products. In *Designing Pleasurable Products*. CRC Press. https://doi.org/10.1201/9780203305683

Jöreskog, K., & Wold, H. (1982). Systems under indirect observation using PLS. *A Second Generation of Multivariate Analysis*, 325–347.

Jørgensen, N. (2001). Putting it all in the trunk: Incremental software development in the FreeBSD open source project. *Information Systems Journal*, *11*(4), 321–336. https://doi.org/10.1046/j.1365-2575.2001.00113.x

Jung, S., & Lee, S. (2016). Developing a Model for Continuous User Engagement in Social Media. *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication*, 1–4. New York, NY, USA: ACM. https://doi.org/10.1145/2857546.2857566

Kadir, W. N. H. W. A., Abdullah, N. S. Y., & Mustapha, I. R. (2019). The application of the fuzzy delphi technique on a component of development of form four STEM-based physics interactive laboratory (I-lab). *International Journal of Scientific and Technology Research*, *8*(12), 2908–2912.

Kaiser, H. F. (1960). The Application of Electronic Computers To Factor Analysis. *Educational and Psychological Measurement*, *20*(1), 141–151.

Kaiser, H. F. (1970). A second generation little jiffy. *Psychometrika*, *35*(4), 401–415. https://doi.org/10.1007/BF02291817

Kamau, J., & Namuye, S. (2012). A Review of Users Adoption of Open Source Software in Africa. *Computer and Information Science*, *5*(5). https://doi.org/ 10.5539/cis.v5n5p45

Kamler, B., & Thomson, P. (2006). Helping Doctoral Students Write. In *Helping Doctoral Students Write*. Routledge. https://doi.org/10.4324/9780203969816

Kaplan, B., & Maxwell, J. A. (2006). Qualitative Research Methods for Evaluating Computer Information Systems. In *Evaluating the Organizational Impact of Healthcare Information Systems* (pp. 30–55). New York: Springer-Verlag. https://doi.org/10.1007/0-387-30329-4_2

Karahanoğlu, A., & Bakırlıoğlu, Y. (2020). Evaluation of the usefulness of path of long-term user experience model in design process. *Behaviour and Information Technology*, 1–19. https://doi.org/10.1080/0144929X.2020.1836256

Karapanos, E. (2010). *Quantifying Diversity in User Experience*.

Karapanos, E., Zimmerman, J., Forlizzi, J., & Martens, J.-B. (2009). User experience over time. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 729–738. New York, NY, USA: ACM. https://doi. org/10.1145/ 1518701.1518814

Kashfi, P., Feldt, R., & Nilsson, A. (2019). Integrating UX principles and practices into software development organizations: A case study of influencing events. *Journal of Systems and Software*, *154*, 37–58. https://doi.org/10.1016/j.jss.2019.03.066

Kashfi, P., Nilsson, A., & Feldt, R. (2017). Integrating user eXperience practices into software development processes: Implications of the UX characteristics. *PeerJ Computer Science*, *2017*(10), e130. https://doi.org/10.7717/peerj-cs.130

Kazimierczak, M., Breckwoldt, J. A., & Wajsman, N. (2020). Open source software in the European Union. *European Union Intellectual Property Office*. https://doi.org/10.2814/866548

Kelvar, K. (2021). What sample size do you really need for UX research? Retrieved June 15, 2021, from UserZoom website: https://www.userzoom.com/blog/what-sample-size-do-you-really-need-for-ux-research/

Kendall, M. G., & Smith, B. B. (1939). The Problem of m Rankings. *The Annals of Mathematical Statistics*, *10*(3), 275–287. https://doi.org/10.1214/aoms/1177732186

Kerrigan, M. (2013). A capability maturity model for digital investigations. *Digital Investigation*, *10*(1), 19–33. https://doi.org/10.1016/j.diin.2013.02.005

Kerzner, H. (2001). Strategic Planning for Project Management, Using a Project Management Maturity Model. In *Griffith University CI 8252EDN* (1st ed.). Retrieved from http://metalab.uniten.edu.my/ ~abdrahim/mitm743/ strategicforPM.pdf

Khanjani, A., & Sulaiman, R. (2011). The process of quality assurance under open source software development. *2011 IEEE Symposium on Computers & Informatics*, 548–552. IEEE. https://doi.org/10.1109/ISCI.2011.5958975

Kieffer, S., Rukonić, L., Kervyn de Meerendré, V., & Vanderdonckt, J. (2020). A Process Reference Model for UX. In *Communications in Computer and Information Science*: *Vol. 1182 CCIS* (pp. 128–152). https://doi.org/10.1007/978-3-030-41590-7_6

Kilamo, T., Lenarduzzi, V., Ahoniemi, T., Jaaksi, A., Rahikkala, J., & Mikkonen, T. (2020). How the Cathedral Embraced the Bazaar, and the Bazaar Became a Cathedral. In *IFIP Advances in Information and Communication Technology*. Springer International Publishing. https://doi.org/10.1007/978-3-030-47240-5_14

Kim, B. J., & Kleinschmit, S. W. (2012). A logistic multilevel model for civic engagement and community group impact in the digital age. *Proceedings of the 6th International Conference on Theory and Practice of Electronic Governance - ICEGOV '12*, 34. New York, New York, USA: ACM Press. https://doi.org/10.1145/2463728.2463737

Kim, H.-Y. (2013). Statistical notes for clinical researchers: assessing normal distribution (2) using skewness and kurtosis. *Restorative Dentistry & Endodontics*, *38*(1), 52. https://doi.org/10.5395/rde.2013.38.1.52

Kirakowski, J., & Corbett, M. (1993). SUMI: the Software Usability Measurement Inventory. *British Journal of Educational Technology*, *24*(3), 210–212. https://doi.org/10.1111/j.1467-8535.1993.tb00076.x

Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. EBSE Technical Report Nr. EBSE-2007-01. In *EBSE Technical Report*. Durham. Retrieved from http://www.dur.ac.uk/ebse/resources/Systematic-reviews-5-8.pdf

Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering-A tertiary study. *Information and Software Technology*, *52*(8), 792–805. https://doi.org/10.1016/j.infsof.2010.03.006

Kobyliński, A. (2013). *The Relationships between Software Development Processes and Software Product Quality*. https://doi.org/10.1007/978-3-642-40823-6_13

Kocaballi, A. B., Laranjo, L., & Coiera, E. (2019). Understanding and Measuring User Experience in Conversational Interfaces. *Interacting with Computers*, *31*(2), 192–207. https://doi.org/10.1093/iwc/iwz015

Kompaniets, V., Lyz, A., & Kazanskaya, A. (2020). An Empirical Study of Goal Setting in UX/UI-design. *2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT)*, 1–5. IEEE. https://doi.org/10.1109/AICT50176.2020.9368570

Kort, J., Vermeeren, A., & Fokker, J. (2007). Conceptualizing and Measuring User eXperience. *Towards a UX Manifesto, COST294-MAUSE Affiliated Workshop*, 57–64. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/ download?doi= 10.1.1.177.7089&rep=rep1&type=pdf#page=63

Kothari, C. (2004). Research methodology: methods and techniques. In *New Age International*. https://doi.org/http://196.29.172. 66:8080/jspui/bitstream/ 1234 56789/2574/ 1/Research%20Methodology.pdf

Kou, Y., & Gray, C. M. (2018). Towards professionalization in an online community of emerging occupation: Discourses among UX practitioners. *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work*, 322–334. https://doi.org/10.1145/3148330.3148352

Kour, R., Karim, R., & Thaduri, A. (2020). Cybersecurity for railways – A maturity model. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, *234*(10), 1129–1148. https://doi.org/10. 1177/0954409719881849

Krawiec, Ł., & Dudycz, H. (2020a). A comparison of heuristics applied for studying the usability of websites. *Procedia Computer Science*, *176*, 3571–3580. https://doi.org/10.1016/j.procs.2020.09.029

Krawiec, Ł., & Dudycz, H. (2020b). Using Heuristics for Assessing the Usability of the Public Information Bulletins in Poland. In *Lecture Notes in Business Information Processing*: *Vol. 380 LNBIP* (pp. 63–75). https://doi.org/10.1007/978- 3-030-43353-6_4

Kremer, S., Schlimm, A., & Lindemann, U. (2017). The ExodUX framework: Supporting comprehensive user experience design. *PICMET 2017 - Portland International Conference on Management of Engineering and Technology: Technology Management for the Interconnected World, Proceedings*, *2017-Janua*, 1–10. https://doi.org/10.23919/PICMET.2017.8125371

Krishnamurthy, A., O'Connor, R. V., & O'Connor, R. V. (2013). An Analysis of the Software Development Processes of Open Source E-Learning Systems. In *Communications in Computer and Information Science*: *Vol. 364 CCIS* (pp. 60–71). https://doi.org/10.1007/978-3-642-39179-8_6

Kronbauer, A. H., Machado, D., & Santos, C. A. S. (2015). *Capture and Analysis of Interaction Data for the Evaluation of User Experience with Mobile Devices*. https://doi.org/10.1007/978-3-319-20898-5_6

Kruskal, W. H. (1960). Some Remarks on Wild Observations. *Technometrics*, *2*(1), 1–3. https://doi.org/10.1080/00401706.1960.10489875

Kundu, A., Swain, D., & Biswas, S. (2010). Application of Free and Open source software and its Impact on society. *International Journal of Computer Sciences and Information Technology*, *1*(4), 226–229.

Kuo, M. S., & Liang, G. S. (2012). A soft computing method of performance evaluation with MCDM based on interval-valued fuzzy numbers. *Applied Soft Computing Journal*, *12*(1), 476–485. https://doi.org/10.1016/j.asoc.2011.08.020

Kurosu, M. (2017). *Human-Computer Interaction. User Interface Design, Development and Multimodality*. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-58071-5

Kuusinen, K. (2015). Task Allocation Between UX Specialists and Developers in Agile Software Development Projects. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 9298, pp. 27–44). https://doi.org/10.1007/978-3-319-22698-9_3

Kuutti, K. (2010). Where are the Ionians of user experience research? *Proceedings of the 6th Nordic Conference on Human-Computer Interaction Extending Boundaries* - *NordiCHI '10*, 715. New York, New York, USA: ACM Press. https://doi.org/10.1145/1868914.1869012

Kuvaja, P., & Bicego, A. (1994). BOOTSTRAP ? a European assessment methodology. *Software Quality Journal*, *3*(3), 117–127. https://doi.org/10.1007/BF00402292

Kuwata, Y., Takeda, K., & Miura, H. (2014). A study on maturity model of open source software community to estimate the quality of products. *Procedia Computer Science*, *35*(C), 1711–1717. Elsevier Masson SAS. https://doi.org/10.1016/ j.procs.2014.08.264

Kwak, Y. H., & Ibbs, C. W. (2002). Project Management Process Maturity (PM)2 Model. *Journal of Management in Engineering*, *18*(3), 150–155. https://doi.org/10.1061/(ASCE)0742-597X(2002)18:3(150)

Lacerda, T. C., & von Wangenheim, C. G. (2018). Systematic literature review of usability capability/maturity models. *Computer Standards & Interfaces*, *55*, 95–105. https://doi.org/10.1016/j.csi.2017.06.001

Lahti, C. B., & Peterson, R. (2007). Overview – The Goals of This Book. In *Sarbanes-Oxley IT Compliance Using Open Source Tools* (pp. 1–34). Elsevier. https://doi.org/10.1016/b978-1-59749-216-4.00001-x

Lakhani, K. (2017). User Experience Assurance (UXA) — UX Testing. Retrieved from Medium website: https://medium.com/@kiritlakhani/user-experience-assurance-uxa-ux-testing-fc200aef64ea

Lakhe, R. R., & Mohanty, R. P. (1994). Total Quality Management. *International Journal of Quality & Reliability Management*, *11*(9), 9–33. https://doi.org/10.1108/02656719410074279

Lallemand, C., Gronier, G., & Koenig, V. (2015). User experience: A concept without consensus? Exploring practitioners' perspectives through an international survey. *Computers in Human Behavior*, *43*, 35–48. https://doi.org/10.1016/j.chb.2014.10.048

Lallemand, C., Koenig, V., & Gronier, G. (2014). How relevant is an expert evaluation of user experience based on a psychological needs-driven approach? *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, 11–20. New York, NY, USA: ACM. https://doi.org/10.1145/2639189.2639214

Lalmas, M., O'Brien, H., & Yom-Tov, E. (2014). Measuring User Engagement. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, *6*(4), 1–132. https://doi.org/10.2200/S00605ED1V01Y201410ICR038

Landis, J. R., & Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, *33*(1), 159. https://doi.org/10.2307/2529310

Lapão, L. V., da Silva, M. M., & Gregório, J. (2017). Implementing an online pharmaceutical service using design science research. *BMC Medical Informatics and Decision Making*, *17*(1), 31. https://doi.org/10.1186/s12911-017-0428-2

Laplante, P., Gold, A., & Costello, T. (2007). Open Source Software: Is It Worth Converting? *IT Professional*, *9*(4), 28–33. https://doi.org/10. 1109/ MITP.2007.72

Lasrado, L. A., Vatrapu, R., & Andersen, K. N. (2015). Maturity Models Development in IS Research: A Literature Review. *Proceedings of the 38th Information Systems Research Seminar in Scandinavia (IRIS 38)*, *6*(August 9-12), 1–12.

Laszlo, G. (2009). Issues and Aspects of Open Source Software Usage and Adoption in the Public Sector. In *Software Applications* (pp. 1577–1591). IGI Global. https://doi.org/10.4018/978-1-60566-060-8.ch093

Laurel, B. (1997). Interface agents: Metaphors with character. *Human Values and the Design of Computer Technology*, 207–219. Retrieved from http://www.google.com.ep.fjernadgang.kb.dk/books?hl=da&lr=&id=-kUysua5GwcC&oi=fnd&pg=PA207&dq=interface+agents&ots=6GgKYc54ml&sig=ZuDyZ7FWyObuei_SXcRtVClZikg%5Cnhttp://www.google.dk.ep.fjernadgang.kb.dk/books?hl=da&lr=&id=-kUysua5GwcC&oi=fnd&pg=PA207&dq=i

Laurent, A. M. S. (2004). Understanding Open Source and Free Software Licensing. In *Ariadne*. Retrieved from http://www.ariadne.ac.uk/issue42/fraser-rvw/

Lavrakas, P. (2013). Target Population. In *Encyclopedia of Survey Research Methods*. 2455 Teller Road, Thousand Oaks California 91320 United States of America: Sage Publications, Inc. https://doi.org/10.4135/9781412963947.n571

Law, E. L.-C. (2011). The measurability and predictability of user experience. *Proceedings of the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing Systems - EICS '11*, 1. New York, New York, USA: ACM Press. https://doi.org/10.1145/1996461.1996485

Law, E. L.-C. C., & Abrahão, S. (2014). Interplay between User Experience (UX) evaluation and system development. *International Journal of Human Computer Studies*, *72*(6), 523–525. https://doi.org/10.1016/j.ijhcs.2014.03.003

Law, E. L.-C., van Schaik, P., & Roto, V. (2014). Attitudes towards user experience (UX) measurement. *International Journal of Human-Computer Studies*, *72*(6), 526–541. https://doi.org/10.1016/j.ijhcs.2013.09.006

Law, E. L. C., & Van Schaik, P. (2012). To measure or not to measure UX: An interview study. *CEUR Workshop Proceedings*, *922*, 58–63.

LeCompte, M. D., & Schensul, J. J. (1999). Analyzing & interpreting ethnographic data. In *Ethnographer's toolkit v. 5*.

Lee, C.-S., Chen, Y.-C., Tsui, P.-L., Che, C.-W., & Chiang, M.-C. (2021). Application of Fuzzy Delphi Technique Approach in Sustainable Inheritance of Rural Cooking Techniques and Innovative Business Strategies Modeling. *Agriculture*, *11*(10), 924. https://doi.org/10.3390/agriculture11100924

Lehmann, J., Lalmas, M., Yom-Tov, E., & Dupret, G. (2012). Models of User Engagement. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*: *Vol. 7379 LNCS* (pp. 164–175). https://doi.org/10.1007/978-3-642-31454-4_14

Lemon, K., Allen, E. B., Carver, J. C., & Bradshaw, G. L. (2007). An empirical study of the effects of Gestalt principles on diagram understandability. *Proceedings - 1st International Symposium on Empirical Software Engineering and Measurement, ESEM 2007*, 156–165. https://doi.org/10. 1109/ ESEM.2007.17

Lenarduzzi, V., Taibi, D., Tosi, D., Lavazza, L., & Morasca, S. (2020). Open Source Software Evaluation, Selection, and Adoption: A Systematic Literature Review. *Proceedings - 46th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2020*, 437–444. https://doi.org/10.1109/ SEAA51224.2020.00076

Lenarduzzi, V., Tosi, D., Lavazza, L., & Morasca, S. (2019). Why Do Developers Adopt Open Source Software? Past, Present and Future. In *IFIP Advances in Information and Communication Technology* (Vol. 556, pp. 104–115). https://doi.org/10.1007/978-3-030-20883-7_10

Lerner, J., & Tirole, J. (2003). Some Simple Economics of Open Source. *The Journal of Industrial Economics*, *50*(2), 197–234. https://doi.org/10.1111/1467-6451.00174

Lethbridge, T., & Laganiere, R. (2004). *Object-oriented software engineering: Practical Software Engineering Using UML and Java* (2nd Editio). McGraw-Hill Publishing Company.

Levy, J. (2015). UX Strategy: How to Devise Innovative Digital Products that People Want. In *How to Devise Innovative Digital Products that People Want*. Retrieved from http://books.google.co.za/books? id=3SB0CQAAQBAJ&pg=PT403&dq=intitle:UX+Strategy+inauthor:Jaime+Levy&hl=&cd=1&source=gbs_api%0Apapers3://publication/uuid/03C41609-D071-4AC7-AFDA-CA12501594C2

Lewis, J. R. (1982). Testing Small System Customer Set-Up. *Proceedings of the Human Factors Society Annual Meeting*, *26*(8), 718–720. https://doi. org/10.1177/ 154193128202600810

Lewis, J. R. (1991). an After-Scenario Questionnaire for Usability Studies: Psychometric Evaluation Over Three Trials. *SIGCHI Bull.*, *23*(4), 79. Retrieved from http://0-delivery.acm.org.impulse.ucdenver. edu/10.1145/1060000/ 1056077/p79-lewis.pdf?ip=140.226.6.63& CFID= 28539265&CFTOKEN= 62872425 &__acm__=1308928978_4b5dd5eba75ff580e09941a280d739c7

Lewis, J. R. (2001). Evaluation of Procedures for Adjusting Problem-Discovery Rates Estimated From Small Samples. *International Journal of Human-Computer Interaction*, *13*(4), 445–479. https://doi.org/10.1207/ S15327590 IJHC1304_06

Li, J., Conradi, R., Bunse, C., Torchiano, M., Slyngstad, O. P. N., & Morisio, M. (2009). Development with Off-the-Shelf Components: 10 Facts. *IEEE Software*, *26*(2), 80–87. https://doi.org/10.1109/MS.2009.33

Liedtke, C., Jolanta Welfens, M., Rohn, H., & Nordmann, J. (2012). LIVING LAB: User-driven innovation for sustainability. *International Journal of Sustainability in Higher Education*, *13*(2), 106–118. https://doi.org/10.1108/ 1467637 1211211809

Lim, Y., Stolterman, E., Jung, H., & Donaldson, J. (2007). Interaction gestalt and the design of aesthetic interactions. *Proceedings of the 2007 Conference on Designing Pleasurable Products and Interfaces - DPPI '07,* 239. New York, New York, USA: ACM Press. https://doi.org/10.1145/1314161.1314183

Lin, H. X., Choong, Y. Y., & Salvendy, G. (1997). A proposed index of usability: A method for comparing the relative usability of different software systems. *Behaviour and Information Technology*, *16*(4–5), 267–277. https://doi.org/ 10.1080/ 014492997119833

Linnemeier, M., Lin, Y.-Y., Laput, G., & Vijjapurapu, R. (2012). StoryCubes. *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, 1321–1326. New York, NY, USA: ACM. https://doi.org/10.1145/2212776.2212447

Linse, C. (2017). *Ambiguity at the heart of design work Sensing and negotiating ambiguity in knowledge-creation work*. Retrieved from http://kth.diva-portal.org/smash/get/diva2:1093346/FULLTEXT03.pdf

Lizano, F., Sandoval, M. M., Bruun, A., & Stage, J. (2013). Is usability evaluation important: The perspective of novice software developers. *HCI 2013 - 27th International British Computer Society Human Computer Interaction Conference: The Internet of Things*, (2007), 1–6. https://doi.org/10.14236/ ewic/hci2013.40

Llerena, L., Castro, J. W., & Acuña, S. T. (2019). A pilot empirical study of applying a usability technique in an open source software project. *Information and Software Technology*, *106*, 122–125. https://doi.org/10.1016/j. infsof. 2018.09.007

Llerena, L., Rodriguez, N., Castro, J. W., & Acuña, S. T. (2019). Adapting usability techniques for application in open source Software: A multiple case study. *Information and Software Technology*, *107*. https://doi.org/10.1016/ j.infsof.2018.10.011

Llerena, L., Rodriguez, N., Gomez-Abajo, P., Castro, J. W., & Acuña, S. T. (2018). "Adoption of the visual brainstorming technique in the open source software development process." *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings*, 232–233. New York, NY, USA: ACM. https://doi.org/10.1145/3183440.3194946

Llerena, L., Rodriguez, N., Llerena, M., Castro, J. W., & Acuña, S. T. (2018). Applying a Usability Technique in the Open Source Software Development Process: Experiences from the Trenches. In *Trends in E-learning*: *Vol. i* (p. 13). InTech. https://doi.org/10.5772/intechopen.74862

Locke, L., Spirduso, W., & Silverman, S. (2013). *Proposals That Work: A Guide for Planning Dissertations and Grant Proposals* (6th ed.).

Long, Y., & Siau, K. (2009). Social Network Structures in Open Source Software Development Teams. In *Human Computer Interaction* (pp. 645–658). IGI Global. https://doi.org/10.4018/978-1-87828-991-9.ch043

López, L., Costal, D., Ralyté, J., Franch, X., Méndez, L., & Annosi, M. C. (2016). OSSAP - A situational method for defining open source software adoption processes. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9694*, 524–539. https://doi.org/10.1007/978-3-319-39696-5_32

Lund, A. M. (2001). Measuring usability with the USE questionnaire. *Usability Interface*, *8*(2), 3–6.

Luojus, S. (2010). From a momentary experience to a lasting one: The concept of and research on expanded user experience of mobile devices. *Acta Universitatis Ouluensis*. Retrieved from http://herkules.oulu.fi/isbn 9789514262463/ isbn9789514262463.pdf

Luther, L., Tiberius, V., & Brem, A. (2020). User experience (UX) in business, management, and psychology: A bibliometric mapping of the current state of research. *Multimodal Technologies and Interaction*, *4*(2). https://doi.org/10.3390/mti4020018

Lysne, O. (2018). Software Quality and Quality Management. In *The Huawei and Snowden Questions* (pp. 87–98). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-74950-1_10

Lyytinen, K. (1987). Different perspectives on information systems: problems and solutions. *ACM Computing Surveys*, *19*(1), 5–46. https://doi.org/10.1145/28865.28867

MacDonald, C. M. (2019). User experience (UX) capacity-building: A conceptual model and research Agenda. *DIS 2019 - Proceedings of the 2019 ACM Designing Interactive Systems Conference*, 187–200. https://doi.org/10.1145/3322276.3322346

MacDonald, C. M., Sosebee, J., & Srp, A. (2022). A Framework for Assessing Organizational User Experience (UX) Capacity. *International Journal of Human–Computer Interaction*, *38*(11), 1064–1080. https://doi.org/10.1080/10447318.2021.1979811

Magee, S., & Thiele, D. (2004). Engineering process standards: state of the art and challenges. *IT Professional*, *6*(5), 38–44. https://doi.org/10.1109/MITP.2004.60

Magin, D. P., Maier, A., & Hess, S. (2015). *Measuring Negative User Experience*. https://doi.org/10.1007/978-3-319-20898-5_10

Mahlke, S. (2005). Understanding users' experience of interaction. *Communications of the ACM*, *2006*(1990), 251–254. Retrieved from http://portal.acm.org/citation. cfm?id=1124702

Mahlke, S. (2008). Visual aesthetics and the user experience in Human-Computer Interaction. *Dagstuhl Seminar Proceedings*, (08292), 1–6. Retrieved from http://drops.dagstuhl.de/opus/volltexte/2008/1624

Mahlke, S., & Thüring, M. (2007). Studying antecedents of emotional experiences in interactive contexts. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 915–918. New York, NY, USA: ACM. https://doi.org/10.1145/1240624.1240762

Majid, R. A., Noor, N. L. M., Adnan, W. A. W., & Mansor, S. (2010). A survey on user involvement in software Development Life Cycle from practitioner's perspectives. *5th International Conference on Computer Sciences and Convergence Information Technology*, 240–243. IEEE. https://doi.org/10.1109/ICCIT.2010.5711064

Maki-Asiala, P., & Matinlassi, M. (2006). Quality Assurance of Open Source Components: Integrator Point of View. *30th Annual International Computer Software and Applications Conference (COMPSAC'06)*, 189–194. IEEE. https://doi.org/10.1109/COMPSAC.2006.153

Manakandan, S. K., Ismai, R., Jamil, M. R. M., & Ragunath, P. (2017). Pesticide applicators questionnaire content validation: A fuzzy delphi method. *Medical Journal of Malaysia*, *72*(4), 228–235.

Mandal, S., Kandar, S., & Ray, P. (2011). Open Incremental Model A Open Source Software Development Life Cycle Model "OSDLC." *International Journal of Computer Applications*, *21*(1), 33–39. https://doi.org/10.5120/2473-3327

Marangunić, N., & Granić, A. (2015). Technology acceptance model: a literature review from 1986 to 2013. *Universal Access in the Information Society*, *14*(1), 81–95. https://doi.org/10.1007/s10209-014-0348-1

Marcilly, R., Ammenwerth, E., Roehrer, E., Pelayo, S., Vasseur, F., & Beuscart-Zéphir, M.-C. (2015). Usability Flaws in Medication Alerting Systems: Impact on Usage and Work System. *Yearbook of Medical Informatics*, *24*(01), 55–67. https://doi.org/10.15265/IY-2015-006

Margarido, I. L., Fari, J. P., Vieira, M., & Vidal, R. M. (2013). Challenges in Implementing CMMI® High Maturity: Lessons Learned and Recommendations. *Software Quality Professional*, *16*(1), 4–22. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=aps&AN=95033301&amp%0Alang=pt-pt&site=eds-live&authtype=sso

Marois, P., Marsan, J., Carillo, K., Stol, K.-J., & Fitzgerald, B. (2022). A Delphi Study of Obsolete Assumptions in Free/Libre and Open Source Software. *European Conference on Information Systems (ECIS)*. Retrieved from https://aisel.aisnet.org/ecis2022_rp/149/

Marques, L., Matsubara, P. G., Nakamura, W. T., Ferreira, B. M., Wiese, I. S., Gadelha, B. F., … Conte, T. U. (2021). Understanding UX Better: A New Technique to Go beyond Emotion Assessment. *Sensors*, *21*(21), 7183. https://doi.org/10.3390/s21217183

Marsden, N., & Holtzblatt, K. (2018). How Do HCI Professionals Perceive Their Work Experience? *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, *2018-April*, 1–6. New York, NY, USA: ACM. https://doi.org/10.1145/3170427.3188501

Mashapa, J. (2013). *A Model for Managing User Experience*. Doctoral thesis, Nelson Mandela Metropolitan University.

Mashapa, Job, Chelule, E., Van Greunen, D., & Veldsman, A. (2013). Managing user experience - Managing change. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8118 LNCS*(PART 2), 660–677. https://doi.org/10.1007/978-3-642-40480-1_46

Masson, A. L. L., Amstutz, T., Lalanne, D., Lisowska Masson, A., Amstutz, T., Lalanne, D., … Lalanne, D. (2017). A Usability Refactoring Process for Large-Scale Open Source Projects. *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems, Part F1276*, 1135–1143. New York, NY, USA, NY, USA: ACM. https://doi.org/10.1145/ 3027063.3053345

Matook, S., & Indulska, M. (2009). Improving the quality of process reference models: A quality function deployment-based approach. *Decision Support Systems*, *47*(1), 60–71. https://doi.org/10.1016/j.dss.2008.12.006

McBride, T. (2010). Organisational theory perspective on process capability measurement scales. *Journal of Software Maintenance and Evolution: Research and Practice*, n/a-n/a. https://doi.org/10.1002/spip.440

McCarthy, J., & Wright, P. (2007). Technology as Experience. In *Technology as Experience*. The MIT Press. https://doi.org/10.7551/mitpress/6687.001.0001

McClure, D. (2007). Startup Metrics for Pirates : AARRR !!! (Startup Metrics for Product Marketing & Product Management). Retrieved May 4, 2022, from The Internet Revolution, Act III website: http://www.slideshare.net/ dmc500hats/ startup -metrics-for-pirates-long-version

McCorcle, M. D., & Bell, E. L. (1986). Case study research: Design and methods. *Evaluation and Program Planning*, *9*(4), 373–374. https://doi.org/10. 1016/ 0149 -7189(86)90052-2

McMillan, S. J., & Hwang, J.-S. (2002). Measures of Perceived Interactivity: An Exploration of the Role of Direction of Communication, User Control, and Time in Shaping Perceptions of Interactivity. *Journal of Advertising*, *31*(3), 29–42. https://doi.org/10.1080/00913367.2002.10673674

McNamara, N., & Kirakowski, J. (2006). Functionality, usability, and user experience. *Interactions*, *13*(6), 26–28. https://doi.org/10.1145/ 1167948. 1167972

Méhat, J., Malanga, K. N., Mehat, J., Ganchev, I., Wandeto, J., & Shivachi, C. (2015). Evaluation of Open Source Software with QualiPSO OMM : a case for Bungeni and AT4AM for All Evaluation of Open Source Software with QualiPSO OMM : a case for Bungeni and AT4AM for All. *Proceedings of free and open source software conference (FOSSC-15)*, (August), 41–46.

Menéndez-Caravaca, E., Bueno, S., & Gallego, M. D. (2021). Exploring the link between free and open source software and the collaborative economy: A Delphi-based scenario for the year 2025. *Technological Forecasting and Social Change*, *173*. https://doi.org/10.1016/j.techfore.2021.121087

Mettler, T. (2009). *A Design Science Research Perspective on Maturity Models in Information Systems*. *41*(0).

Mettler, T. (2010). Thinking in Terms of Design Decisions When Developing Maturity Models. *International Journal of Strategic Decision Sciences*, *1*(4), 76–87. https://doi.org/10.4018/jsds.2010100105

Meulman, J., & Heiser, W. (2013). *IBM SPSS Categories 21*.

Miles, D. A. (2017). Let’s Stop the Madness Part 2: Understanding the Difference Between Limitations vs. Delimitations. *5th Annual 2017 Black Doctoral Network Conference in Atlanta*, (August), 1–17.

Minge, M., Thüring, M., Wagner, I., & Kuhr, C. V. (2017). The meCUE questionnaire: A modular tool for measuring user experience. *Advances in Intelligent Systems and Computing*, *486*(January), 115–128. https://doi. org/10.1007/ 978-3-319-41685-4_11

Miron, W., & Muita, K. (2014). Cybersecurity Capability Maturity Models for Providers of Critical Infrastructure. *Technology Innovation Management Review*, *4*(10), 33–39. https://doi.org/10.22215/timreview837

Mitchell, G. (2006). Max Wertheimer and Gestalt theory. In *Journal of the History of the Behavioral Sciences*.

Mlekus, L., Bentler, D., Paruzel, A., Kato-Beiderwieden, A. L., & Maier, G. W. (2020). How to raise technology acceptance: user experience characteristics as technology-inherent determinants. *Gruppe. Interaktion. Organisation. Zeitschrift Fur Angewandte Organisationspsychologie*, *51*(3), 273–283. https://doi.org/10. 1007/s11612-020-00529-7

Mockus, A., Fielding, R. T., & Herbsleb, J. D. (2000). A case study of open source software development. *Proceedings of the 22nd International Conference on Software Engineering - ICSE '00*, *11*(3), 309–346. https://doi.org/10. 1145/ 337180.337209

Mockus, A., Fielding, R. T., & Herbsleb, J. D. (2002). Two case studies of open source software development. *ACM Transactions on Software Engineering and Methodology*, *11*(3), 309–346. https://doi.org/10.1145/567793.567795

Mohajan, H. K. (2017). Two criteria for good measurements in research: validity and reliability. *Annals of Spiru Haret University. Economic Series*, *17*(4), 59–82. https://doi.org/10.26458/1746

Mohamed Yusoff, A. F., Hashim, A., Muhamad, N., & Wan Hamat, W. N. (2021). Application of Fuzzy Delphi Technique Towards Designing and Developing the Elements for the e-PBM PI-Poli Module. *Asian Journal of University Education*, *17*(1), 292. https://doi.org/10.24191/ajue.v17i1.12625

Mohammed, I., & Bade, A. M. (2019). *Cybersecurity Capability Maturity Model for Network System*. *09*, 28637–28641.

Molich, R. (2021). UX maturity for UX Professionals. Retrieved from https://www.dialogdesign.dk/a-ux-maturity-scale/

Molich, R., & Winter, D. (2022). UX Integration: The Proverbial Stumbling Stones on the Road to UX Paradise. Retrieved February 16, 2022, from User Experience Magazine website: https://uxpamagazine.org/ux-integration-the-proverbial-stumbling-stones-on-the-road-to-ux-paradise/

Monteiro, E. L., & Maciel, R. S. P. (2020). Maturity Models Architecture: A large systematic mapping. *ISys - Brazilian Journal of Information Systems*, *13*(2), 110–140. https://doi.org/10.5753/isys.2020.761

Moon, M. D. (2019). Triangulation: A Method to Increase Validity, Reliability, and Legitimation in Clinical Research. *Journal of Emergency Nursing*, *45*(1), 103–105. https://doi.org/10.1016/j.jen.2018.11.004

Morville, P. (2004). User Experience Design. Retrieved August 19, 2022, from Semantic Studios website: http://semanticstudios.com/ publications/ semantics/ 000029.php

Moshagen, M., & Thielsch, M. T. (2010). Facets of visual aesthetics. *International Journal of Human Computer Studies*, *68*(10), 689–709. https://doi.org /10.1016/j. ijhcs.2010.05.006

Mtebe, J. S. (2019). Examining user experience of eLearning systems implemented in two universities in Tanzania. *Interactive Technology and Smart Education*, *17*(1), 39–55. https://doi.org/10.1108/ITSE-05-2019-0025

Murphy-Hill, E., Jaspan, C., Sadowski, C., Shepherd, D., Phillips, M., Winter, C., … Jorde, M. (2021). What Predicts Software Developers' Productivity? *IEEE Transactions on Software Engineering*, *47*(3), 582–594. https://doi.org/10.1109/ TSE.2019.2900308

Murray, T. J., Pipino, L. L., & Van Gigch, J. P. (1985). A pilot study of fuzzy set modification of delphi. *Human Systems Management*, *5*(1), 76–80. https:// doi.org/ 10.3233/HSM-1985-5111

Murry, J. W., & Hammons, J. O. (1995). Delphi: A Versatile Methodology for Conducting Qualitative Research. *The Review of Higher Education*, *18*(4), 423–436. https://doi.org/10.1353/rhe.1995.0008

Mustapha, R., Jafar, M. F., Mohamad, A., Mahmud, M., Malkan, S. N. A., Hassan, S. H., & Ibrahim, R. A. (2021). Development of e-xtvt guidelines in promoting active online learning in higher education: the fuzzy delphi approach. *Psychology and Education Journal*, *58*(1), 5681–5696. https://doi.org/10.17762/pae.v58i1.1972

Mutula, S., & Kalaote, T. (2010). Open source software deployment in the public sector: A review of Botswana and South Africa. *Library Hi Tech*, *28*(1), 63–80. https://doi.org/10.1108/07378831011026698

MWTC. (2016). Nictp. *National Information and Communications Technology Policy*, (May), 1–45.

Myers, B. A., & Rosson, M. B. (1992). Survey on user interface programming. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '92*, 195–202. New York, New York, USA: ACM Press. https://doi.org/10.1145/142750.142789

Nagle, F., & Hoffman, J. L. (2020). *Vulnerabilities in the Core Preliminary Report and Census II of Open Source Software*. Retrieved from www.linuxfoundation.org%7Cwww.coreinfrastructure.org%7C

Nagunwa, T., & Lwoga, E. (2012). Developing eLearning technologies to implement competency based medical education: Experiences from Muhimbili University of Health and Allied Sciences. *International Journal of Education and Development Using Information and Communication Technology*, *8*(3), 7–21. https://doi.org/1 0.31 09/01421 59X.2010.501 190 DOI: 10.1016/j.sbspro.2009.01.087

Nagy, D., Yassin, A. M., Bhattacherjee, A., Spinellis, D., & Giannikas, V. (2012). Organizational adoption of open source software. *Journal of Systems and Software*, *85*(3), 666–682. https://doi.org/10.1016/j.jss.2011.09.037

Nakano, D., & Muniz Jr., J. (2018). Writing the literature review for empirical papers. *Production*, *28*. https://doi.org/10.1590/0103-6513.20170086

Namayala, P., Kondo, T., Manyilizu, M., & Hassan, K. (2022). Practical applicability of user experience capability/maturity models in the development processes of free and open-source software: a systematic literature review. *Behaviour & Information Technology*, 1–28. https://doi.org/10. 1080/0144929X. 2022.203 2348

Napoleao, B. M., Petrillo, F., & Halle, S. (2020). Open Source Software Development Process: A Systematic Review. *2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC)*, 135–144. IEEE. https://doi.org/10.1109/EDOC49727.2020.00025

Neeman, P. (2012). Organizational Challenges for UX Professionals. Retrieved December 12, 2022, from UXmatters website: https://www.uxmatters.com/mt/ archives/2012/04/organizational-challenges-for-ux-professionals.php

Negus, C. (2015). GNU General Public License. In *Linux® Bible* (pp. 860–872). Wiley. https://doi.org/10.1002/9781119209539.oth1

Neilsen, J. (1995). 10 Usability Heuristics for User Interface Design. *Conference Companion on Human Factors in Computing Systems CHI 94*, 152–158. Retrieved from http://portal.acm.org/citation.cfm?doid= 259963.260333 %5Cnhttp: //www.nngroup.com/articles/ten-usability-heuristics/

Newman, S. C. (2003). Sample Size and Power. In *Biostatistical Methods in Epidemiology* (pp. 281–294). Hoboken, NJ, USA: John Wiley & Sons, Inc. https://doi.org/10.1002/0471272612.ch14

Nguyen-Duc, A., Kemell, K. K., & Abrahamsson, P. (2021). The entrepreneurial logic of startup software development: A study of 40 software startups. *Empirical Software Engineering*, *26*(5). https://doi.org/10.1007/s10664-021-09987-z

Nguyen, T. (2020). *Improving Students' UX in Online Learning Platform*. Bachelor thesis, lab university of applied sciences

Nichols, D. M., & Twidale, M. B. (2005). The usability of open source software. *First Monday*, *2*(SPEC).

Nichols, D. M., & Twidale, M. B. (2006). Usability processes in open source projects. *Software Process: Improvement and Practice*, *11*(2), 149–162. https://doi.org/10.1002/spip.256

Nichols, D., & Twidale, M. (2003). The Usability of Open Source Software. *First Monday*, *8*(1), 21. https://doi.org/10.5210/fm.v8i1.1018

Nicolini, D. (2011). Practice as the Site of Knowing: Insights from the Field of Telemedicine. *Organization Science*, *22*(3), 602–620. https://doi.org/10.1287/orsc.1100.0556

Nielsen, J. (1999). Trust or Bust: Communicating Trustworthiness in Web Design. Retrieved from Jakob Nielsen's Alertbox website: http://www.nngroup.com/articles/trust-or-bust-communicating-trustworthiness-in-web-design/

Nielsen, J. (2000). Why You Only Need to Test with 5 Users. https://doi.org/http://www.useit.com/alertbox/20000319.html

Nielsen, J. (2006). Corporate UX Maturity: Stages 5-8.

Nielsen, J. (2010). PART 1 Defining Usability. In *Usability Engineering* (pp. 1–22). Retrieved from https://booksite.elsevier.com/ samplechapters/9780123751140/02~Chapter_1.pdf

Nielsen, J. (2012). Usability 101: Introduction to Usability. Retrieved from All Usability website: https://www.nngroup.com/ articles/usability-101-introduction-to-usability/

Nielsen, J., & Norman, D. (2017). Nielsen Norman Group The Definition of User Experience Author. Retrieved December 31, 2020, from https://www. nngroup.com/ articles/definition-user-experience

Noor, A. Z. M., Zainuddin, N. M., Ibrahim, R., & ... (2020). Evaluating The User Experience of A Muslimah Fashion E-Commerce Website Using Heuristic Evaluation Method. *... of Informatics (OIJI)*, *8*(2), 1–17. Retrieved from http://apps.razak.utm.my/ojs/index.php/oiji/article/view/250

Norman, D. (2016). The Design of Everyday Things. In *The Design of Everyday Things*. Vahlen. https://doi.org/10.15358/9783800648108

Norman, D. A. (2013). The design of everyday things: Revised and expanded edition. In *Requirements Engineering*. The MIT Press.

Novák, J. Š., Masner, J., Benda, P., Šimek, P., & Merunka, V. (2023). Eye Tracking, Usability, and User Experience: A Systematic Review. *International Journal of Human–Computer Interaction*, 1–17. https://doi.org/10.1080/ 10447318. 2023.2221600

Nunnally, J. C. J. C., Bernstein, I. H. H., Hundleby, J. D., & Nunnally, J. C. J. C. (1994). Psychometric Theory. In *Encyclopedia of Behavioral Medicine* (3rd ed, Vol. 5). New York: McGraw-Hill. https://doi.org/10.2307/1161962

O'Brien, H. (2016). Theoretical Perspectives on User Engagement. In *Why Engagement Matters* (pp. 1–26). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-27446-1_1

O'Leary, Z. (2017). The Essential Guide to Doing Your Research Project. In *SAGE Publications* (3rd Editio). London ; Thousand Oaks, California: Sage Publications Ltd.

Obrist, M., Roto, V., Law, E. L. C., Väänänen-Vainio-Mattila, K., Vermeeren, A., & Buie, E. (2012). Theories behind UX research and how they are used in practice. *Conference on Human Factors in Computing Systems - Proceedings*, (May), 2751–2754. New York, NY, USA: ACM. https://doi.org/10.1145/2212776.2212712

Obrist, M., Roto, V., & Väänänen-Vainio-Mattila, K. (2009). User experience evaluation. *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, 2763–2766. New York, NY, USA: ACM. https://doi.org/10.1145/1520340.1520401

Office of Government Commerce - OGC. (2010). *Portfolio, Programme and Project Management Maturity Model*. 1–6. Retrieved from http://www.p3m3-officialsite.com/P3M3Model/Model_mhtry.aspx

Okoli, C. (2021). Developing Novel and Relevant Theoretical Contributions with Literature Reviews. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3774487

Okoli, C., & Pawlowski, S. D. (2004). The Delphi method as a research tool: An example, design considerations and applications. *Information and Management*, *42*(1), 15–29. https://doi.org/10.1016/j.im.2003.11.002

Okręglicka, M., Mynarzová, M., & Kaňa, R. (2015). Business process maturity in small and medium-sized enterprises. *Polish Journal of Management Studies*, *12*(1), 121–131.

Oosterhof, A. (2001). *Classroom Applications of Educational Measurement. Third Edition* (3rd Editio). Retrieved from http://proxy.library.adelaide.edu.au/login?url=http://search.proquest.com/docview/62351917?accountid=8203

Open Source Initiative. (2010). The Open Source Definition - Open Source Initiative. Retrieved from The Open Source Definition website: https://opensource.org/osd

OpenBRR.org. (2005). Business Readiness Rating for Open Source. *Access*, 1–22.

Oreku, G. S., & Mtenzi, F. J. (2013). Adoption and diffusion of open source software in Tanzania: A way forward. *2013 IST-Africa Conference and Exhibition, IST-Africa 2013*, 1–10. Nairobi, Kenya: IEEE.

Osborne, J. W., & Overbay, A. (2004). The power of outliers (and why researchers should ALWAYS check for them). *Practical Assessment, Research and Evaluation*, *9*(6).

Osinusi, K. (2020). Make It Count – A Guide to Measuring the User Experience. Retrieved May 2, 2022, from Https://Www.Toptal.Com/ website: https://www.toptal.com/designers/ux/measuring-the-user-experience

Otchere, W. (2022). How your organisation can prepare for Usability Testing. Retrieved February 6, 2022, from Lion & Mason Ltd website: https://www. lionandmason. com/ux-blog/how-your-organisation-can-prepare-for-usability-testing/

Othman, M. K., Petrie, H., & Power, C. (2011). Engaging Visitors in Museums with Technology: Scales for the Measurement of Visitor and Multimedia Guide Experience. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*: *Vol. 6949 LNCS* (pp. 92–99). https://doi.org/10.1007/978-3-642-23768-3_8

Otto, L., Bley, K., & Harst, L. (2020). Designing and evaluating prescriptive maturity models: A design science-oriented approach. *Proceedings - 2020 IEEE 22nd Conference on Business Informatics, CBI 2020*, *2*(100310385), 40–47. https://doi.org/10.1109/CBI49978.2020.10056

Ouhbi, S., Idri, A., Aleman, J. L. F., & Toval, A. (2014). Evaluating Software Product Quality: A Systematic Mapping Study. *2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement*, 141–151. IEEE. https://doi.org/10.1109/IWSM.Mensura.2014.30

Ouhbi, S., Idri, A., Fernández-Alemán, J. L., & Toval, A. (2015). Predicting software product quality: A systematic mapping study. *Computacion y Sistemas*, *19*(3), 547–562. https://doi.org/10.13053/CyS-19-3-1960

Ovad, T., & Larsen, L. B. (2015). The Prevalence of UX Design in Agile Development Processes in Industry. *Proceedings - 2015 Agile Conference, Agile 2015*, 40–49. IEEE. https://doi.org/10.1109/Agile.2015.13

P. Miguel, J., Mauricio, D., & Rodríguez, G. (2014). A Review of Software Quality Models for the Evaluation of Software Products. *International Journal of Software Engineering & Applications*, *5*(6), 31–53. https://doi.org/10.5121/ijsea.2014.5603

Paas, F., & Ayres, P. (2014). Cognitive Load Theory: A Broader View on the Role of Memory in Learning and Education. *Educational Psychology Review*, *26*(2), 191–195. https://doi.org/10.1007/s10648-014-9263-5

Palmius, J. (2007). Criteria for measuring and comparing information systems. *Proceedings of the 30th Information Systems Research Seminar in Scandinavia IRIS*, 1–24. Retrieved from http://www.palmius.com/joel/text/IRIS-30-final.pdf

Pankin, J. (2013). Schema Theory Basic Concepts. *Web.Mit*, 1–5. Retrieved from http://web.mit.edu/pankin/www/Schema_Theory_and_Concept_Formation.pdf

Patino-Toro, O. N., Valencia-Arias, A., Gomez-Molina, S., & Bermeo-Giraldo, M. C. (2022). Open-Source Software Adoption Among University Students in Emerging Countries. *IEEE Revista Iberoamericana de Tecnologias Del Aprendizaje*, *17*(2), 185–196. https://doi.org/10.1109/RITA.2022.3166950

Paulk, M.C., Curtis, B., Chrissis, M. B., & Weber, C. V. (1993). Capability maturity model, version 1.1. *IEEE Software*, *10*(4), 18–27. https://doi.org/10.1109/52.219617

Paulk, Mark C., Weber, C. V., Curtis, B., & Chrissis, M. B. (1996). The capability maturity model: Guidelines for improving the software process. In *Computers & Mathematics with Applications* (1st ed., Vol. 31). Addison-Wesley Professional. https://doi.org/10.1016/s0898-1221(96)90461-7

Paulk, Mark C, Curtis, B., Chrissis, M. B., & Weber, C. V. (2011). The capability maturity model for software. *Software Process Improvement*, 49–52. https://doi.org/10.1109/9781118156667.ch2

Păunescu, C., & Acatrinei, C. (2012). Managing maturity in process-based improvement organizations: A perspective of the Romanian companies. *Journal of Business Economics and Management*, *13*(2), 223–241. https://doi.org/ 10.3846/ 16111699.2011.620149

Pavliscak, P. (2014). Choosing the Right Metrics for User Experience. Retrieved from UX Matters website: http://www.uxmatters.com/mt/archives/ 2014/06/choosing-the-right-metrics-for-user-experience.php

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753 /MIS0742-1222240302

Peres, A. L., Da Silva, T. S., Silva, F. S., Soares, F. F., De Carvalho, C. R. M., & De Lemos Meira, S. R. (2014). AGILEUX model: Towards a reference model on integrating UX in developing software using agile methodologies. *Proceedings - 2014 Agile Conference, AGILE 2014*, 61–63. IEEE. https://doi.org/10.1109/ AGILE.2014.15

Peres, A. L., & Meira, S. L. (2015). Towards a framework that promotes integration between the UX design and SCRUM, Aligned to CMMI. *2015 10th Iberian Conference on Information Systems and Technologies, CISTI 2015*, 1–4. IEEE. https://doi.org/10.1109/CISTI.2015.7170443

Pernice, K., Gibbons, S., Moran, K., & Whitenton, K. (2021). The 6 Levels of UX Maturity. Retrieved from Nielsen Norman Group website: https://www .nngroup. com/articles/ux-maturity-model/%0Ahttp://files/70/ux-maturity-model.html

Peters, K., & Halcomb, E. (2015). Interviews in qualitative research. *Nurse Researcher*, *22*(4), 6–7. https://doi.org/10.7748/nr.22.4.6.s2

Petersen, K., Badampudi, D., Shah, S. M. A., Wnuk, K., Gorschek, T., Papatheocharous, E., … Cicchetti, A. (2018). Choosing Component Origins for Software Intensive Systems: In-House, COTS, OSS or Outsourcing? - A Case Survey. *IEEE Transactions on Software Engineering*, *44*(3), 237–261. https://doi.org/10.1109/TSE.2017.2677909

Petrinja, E., & Succi, G. (2012). Assessing the Open Source Development Processes Using OMM. *Advances in Software Engineering*, *2012*, 1–17. https://doi.org/10. 1155/2012/235392

Petrov, D., & Obwegeser, N. (2018). Adoption Barriers of Open-Source Software: A Systematic Review. *Proceedings of the 27th International Conference on Information Systems Development: Designing Digitalization, ISD 2018*, (March). Retrieved from https://ssrn.com/abstract=3138085

Pettersson, I., Lachner, F., Frison, A.-K., Riener, A., & Butz, A. (2018). A Bermuda Triangle? *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, *2018-April*, 1–16. New York, NY, USA: ACM. https://doi.org/10.1145/3173574.3174035

Pezer, M., Lazic, N., & Odak, M. (2017). Free and Open Source Software in the secondary education in Bosnia and Herzegovina. *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 882–886. IEEE. https://doi.org/10.23919/ MIPRO.2017.7973546

Philipps, A., & Mrowczynski, R. (2021). Getting more out of interviews. Understanding interviewees' accounts in relation to their frames of orientation. *Qualitative Research*, *21*(1), 59–75. https://doi.org/10.1177/1468794119867548

Pickett, P. (2019). How Open-Source Software Works. Retrieved June 18, 2020, from https://www. thebalancecareers.com/what-is-open-source-software-2071941

Pigosso, D. C. ., Rozenfeld, H., & McAloone, T. C. (2013). Ecodesign maturity model: a management framework to support ecodesign implementation into manufacturing companies. *Journal of Cleaner Production*, *59,* 160–173. https://doi.org/10.1016/j.jclepro.2013.06.040

Pinto, G., Steinmacher, I., Dias, L. F., & Gerosa, M. (2018). On the challenges of open-sourcing proprietary software projects. *Empirical Software Engineering*, *23*(6), 3221–3247. https://doi.org/10.1007/s10664-018-9609-6

Poizot, E., & Méar, Y. (2010). Using a GIS to enhance grain size trend analysis. *Environmental Modelling and Software*, *25*(4), 513–525. https://doi.org/ 10.1016/j. envsoft.2009.10.002

Polit, D. F., & Beck, C. T. (2017). *Nursing Research Generating and Assessing Evidence for Nursing Practice* (10th Editi). LWW.

Polson, P. G., Lewis, C., Rieman, J., & Wharton, C. (1992). Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *International Journal of Man-Machine Studies*, *36*(5), 741–773. https://doi.org/10.1016/0020-7373(92) 90039-N

Pöppelbuß, J., & Röglinger, M. (2011). What makes a useful maturity model? A framework of general design principles for maturity models and its demonstration in business process management. *19th European Conference on Information Systems, ECIS 2011*, *4801*.

Prasad, D., & Reddy, C. S. (2013). Understanding the Differences between Proprietary & Free and Open Source Software. *International Journal o Engine*, *3*(4), 2295–2299.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., & Carey, T. (1994). *Human-Computer Interaction: Concepts And Design* (1st Editio). Addison Wesley.

Pressman, R. S., & Maxim, B. R. (2014). *Software Engineering: A Practitioner's Approach 8th Edition* (8th Editio). New York.: McGraw Hill.

Pretorius, M., Hobbs, J., & Fenn, T. (2015). The User Experience Landscape of South Africa. *ACM International Conference Proceeding Series, 28-30-Sept*, 1–9. New York, New York, USA: ACM Press. https://doi.org/10.1145/2815782.2815807

Price, J. H., & Murnan, J. (2004). Research Limitations and the Necessity of Reporting Them. *American Journal of Health Education*, *35*(2), 66–67. https://doi.org/10.1080/19325037.2004.10603611

Price, J. L., de Ven, A. H. Van, & Ferry, D. L. (1981). Measuring and Assessing Organizations. *Administrative Science Quarterly*, *26*(2), 324. https://doi.org/10.2307/2392488

Proença, D., Vieira, R., Antunes, G., Da Silva, M. M., Borbinha, J., Becker, C., & Kulovits, H. (2013). Evaluating a process for developing a capability maturity model. *Proceedings of the ACM Symposium on Applied Computing*, 1474–1475. https://doi.org/10.1145/2480362.2480637

Project Management Institute. (2008). Organizational project management maturity model (OPM3) : knowledge foundation. *Organizational Project Management Maturity Model*, 186.

Provost, G., & Robert, J.-M. (2013). The Dimensions of Positive and Negative User Experiences with Interactive Products. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*: *Vol. 8012 LNCS* (pp. 399–408). https://doi.org/10.1007/978-3-642-39229-0_43

Pudari, R., & Ernst, N. A. (2023). *From Copilot to Pilot: Towards AI Supported Software Development*. Retrieved from http://arxiv.org/abs/2303.04142

QualiPSo. (2007). QualiPSo Project (Quality Platform for Open Source Software). Retrieved from https://cordis.europa.eu/project/id/034763

Quiñones, D., & Rusu, C. (2017). How to develop usability heuristics: A systematic literature review. *Computer Standards & Interfaces*, *53*, 89–122. https://doi.org/10.1016/j.csi.2017.03.009

Raffoul, E., Domínguez, K., Pérez, M., Mendoza, L. E., & Grimán, A. C. (2008). Quality model for the selection of floss-based issue tracking system. *Proceedings of the IASTED International Conference on Software Engineering, SE 2008*, 43–49.

Raja, U., & Barry, E. (2005). Investigating quality in large-scale Open Source Software. *Proceedings of the Fifth Workshop on Open Source Software Engineering - 5-WOSSE*, 1–4. New York, New York, USA: ACM Press. https://doi.org/10.1145/1083258.1083268

Rajanen, M. (2023). Open Source Usability and User Experience. *Computer, 56*(2), 106–110. https://doi.org/10.1109/MC.2022.3219634

Rajanen, M., & Iivari, N. (2013). Open source and human computer interaction philosophies in open source projects - Incompatible or Co-existent? *Proceedings of the 17th International Academic MindTrek Conference: "Making Sense of Converging Media", MindTrek 2013*, 67–74. New York, New York, USA: ACM Press. https://doi.org/10.1145/2523429.2523463

Rajanen, M., & Iivari, N. (2015a). Examining Usability Work and Culture in OSS. *IFIP Advances in Information and Communication Technology*, 58–67. https://doi.org/10.1007/978-3-319-17837-0_6

Rajanen, M., & Iivari, N. (2015b). Power, Empowerment and Open Source Usability. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, *2015-April*, 3413–3422. New York, NY, USA: ACM. https://doi.org/10.1145/2702123.2702441

Rastogi, V. (2015). Software Development Life Cycle Models- Comparison , Consequences. *International Journal of Computer Science and Information Technologies*, *6*(1), 168–172.

Ratan, S., Anand, T., & Ratan, J. (2019). Formulation of research question – Stepwise approach. *Journal of Indian Association of Pediatric Surgeons*, *24*(1), 15. https://doi.org/10.4103/jiaps.JIAPS_76_18

Raymond, E. (2005). The cathedral and the bazaar. In *First Monday* (Vol. 2). O'Reilly Media, Inc. https://doi.org/10.1007/s12130-999-1026-0

Raza, A., & Capretz, L. F. (2010). Contributors Preference in Open Source Software Usability: An Empirical Study. *International Journal of Software Engineering & Applications*, *1*(2), 45–64. https://doi.org/10.5121/ijsea.2010.1204

Raza, A., Capretz, L. F., & Ahmed, F. (2010). Improvement of Open Source Software Usability: An Empirical Evaluation from Developers' Perspective. *Advances in Software Engineering*, *2010*, 1–12. https://doi.org/10. 1155/2010/517532

Raza, A., Capretz, L. F., & Ahmed, F. (2012a). An open source usability maturity model (OS-UMM). *Computers in Human Behavior*, *28*(4), 1109–1121. https://doi.org/10.1016/j.chb.2012.01.018

Raza, A., Capretz, L. F., & Ahmed, F. (2012b). Users' perception of open source usability: An empirical study. *Engineering with Computers*, *28*(2), 109–121. https://doi.org/10.1007/s00366-011-0222-1

Raza, A., Capretz, L. F., & Ahmed, F. (2013). An empirical study of open source software usability: The industrial perspective. *Open Source Software Dynamics, Processes, and Applications*, 1–17. https://doi.org/10.4018/978-1-4666-2937-0.ch001

Riehle, D., & Harutyunyan, N. (2019). Open-Source License Compliance in Software Supply Chains. In *Towards Engineering Free/Libre Open Source Software (FLOSS) Ecosystems for Impact and Sustainability* (pp. 83–95). Singapore: Springer Singapore. https://doi.org/10.1007/978-981-13-7099-1_5

Rifkin, M. D. (2002). Introduction to Literature Reviews. https://doi.org /10.1097/00013644-200209000-00006

Rivero, L. ., & Conte, T. (2017). A Systematic Mapping Study on Research Contributions on UX Evaluation Technologies. *IHC 2017: Proceedings of the XVI Brazilian Symposium on Human Factors in Computing Systems*, 1–10. Joinville, Brazil. https://doi.org/10.1145/3160504.3160512

Robbio, A. (2019). User Experience Is Now Your Business Strategy. Retrieved January 2, 2021, from Forbes Technology Council website: https://www.forbes.com/sites/forbestechcouncil/2019/10/15/user-experience-is-now-your-business-strategy/?sh=3e428fe982f6

Robert, B. J., & Larry, C. B. (2019). *Educational Research Quantitative, Qualitative, and Mixed Approaches* (7th ed.). SAGE Publications.

Robert, J.-M., & Lesage, A. (2017a). Designing and Evaluating User Experience. In *The Handbook of Human-Machine Interaction* (pp. 321–338). CRC Press. https://doi.org/10.1201/9781315557380-16

Robert, J.-M., & Lesage, A. (2017b). From Usability to User Experience with Interactive Systems. In G. A. Boy (Ed.), *The Handbook of Human-Machine Interaction* (pp. 303–320). Ashgate, U.K: CRC Press. https://doi.org/10.1201/9781315557380-15

Robert, J. M., & Larouche, A. (2012). The dimensions of user experience with interactivesystems. *Proceedings of the IADIS International Conference Interfaces and Human Computer Interaction 2012, IHCI 2012, Proceedings of the IADIS International Conference Game and Entertainment Technologies 2012*, 89–96.

Roberts, C. W. (2015). Content Analysis. In *International Encyclopedia of the Social & Behavioral Sciences: Second Edition* (Vol. 2, pp. 769–773). https://doi.org/10.1016/B978-0-08-097086-8.44010-9

Robles, G., Steinmacher, I., Adams, P., & Treude, C. (2019). Twenty Years of Open Source Software: From Skepticism to Mainstream. *IEEE Software*, *36*(6), 12–15. https://doi.org/10.1109/MS.2019.2933672

Rock, I., & Palmer, S. (1990). The Legacy of Gestalt Psychology. *Scientific American*, *263*(6), 84–90. https://doi.org/10.1038/scientificamerican1290-84

Rodden, K., Hutchinson, H., & Fu, X. (2010). Measuring the user experience on a large scale. *Proceedings of the 28th International Conference on Human Factors in Computing Systems - CHI '10*, *4*, 2395. New York, New York, USA: ACM Press. https://doi.org/10.1145/1753326.1753687

Roets, R. (University of F. H., Minnaar, M. (University of F. H., & Wright, K. (University of F. H. (2007). Open Source : Towards Successful Systems Open Source : Towards Successful Systems. *Proceedings of the 9 Th International Conference on Social Implications of Computers in Developing Countries, São Paulo, Brazil, May 2007*, (May 2007), 1–19. São Paulo, Brazil. Retrieved from http://www.ifipwg94.org.br/fullpapers/R0061-1.pdf

Rohles, B. (2021). Selected UX maturity models. Retrieved April 28, 2022, from https://rohles.net/en/articles/ux-maturity-models

Roopa, S., & Rani, M. (2012). Questionnaire Designing for a Survey. *The Journal of Indian Orthodontic Society*, *46*, 273–277. https://doi.org/10.5005/jp-journals-10021-1104

Rose, E. J., Putnam, C., & MacDonald, C. M. (2020). Preparing Future UX Professionals. *Proceedings of the 38th ACM International Conference on Design of Communication*, 1–8. New York, NY, USA: ACM. https://doi.org/10.1145/3380851.3416774

Rose, S., Spinks, N., & Canhoto, A. I. (2014). Management research: Applying the principles. *Management Research: Applying the Principles*, 1–419.

Roto, V. (2007). User experience from product creation perspective. *Towards a UX Manifesto Workshop*, (Roto), 1–4. Retrieved from https://research.nokia.com/files/UXmanifesto-Roto.pdf

Roto, V., Law, E., Vermeeren, A., & Hoonhout, J. (2010). Bringing clarity to the concept of user experience. In *User Experience White Paper* (p. 12). Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:USER+EXPERIENCE+WHITE+PAPER#0%5Cnhttp://www.allaboutux.org/uxwhitepaper

Roto, V., & Rautava, M. (2008). User Experience Elements and Brand Promise. *Proceedings of the International Engagability & Design Conference (Idec4'08)*.

Roulston, K. (2014). Analysing Interviews. In *The SAGE Handbook of Qualitative Data Analysis* (pp. 297–312). 1 Oliver's Yard, 55 City Road London EC1Y 1SP: SAGE Publications, Inc. https://doi.org/10.4135/9781446282243.n20

Rout, T., O'Connor, R. V., & Dorling, A. (2015). *Software Process Improvement and Capability Determination*. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-19860-6

Rowley, J. (2014). Designing and using research questionnaires. *Management Research Review*, *37*(3), 308–330. https://doi.org/10.1108/MRR-02-2013-0027

Rukonić, L., Kervyn de Meerendré, V., & Kieffer, S. (2019). Measuring UX Capability and Maturity in Organizations. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*: *Vol. 11586 LNCS* (pp. 346–365). https://doi.org/10.1007/978-3-030-23535-2_26

Rumelhart, D. E. (1984). Schemata and Cognitive Systems. *Handbook of Social Cognition*, *1*, 161–188. Retrieved from http://ci.nii.ac.jp/naid/10006841329/en/

Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, *39*(6), 1161–1178. https://doi.org/10.1037/h0077714

Russell, J. A. (2003). Core affect and the psychological construction of emotion. *Psychological Review*, *110*(1), 145–172. https://doi.org/10.1037/0033-295X.110.1.145

Russell, J. A., & Barrett, L. F. (1999). Core affect, prototypical emotional episodes, and other things called emotion: Dissecting the elephant. *Journal of Personality and Social Psychology*, *76*(5), 805–819. https://doi.org/10.1037/0022-3514.76.5.805

Saad, J., Martinelli, S., Machado, L. S., de Souza, C. R. B., Alvaro, A., & Zaina, L. (2021). UX work in software startups: A thematic analysis of the literature. *Information and Software Technology*, *140*, 106688. https://doi.org/10.1016/j.infsof.2021.106688

Safder, A., & Yousaf, S. (2018). Influence of ISO 9001 certification on project management performance in software industry. *European Online Journal of Natural and Social Sciences*, *7*(3), 19–29.

Saffie, N. A. M., Shukor, N. M., & Rasmani, K. A. (2016). Fuzzy delphi method: Issues and challenges. *2016 International Conference on Logistics, Informatics and Service Sciences (LISS)*, 1–7. IEEE. https://doi.org/10. 1109/LISS. 2016.7854490

Sagar, K., & Gupta, D. (2018). Current Trends and Challenges in User Experience Evaluation and System Development. *International Journal of Pure and Applied Mathematics*, *118*(20), 449–453.

Saini, M., & Kaur, K. (2014). A review of open source software development life cycle models. *International Journal of Software Engineering and Its Applications*, *8*(3), 417–434. https://doi.org/10.14257/ijseia.2014.8.3.38

Sakhardande, P., & Thanawala, R. (2014). UX Maturity Model: From Usable to Delightful. User Experience Magazine. Retrieved from User Experience, The magazine of the user Experience Professionals Association website: http://uxpamagazine.org/ux-maturity-model/

Salah, D., Paige, R., & Cairns, P. (2014). Integrating agile development processes and user centred design - a place for usability maturity models? *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8742*, 108–125. https://doi.org/10.1007 /978-3-662-44811-3_7

Saleh, A., & Bista, K. (2017). Examining Factors Impacting Online Survey Response Rates in Educational Research: Perceptions of Graduate Students. *Journal of MultiDisciplinary Evaluation*, *13*(29), 63–74.

Salkind, N. (2010). *Encyclopedia of Research Design*. 2455 Teller Road, Thousand Oaks California 91320 United States: SAGE Publications, Inc. https://doi. org/10. 4135/9781412961288

Salviano, C. F., & Figueiredo, A. M. C. M. (2008). Unified basic concepts for Process Capability Models. *20th International Conference on Software Engineering and Knowledge Engineering, SEKE 2008*, 173–178.

Samoladas, I., Gousios, G., Spinellis, D., & Stamelos, I. (2008). The SQO-OSS Quality Model: Measurement Based Open Source Software Evaluation. In *IFIP International Federation for Information Processing* (Vol. 275, pp. 237–248). https://doi.org/10.1007/978-0-387-09684-1_19

Sanchez, L., Delgado, A., Ruiz, F., Garcia, F., & Piattini, M. (2009). Measurement and Maturity of Business Processes. In *Handbook of Research on Business Process Modeling* (pp. 532–556). IGI Global. https://doi.org/10.4018/978-1-60566-288-6.ch024

Sanchez, V. R., Ayuso, P. N., Galindo, J. A., & Benavides, D. (2020). Open Source Adoption Factors—A Systematic Literature Review. *IEEE Access*, *8*, 94594–94609. https://doi.org/10.1109/ACCESS.2020.2993248

Sanders, M. (2011). Software Process Improvement for Small Organizations. In *Software Process Improvement* (pp. 475–489). IEEE. https://doi.org/10.1109/9781118156667.ch10

Sangaiah, A. K., & Thangavelu, A. K. (2013). An exploration of FMCDM approach for evaluating the outcome/success of GSD projects. *Central European Journal of Engineering*, *3*(3), 419–435. https://doi.org/10.2478/s13531-012-0070-9

Santoso, H. B., Schrepp, M., Yugo Kartono Isal, R., Utomo, A. Y., & Priyogi, B. (2016). Measuring user experience of the student-centered E-learning environment. *Journal of Educators Online*, *13*(1), 1–79.

Sarrab, M., & Rehman, O. M. H. H. (2014). Empirical study of open source software selection for adoption, based on software quality characteristics. *Advances in Engineering Software*, *69*, 1–11. https://doi.org/10.1016/j. advengsoft. 2013.12.001

Sauer, J., Sonderegger, A., & Schmutz, S. (2020). Usability, user experience and accessibility: towards an integrative model. *Ergonomics*, *63*(10), 1207–1220. https://doi.org/10.1080/00140139.2020.1774080

Saunders, M., Lewis, P., & Thornhill, A. (2019). Research Methods for Business Students Eighth Edition. In *Research Methods for Business Students* (8th Editio). England: Prentice Hall. Retrieved from www.pearson.com/uk%0Ahttps://www.amazon.com/Research-Methods-for-Business-Students/dp/1292208783/ref=sr_1_2?dchild=1&qid=1614706531&refinements=p_27%3AAdrian+Thornhill+%2F+Philip+Lewis+%2F+Mark+N.+K.+Saunders&s=books&sr=1-2&text=Adrian+Thornhill+%2F+Phili

Sauro, J., Johnson, K., & Meenan, C. (2017). From snake-oil to science: Measuring UX maturity. *Conference on Human Factors in Computing Systems - Proceedings*, *Part F1276*, 1084–1091. New York, New York, USA: ACM Press. https://doi.org/10.1145/3027063.3053350

Sauro, J., & Lewis, J. R. (2016). *Quantifying The User Experience. Practical Statistics for user research* (2nd ed.). Morgan Kaufmann.

Scacchi, W., Feller, J., Fitzgerald, B., Hissam, S., & Lakhani, K. (2006). Understanding free/open source software development processes. *Software Process Improvement and Practice*, *11*(2), 95–105. https://doi.org/10.1002/spip.255

Schemattow, M. (2009). Controlling the Usability Evaluation Process under Varying Defect Visibility. *People and Computers XXIII Celebrating People and Technology - Proceedings of HCI 2009*, 188–197. https://doi.org/10.14236/ewic/HCI2009.22

Schild, J., LaViola, J., & Masuch, M. (2012). Understanding user experience in stereoscopic 3D games. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 89–98. New York, NY, USA: ACM. https://doi.org/10.1145/2207676.2207690

Schmettow, M. (2012). Sample size in usability studies. *Communications of the ACM*, *55*(4), 64–70. https://doi.org/10.1145/2133806.2133824

Schmidt, R. C. (1997). Managing Delphi Surveys Using Nonparametric Statistical Techniques. *Decision Sciences*, *28*(3), 763–774. https://doi.org/10.1111/j.1540-5915.1997.tb01330.x

Schweik, C., & Semenov, A. (2003). The Institutional Design of Open Source Programming: Implications for Addressing Complex Public Policy and Management Problems. *First Monday*, *8*(1). https://doi.org/10.5210/fm.v8i1.1019

Scott, J. R., Thompson, A. R., Wright-thomas, D., Xu, X., & Barchard, K. A. (2008). *Data entry method: Is double entry the way to go?* 1–2.

SEI. (2000). CMMI for Systems Engineering/Software Engineering, Version 1.02, Staged Representation (CMMI-SE/SW, V1.02, Staged). Retrieved January 19, 2021, from http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=5263

SEI. (2002). Capability Maturity Model® Integration (CMMISM), Version 1.1 CMMISM for Software Engineering. *Software Engineering Institute*, (August), 1–639. Retrieved from http://www.sei.cmu.edu/reports/02tr029.pdf

SEI. (2010). CMMI® for Development, Version 1.3 CMMI-DEV, V1.3 - Improving processes for developing better products and services. *Engineering*, 482.

SEI - Software Engineering Institute. (2006). *Capability Maturity Model® for Development Version 1.2 (CMMI-DEV v1.2)*. (August).

Semeteys, R. (2008). Method for Qualification and Selection of Open Source Software. Retrieved October 5, 2020, from Open Source Business Resource website: http://timreview.ca/article/146

SenthilMurugan, C., & Prakasam, S. (2013). A Literal Review of Software Quality Assurance. *International Journal of Computer Applications*, *78*(8), 25–30. https://doi.org/10.5120/13511-1279

Senyard, A., & Michlmayr, M. (2004). How to Have a Successful Free Software Project. *11th Asia-Pacific Software Engineering Conference*, 84–91. IEEE. https://doi.org/10.1109/APSEC.2004.58

Services U S Department of Health and Human. (2015). User experience basics | Usability.gov. Retrieved June 18, 2020, from Usability.gov website: https://www.usability.gov/what-and-why/user-experience.html%0Ahttps://www.usability.gov/what-and-why/user-experience.html%0Ahttps://www.usability.gov/%0Ahttp://www.usability.gov/what-and-why/user-experience.html

Shanks, G., & Bekmamedova, N. (2018). Case study research in information systems. In *Research Methods* (pp. 193–208). Elsevier. https://doi.org/10.1016/B978-0-08-102220-7.00007-8

Shekgola, M., Maluleka, J., & Rodrigues, A. (2021). Factors influencing the adoption of free and open-source software for electronic records management by municipalities in Gauteng Province, South Africa. *Journal of the South African Society of Archivists*, *54*, 43–54. https://doi.org/10.4314/jsasa.v54i1.4

Shneiderman, B., & Plaisant, C. (2006). *The Eight Golden Rules of Interface Design* (6th ed.). Cambridge: Pearson Publishing.

Silva, D. G., Coutinho, C., & Costa, C. J. (2023). Factors influencing free and open-source software adoption in developing countries—an empirical study. *Journal of Open Innovation: Technology, Market, and Complexity*, *9*(1), 100002. https://doi.org/10.1016/j.joitmc.2023.01.002

Singh, A. (2011). Sampling Techniques & Determination of Sampel Size in Applied Statistics Reseaech: An Overview. *Inwood Magazine*, *II*(96), 32–33. Retrieved from http://ijecm.co.uk/

Sivaji, A., Abdullah, A., & Downe, A. G. (2011). Usability Testing Methodology: Effectiveness of Heuristic Evaluation in E-Government Website Development. *2011 Fifth Asia Modelling Symposium*, 68–72. IEEE. https://doi.org/10.1109/AMS.2011.24

Sohn, D., Leckenby, J. D., & Jee, J. (2000). The role of perceived interactivity in interactive ad processing. Retrieved from http://ezproxy.net.ucf.edu/login?url=http://search.proquest.com/docview/304620610?accountid=10003%5Cnhttp://sfx.fcla.edu/ucf?url_ver=Z39.88-2004&rft_val_fmt= info:ofi/fmt:kev:mtx:dissertation&genre=dissertations+&+theses&sid=ProQ:ProQuest+Dissertations+&+T

Sohn, D., & Lee, B.-K. (2006). Dimensions of Interactivity: Differential Effects of Social and Psychological Factors. *Journal of Computer-Mediated Communication*, *10*(3), 00–00. https://doi.org/10.1111/j.1083-6101.2005.tb00254.x

Solli-Sæther, H., & Gottschalk, P. (2010). The Modeling Process for Stage Models. *Journal of Organizational Computing and Electronic Commerce*, *20*(3), 279–293. https://doi.org/10.1080/10919392.2010.494535

Soomro, T. R., & Ahmad, R. (2012). Quality in Higher Education: United Arab Emirates Perspective. *Higher Education Studies*, *2*(4). https://doi.org/10.5539/hes.v2n4p148

Souley, B., & Sambo, A. S. (2013). A Comparative Performance Analysis of Popular Internet Browsers in Current Web Applications. *West African Journal of Industrial and Academic Research*, *4*(1), 62–68.

Spinellis, D. (2019). How to Select Open Source Components. *Computer*, *52*(12), 103–106. https://doi.org/10.1109/MC.2019.2940809

Spool, J., & Schroeder, W. (2001). Testing web sites. *CHI ’01 Extended Abstracts on Human Factors in Computing Systems - CHI ’01*, 285. New York, New York, USA: ACM Press. https://doi.org/10.1145/634067.634236

Srinivasa, K. G. (2017). *Free and Open Source Software in Modern Data Science and Business Intelligence: Emerging Research and Opportunities (Advances in Computer and Electrical Engineering)* (1st ed.; C. D. Ganesh & P. M. Krishnaraj, Eds.). IGI Global.

Staggers, N., Rodney, M., Alafaireet, P., Backman, C., Bochinski, J., Schumacher, B., … Xiao, Y. (2015). Promoting Usability in Health Organizations: Initial Steps and Progress Towards a Healthcare Usability Maturity Mode. In *Healthcare Information and Management Systems Society*. Retrieved from https://www.himss.org/sites/himssorg/files/HIMSSorg/Content/files/HIMSS_Promoting_Usability_in_Health_Org.pdf

Stallman, R. (2002). Free Software, Free Society. In *World Wide Web Internet And Web Information Systems*. Retrieved from http://shop.fsf.org/product/free-software-free-society-2/

Statista. (2021). Reasons for adopting open-source software according to companies worldwide 2019-2020. *Statista*. Retrieved from https://www.statista.com/statistics/629245/worldwide-enterprises-open-source-benefits/

Statscounter. (2019). Desktop Operating System Market Share Worldwide. *StatCounter Global Stats*, 1–5. Retrieved from https://gs.statcounter.com/os-market-share/desktop/worldwide

Steinmacher, I., Robles, G., Fitzgerald, B., & Wasserman, A. (2017). Free and open source software development: the end of the teenage years. *Journal of Internet Services and Applications*, *8*(1), 17. https://doi.org/10.1186/s13174-017-0069-9

Steinmacher, I., Treude, C., & Gerosa, M. A. (2019). Let Me In: Guidelines for the Successful Onboarding of Newcomers to Open Source Projects. *IEEE Software*, *36*(4), 41–49. https://doi.org/10.1109/MS.2018.110162131

Steuer, J. (1992). Defining Virtual Reality: Dimensions Determining Telepresence. *Journal of Communication*, *42*(4), 73–93. https://doi.org/10.1111/j.1460-2466.1992.tb00812.x

Stevens, J. P. (2012). Applied Multivariate Statistics for the Social Sciences. In *Applied Multivariate Statistics for the Social Sciences*. https://doi.org/10.4324/9780203843130

Stol, K.-J., & Ali Babar, M. (2010). Challenges in using open source software in product development. *Proceedings of the 3rd International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development - FLOSS '10*, 17–22. New York, New York, USA: ACM Press. https://doi.org/10.1145/1833272.1833276

Stol, K. J., & Ali Babar, M. (2010). A comparison framework for Open Source Software evaluation methods. In *IFIP Advances in Information and Communication Technology*: *Vol. 319 AICT* (pp. 389–394). https://doi.org/10.1007 /978-3-642-13244-5_36

Stone, M., Bentley, F., White, B., & Shebanek, M. (2016). Embedding User Understanding in the Corporate Culture. *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems, 07-12-May-*, 823–832. New York, NY, USA: ACM. https://doi.org/ 10.1145/ 2851581.2851592

Sung, W. J., Kim, J. H., & Rhew, S. Y. (2007). A Quality Model for Open Source Software Selection. *Sixth International Conference on Advanced Language Processing and Web Information Technology (ALPIT 2007)*, 515–519. IEEE. https://doi.org/10.1109/ALPIT.2007.81

Sutcliffe, A. (2009). Designing for User Engagement: Aesthetic and Attractive User Interfaces. *Synthesis Lectures on Human-Centered Informatics*, *2*(1), 1–55. https://doi.org/10.2200/S00210ED1V01Y200910HCI005

Swarts, J. (2019). Open-Source Software in the Sciences: The Challenge of User Support. *Journal of Business and Technical Communication*, *33*(1), 60–90. https://doi.org/10.1177/1050651918780202

Symons, J. (2019). The effect of deductive and inductive teaching methods in an introductory programming course. *American Society for Engineering Education*.

Tabachnick, B. G., & Fidell, L. S. (2019). Using Multivariate Statistics. In *Pearson Education, Inc* (7th Editio). 330 Hudson Street, NY NY 10013: Pearson.

Taber, K. S. (2018). The Use of Cronbach's Alpha When Developing and Reporting Research Instruments in Science Education. *Research in Science Education*, *48*(6), 1273–1296. https://doi.org/10.1007/s11165-016-9602-2

Taha, A. M., Abbood, A. A., Razzaq, A. A. A., & Al-Bahri, A. S. (2018). Identifying the affecting factors for adoption of open source software in it community. *Journal of Engineering and Applied Sciences*, *13*(14), 5771–5780. https://doi.org/10.3923/jeasci.2018.5771.5780

Taibi, D. (2015). An Empirical Investigation on the Motivations for the Adoption of Open Source Software. *ICSEA - The Tenth International Conference on Software Engineering Advances*, 426–431.

Taibi, D., Lavazza, L., & Morasca, S. (2007). OpenBQR: A framework for the assessment of OSS. In *IFIP International Federation for Information Processing* (Vol. 234, pp. 173–186). https://doi.org/10.1007/978-0-387-72486-7_14

Tang, C.-W., & Wu, C.-T. (2010). Obtaining a picture of undergraduate education quality: a voice from inside the university. *Higher Education*, *60*(3), 269–286. https://doi.org/10.1007/s10734-009-9299-5

Tarhan, A., & Giray, G. (2017). On the Use of Ontologies in Software Process Assessment. *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, 2–11. New York, NY, USA: ACM. https://doi.org/10.1145/3084226.3084261

Tassone, J., Xu, S., Wang, C., Chen, J., & Du, W. (2018). Quality Assessment of Open Source Software: A Review. *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, 411–416. IEEE. https://doi.org/10.1109/ICIS.2018.8466436

Taylor, J., & Dantu, R. (2022). For Love or Money? Examining Reasons behind OSS Developers' Contributions. *Information Systems Management*, *39*(2), 122–137. https://doi.org/10.1080/10580530.2021.1879323

Teixeira, J. G., Patrício, L., Huang, K.-H., Fisk, R. P., Nóbrega, L., & Constantine, L. (2017). The MINDS Method. *Journal of Service Research*, *20*(3), 240–258. https://doi.org/10.1177/1094670516680033

Ten, A. C., & Paz, F. (2017). A systematic review of user experience evaluation methods in information driven websites. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*: *Vol. 10288 LNCS* (pp. 492–506). https://doi.org/10.1007 /978-3-319-58634-2_36

Terry, M., Kay, M., & Lafreniere, B. (2010). Perceptions and practices of usability in the free/open source software (FOSS) community. *Conference on Human Factors in Computing Systems - Proceedings*, *2*, 999–1008. https://doi.org/10.1145/ 1753326.1753476

Thabane, L., Ma, J., Chu, R., Cheng, J., Ismaila, A., Rios, L. P., … Goldsmith, C. H. (2010). A tutorial on pilot studies: the what, why and how. *BMC Medical Research Methodology*, *10*(1), 1. https://doi.org/10.1186/1471-2288-10-1

Theofanidis, D., & Fountouki, A. (2018). Limitations and delimitations in the research process. *Perioperative Nursing*, *7*(3), 155–162. Retrieved from http://doi.org/10.5281/zenodo.2552022

Thomas, P., O’Brien, H., & Rowlands, T. (2016). Measuring Engagement with Online Forms. *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*, 325–328. New York, NY, USA: ACM. https://doi.org/10.1145/2854946.2854988

Thüring, M., & Mahlke, S. (2007). Usability, aesthetics and emotions in human–technology interaction. *International Journal of Psychology*, *42*(4), 253–264. https://doi.org/10.1080/00207590701396674

Titov, S., Bubnov, G., Guseva, M., Lyalin, A., & Brikoshina, I. (2016). Capability maturity models in engineering companies: case study analysis. *ITM Web of Conferences*, *6*, 03002. https://doi.org/10.1051/itmconf/20160603002

Todorovic, D. (2008). Gestalt principles. *Scholarpedia*, *3*(12), 5345. https://doi.org/10 .4249/scholarpedia.5345

Tognazzini, B. (2015). First Principles of Interaction Design (Revised & Expanded). Retrieved May 2, 2022, from AskTog website: https://asktog.com/atc/ principles-of-interaction-design/

Tomhave, B. L. (2005). Alphabet soup: Making sense of models, frameworks, and methodologies. *Egov.Ufsc.Br*, 1–57. Retrieved from http://egov.ufsc. br/portal/sites/ default /files/alphabet_ soup.pdf%5Cnwww. Secure consulting.net/Papers/Alphabet_Soup.pdf%5Cnhttp://secureconsulting.net/papers-publications.html

Tractinsky, N. (2018). The Usability Construct: A Dead End? *Human–Computer Interaction*, *33*(2), 131–177. https://doi.org/10.1080/07370024.2017.1298038

Traynor, B. (2022). UX Standards and UX Maturity. *Journal of Usability Studies*, *17*(2), 31–40.

Tsai, J. (2008). For Better or Worse: Introducing the GNU General Public License Version 3. *Berkeley Tech*, *23*, 547–581.

Tsou, M., & Smith, J. (2011). *Free and Open Source Software for GIS education*. (January), 1–18.

Tullis, T., & Albert, B. (2013). Measuring the User Experience. In *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics: Second Edition*. Elsevier. https://doi.org/10.1016/C2011-0-00016-9

Turner, P. (2008). Towards an account of intuitiveness. *Behaviour & Information Technology*, *27*(6), 475–482. https://doi.org/10.1080/01449290701292330

Twidale, M. B., & Nichols, D. M. (2005). Exploring usability discussions in open source development. *Proceedings of the Annual Hawaii International Conference on System Sciences,* 198. https://doi.org/10.1109/hicss.2005.266

Umair, M. (2018). Research Fundamentals: Study Design, Population, and Sample Size. *Undergraduate Research in Natural and Clinical Science and Technology (Urncst) Journal*, *2*(1), 1–7. Retrieved from https://www.researchgate. net/publication/322375665_Research_Fundamentals_Study_Design_Population _and_Sample_Size

Umm-E-Laila, Zahoor, A., Mehboob, K., & Natha, S. (2017). Comparison of open source maturity models. *Procedia Computer Science*, *111*, 348–354. https://doi. org/10. 1016/j.procs.2017.06.033

Unterkalmsteiner, M., Abrahamsson, P., Wang, X. F., Nguyen-Duc, A., Shah, S., Bajwa, S. S., … Yagüe, A. (2016). Software startups-A research agenda. *E-Informatica Software Engineering Journal*, *10*(1), 89–123. https://doi.org/ 10.5277/e-Inf160105

Uysal, Z. C. (2012). Architectural Type as a Cultural Schema and Its Cognitive Use in Architectural Design: An Analysis of the Aga Khan Award Winning Dwellings in Turkey. Retrieved from http://ezproxy.lib.ryerson. ca/login? url=https://search.proquest.com/docview/1033203402?accountid=13631%0Ahtt p://sfx.scholarsportal.info/ryerson??url_ver=Z39.88-2004&rft_ val_ fmt= info: ofi/fmt:kev:mtx:dissertation&genre=dissertations+%26+theses&sid=ProQ:ProQ

Väänänen-Vainio-Mattila, K., Roto, V., & Hassenzahl, M. (2008). Towards Practical User Experience Evaluation Methods. *Proceedings of the International Workshop on Meaningful Measure: Valid Useful User Experience Measurement (VUUM 2008)*, (April), 19–22. Retrieved from http://74.125. 155.132/scholar?q=cache:Wb6b-XnJXT0J:scholar. google.com/& hl=en&%2338;as_ sdt=2001

Vallerand, J., Lapalme, J., & Moïse, A. (2017). Analysing enterprise architecture maturity models: a learning perspective. *Enterprise Information Systems*, *11*(6), 859–883. https://doi.org/10.1080/17517575.2015.1091951

Van Angeren, J., Kabbedijk, J., Jansen, S., & Popp, K. M. (2011). A survey of associate models used within large software ecosystems. *CEUR Workshop Proceedings*, *746*, 27–39.

Van den Broeck, J., Sandøy, I. F., & Brestoff, J. R. (2013). The Recruitment, Sampling, and Enrollment Plan. In *Epidemiology: Principles and Practical Guidelines* (pp. 171–196). Dordrecht: Springer Netherlands. https://doi.org/10. 1007/978-94-007-5989-3_9

van der Burg, S., Dolstra, E., McIntosh, S., Davies, J., German, D. M., & Hemel, A. (2014). Tracing software build processes to uncover license compliance inconsistencies. *Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering*, 731–742. New York, NY, USA: ACM. https://doi.org/10.1145/2642937.2643013

van der Merwe, A., Gerber, A., & Smuts, H. (2020). Guidelines for Conducting Design Science Research in Information Systems. In *Communications in Computer and Information Science*: *Vol. 1136 CCIS* (pp. 163–178). https://doi.org/10.1007/978-3-030-35629-3_11

Van Looy, A., De Backer, M., & Poels, G. (2011). Defining business process maturity. A journey towards excellence. *Total Quality Management & Business Excellence*, *22*(11), 1119–1137. https://doi.org/10.1080/14783363.2011.624779

Van Steenbergen, M., Bos, R., Brinkkemper, S., Van De Weerd, I., & Bekkers, W. (2010). The design of focus area maturity models. *CEUR Workshop Proceedings*, *662*, 17–19.

Van Tyne, S. (2009). Corporate User-Experience Maturity Model. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*: *Vol. 5619 LNCS* (pp. 635–639). https://doi.org/10.1007/978-3-642-02806-9_74

Velmourougan, S., Dhavachelvan, P., Baskaran, R., & Ravikumar, B. (2014). Software development Life cycle model to build software applications with usability. *Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2014*, 271–276. https://doi.org/10.1109/ICACCI.2014.6968610

Venable, J., & Baskerville, R. (2012). Eating our own cooking: Toward a more rigorous design science of research methods. *Electronic Journal of Business Research Methods, 10*(2), 141–153. Retrieved from http://books.google.ch/books?hl=en&lr=&id=97kcAgAAQBAJ&oi=fnd&pg=PA399&ots=IzH-LljcXv&sig=mtZF-wfdbLg5HfnSZcX3ZisbhxU

Venkatesh, V., & Bala, H. (2008). Technology acceptance model 3 and a research agenda on interventions. *Decision Sciences*, *39*(2), 273–315. https://doi.org/10.1111/j.1540-5915.2008.00192.x

Vermeeren, A. P. O. S., Law, E. L. C., Roto, V., Obrist, M., Hoonhout, J., & Väänänen-Vainio-Mattila, K. (2010). User experience evaluation methods: Current state and development needs. *NordiCHI 2010: Extending Boundaries - Proceedings of the 6th Nordic Conference on Human-Computer Interaction*, 521–530. https://doi.org/10.1145/1868914.1868973

Virzi, R. A. (1992). Refining the Test Phase of Usability Evaluation: How Many Subjects Is Enough? *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *34*(4), 457–468. https://doi.org/10.1177/ 001872089203400407

Von-Wangenheim, C. G. V., Hauck, J. C. R., Salviano, C. F., & Von-Wangenheim, A. (2010). Systematic literature review of software process capability/maturity models. *10th International SPICE Conference on Software Process Improvement and Capability Determination, SPICE 2010*, (April 2015).

von Eye, A., & Mun, E. Y. (2014). Analyzing Rater Agreement. In *Analyzing Rater Agreement: Manifest Variable Methods*. Psychology Press. https://doi.org/10.4324/9781410611024

von Krogh, G., & von Hippel, E. (2006). The Promise of Research on Open Source Software. *Management Science*, *52*(7), 975–983. https://doi.org/10.1287/mnsc.1060.0560

Von Wangenheim, C. G., Hauck, J. C. R., Zoucas, A., Salviano, C. F., McCaffery, F., & Shull, F. (2010). Creating software process capability/maturity models. *IEEE Software*, *27*(4), 92–94. https://doi.org/10.1109/MS.2010.96

VonFange, M., & Lavigne, D. (2011). Cost Optimization Through Open Source Software. Retrieved December 13, 2022, from The Technology Innovation Management Review website: https://timreview.ca/article/408

Vuorio, J., Okkonen, J., & Viteli, J. (2017). Enhancing user value of educational technology by three layer assessment. *Proceedings of the 21st International Academic Mindtrek Conference, AcademicMindtrek 2017*, *2017-Janua*, 220–226. New York, New York, USA: ACM Press. https://doi.org/10.1145/3131085.3131105

Wagemans, J., Elder, J. H., Kubovy, M., Palmer, S. E., Peterson, M. A., Singh, M., & von der Heydt, R. (2012). A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure-ground organization. *Psychological Bulletin*, *138*(6), 1172–1217. https://doi.org/10.1037/a0029333

Wagenstein, H. N. (2006). A capability maturity model for training & education. *PMI® Global Congress 2006—EMEA*, *4*. Madrid, Spain. Newtown Square: PA: Project Management Institute.

Wale-Kolade, A., & Nielsen, P. A. (2016). Apathy Towards the Integration of Usability Work: A Case of System Justification. *Interacting with Computers*, *28*(4), 437–450. https://doi.org/10.1093/iwc/iwv016

Wallace, D. F., Norman, K. L., & Plaisant, C. (1988). The American Voice And Robotics "Guardian: System: A Case Study In User Interface Usability Evaluation. *University of Maryland*.

Wallach, D., Conrad, J., & Steimle, T. (2017). The UX metrics table: A missing artifact. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10288 LNCS*, 507–517. https://doi.org/10.1007/978-3-319-58634-2_37

Wang, Y., & King, G. (2000). Software engineering processes: Principles and applications. *Software Engineering Processes: Principles and Applications*, 1–736.

Watson, D, & Clark, L. . (1994). The PANAS-X: Manual for the positive and negative affect schedule ? expanded form [Internet]. *The University of Iowa*, (January 1999).

Watson, David, Clark, L. A., & Tellegen, A. (1988). Development and Validation of Brief Measures of Positive and Negative Affect: The PANAS Scales. *Journal of Personality and Social Psychology*, *54*(6), 1063–1070. https://doi.org/10.1037/0022-3514.54.6.1063

Weichert, S., Quint, G., & Bartel, T. (2018). *Quick Guide UX Management. So verankern Sie Usability und User Experience im Unternehmen*. Retrieved from http://www.springer.com/series/15709

Weick, K. E. (1995). Sensemaking in Organizations (Foundations for Organizational Science). In *Star*.

Weinschenk, S., & Barker, D. (2000). Designing Effective Speech Interfaces. In *John Wiley & Sons*. Retrieved from https://dl.acm.org/citation.cfm?id=332034%0Ahttp://go.galegroup.com/ps/i.do?id=GALE%7CA74827455&sid=googleScholar&v=2.1&it=r&linkaccess=fulltext&issn=00493155&p=AONE&sw=w&authCount=1&u=mannh&selfRedirect=true%0Ahttp://proquest.techbus.safaribooksonline.de

Wendler, R. (2012). The maturity of maturity model research: A systematic mapping study. *Information and Software Technology*, *54*(12), 1317–1339. https://doi.org/10. 1016/j.infsof.2012.07.007

Whitfield, W. (1995). Nursing Research: Theory and Practice. *Journal of Advanced Nursing*, *21*(1), 196–196. https://doi.org/10.1046/j.1365-2648.1995.21010191-12.x

WHO. (2001). *Health Research Methodology -A Guide for Training Research Methods* (2nd Editio). Retrieved from https://www.google.co.in/books/edition/Health_Research_Methodology/q1oCARkbSNgC?hl=en&gbpv=1&dq=health+research+methodology+a+guide+for+training+in+research+methods&printsec=frontcover

Wiegers, K. E., & Sturzenberger, D. C. (2000). A modular software process mini-assessment method. *IEEE Software*, *17*(1), 62–69. https://doi.org/10.1109/52.819970

Wiezer, N., Bakhuys Roozeboom, M., & Oprins, E. (2013). Serious Gaming Used as Management Intervention to Prevent Work-Related Stress and Raise Work–Engagement among Workers. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*: *Vol. 8026 LNCS* (pp. 149–158). https://doi.org/10.1007/978-3-642-39182-8_18

Wilcox, R. R. (1998). How many discoveries have been lost by ignoring modern statistical methods? *American Psychologist*, *53*(3), 300–314. https://doi.org/10.1037/0003-066X.53.3.300

William, A., & Thomas, T. (2013). *MeasuringThe User Experience: Collecting, Analyzing, and Presenting Usability Metrics*.

Williams, G. (2010). Maturity Model (P2MM). *The Office of Government and Commerce*, 1–34.

Yamane, T. (1967). *STATISTICS;An IntroductoryAnalysis* (2nd ed.). New York: Harper and Row.

Yang, B., Wei, L., & Pu, Z. (2020). Measuring and Improving User Experience Through Artificial Intelligence-Aided Design. *Frontiers in Psychology*, *11*. https://doi.org/10.3389/fpsyg.2020.595374

Yang, Q., Zimmerman, J., Steinfeld, A., & Tomasic, A. (2016). Planning Adaptive Mobile Experiences When Wireframing. *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, 565–576. New York, NY, USA: ACM. https://doi.org/10.1145/2901790.2901858

Yassien, E. (2020). The challenges of capability maturity model integration application in the dynamic environment. *International Journal of Information Systems and Change Management*, *12*(1), 17. https://doi.org/10.1504/IJISCM.2020.112045

Yeates, N. (2015). The future of open source is a better user experience. Retrieved December 23, 2022, from Opensource.com website: https://opensource.com/life/15/3/user-experience-open-source-future#comment-41555

Yeh, K. B., Adams, M. L., Marshall, E. S., Dasgupta, D., Zhunushov, A., Richards, A. L., & Hay, J. (2017). Applying a Capability Maturity Model (CMM) to evaluate global health security-related research programmes in under-resourced areas. *Global Security: Health, Science and Policy*, *2*(1), 1–9. https://doi.org/10.1080/23779497.2017.1279022

Yong, L. T. (2013). User experience evaluation methods for mobile devices. *Third International Conference on Innovative Computing Technology (INTECH 2013)*, 281–286. IEEE. https://doi.org/10.1109/INTECH.2013.6653647

Young, S. W. H., Chao, Z., & Chandler, A. (2020). User Experience Methods and Maturity in Academic Libraries. *Information Technology and Libraries*, *39*(1). https://doi.org/10.6017/ital.v39i1.11787

Yusop, N. S. M., Grundy, J., & Vasa, R. (2017). Reporting Usability Defects: A Systematic Literature Review. *IEEE Transactions on Software Engineering*, *43*(9), 848–867. https://doi.org/10.1109/TSE.2016.2638427

Zaffar, M. A., Kumar, R. L., & Zhao, K. (2011). Diffusion dynamics of open source software: An agent-based computational economics (ACE) approach. *Decision Support Systems*, *51*(3), 597–608. https://doi.org/10.1016/j.dss.2010.11.034

Zarour, M., & Alharbi, M. (2017). User experience framework that combines aspects, dimensions, and measurement methods. *Cogent Engineering*, *4*(1), 1421006. https://doi.org/10.1080/23311916.2017.1421006

Zhang, P., Carey, J., Te'eni, D., & Tremaine, M. (2005). Integrating Human-Computer Interaction Development into the Systems Development Life Cycle: A Methodology. *Communications of the Association for Information Systems*, *15*. https://doi.org/10.17705/1cais.01529

Zhang, P., & Li, N. (2004). Love at First Sight or Sustained Effect? The Role of Perceived Affective Quality on Users' Cognitive Reactions to Information Technology. *Icis*, 283–296.

Zhao, J. D., Liu, Y., & Zhu, L. (2016). Analysis of Competition between Open Source Software and Proprietary Software. *Proceedings of the 2016 6th International Conference on Applied Science, Engineering and Technology*. Paris, France: Atlantis Press. https://doi.org/10.2991/icaset-16.2016.44

Zhao, L., & Deek, F. P. (2005). Improving open source software usability. *Association for Information Systems - 11th Americas Conference on Information Systems, AMCIS 2005: A Conference on a Human Scale*, *6*, 2858–2863.

Zhao, L., Deek, F. P., & McHugh, J. A. (2010). Exploratory inspection-a user-based learning method for improving open source software usability. *Journal of Software Maintenance and Evolution*, *22*(8), 653–675. https://doi.org/10.1002/smr.455

Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, *237*, 350–361. https://doi.org/10.1016/j.neucom.2017.01.026

Zolkifli, N. N., Ngah, A., & Deraman, A. (2018). Version Control System: A Review. *Procedia Computer Science*, *135*, 408–415. https://doi.org/10.1016/j.procs.2018.08.191

APPENDICES

Appendix I: Interview Guide

INTERVIEW GUIDE FOR STAKEHOLDERS IN THE FOSS COMMUNITY

Dear FOSS stakeholders,

This survey aims to gather your thoughts on UX maturity and the factors influencing it in the FOSS community. Your feedback will help get deeper insights into the factors influencing the UX maturity of FOSS projects.

All collected data will anonymously be kept and used only to meet the study's goal.

This interview has four questions and is expected to last between thirty and sixty minutes. You are kindly requested to provide honest answers.

1. What do you understand by the term User Experience maturity?
2. The Nielsen and Norman group defines UX maturity as "measures an organization's desire and ability to deliver user-centered design successfully." Does your understanding of UX maturity differ from Nielsen and Norman's group? Kindly clarify your response.
3. From your experience of working in the FOSS community, have you been involved directly or indirectly in either planning, executing, or reporting UX maturity assessment? If yes, what experience have you acquired?
4. Based on your knowledge, what factors influence UX maturity in the FOSS community? If possible, clarify how they affect the community.

Thank you very much for your time.

Appendix II: Introducing the Fuzzy Delphi Method, Lenses for Preparing FOSS Community for UX Maturity

E-MAIL TO EXPERTS TO DETERMINE LENSES FOR PREPARING THE FOSS COMMUNITY FOR UX MATURITY.

Dear XXXXXXXX

Thank you for agreeing to participate in a survey for a PhD study entitled **"Optimizing the Design for UX in the Development Process of Desktop-based FOSS Projects."** Your participation will be kept confidential and anonymous throughout the study. The information acquired in this survey will not be made public and will only be used to achieve the study's objectives.

The study aims to identify and rank critical issues or lenses to assess the readiness of the FOSS community for the UX maturity of its projects. This e-mail serves as an introduction to a Fuzzy Delphi questionnaire. As a requirement, the fuzzy Delphi process requires a firm commitment from engaged respondents. It will iteratively involve questioning you to establish a list of possible factors influencing the readiness for UX maturity in the FOSS community. You and other panelists will also validate the gathered analytical.

In different stages, you will be asked to fill in open-ended and 7-point rating scale questionnaires ranging from extremely unimportant (1) to extremely important (7).

Regards

Namayala, Phesto.

Appendix III: Establishing Possible Analytical Lenses for Determining Preparation of FOSS community for UX maturity

This questionnaire aims to establish lenses affecting the preparation of the FOSS community for UX maturity. Please submit honest and sincere responses to help enhance UX design in the development process of desktop-based FOSS applications. The information and data obtained from respondents are strictly confidential and will only be utilized for this study.

1. Date : …../…../20…
2. Gender ☐ Male ☐ Female
3. Academic rank ☐ Professor ☐ Assistant professor ☐ Senior lecturer ☐ Lecturer ☐ Others
4. Academic working experience in years ………………………………
5. Country or residence ……………………………………………………
6. The completed literature review identified dozens of factors and detailed descriptions of why they might affect UX maturity in FOSS projects. Twelve factors, as shown below, are given to you and other panelists for brainstorming and discussion while discovering the right factors

.Sn.	Lenses	Reference(s)	Initial Justification(s)
1	Accommodate UX professionals as decision-makers	Cheng and Guo (2018) and Inal et al. (2020)	"Having decision-making teams unfamiliar with UX fail projects aimed at producing goods with desirable UX."
2	Availability of FOSS-exclusive UX assessment tools	Namayala et al. (2022), Terry et al. (2010), Isomursu et al.(2012), Law et al.(2014) and Yeates (2015)	"When developing methodologies, strategies, tools, and frameworks for assessing UX maturity, the FOSS community must consider its dynamics."
3	Uniform understanding of UX	Garcia et al.(2019),	"A shared experience of UX maturity evaluation ideas boosts

.Sn.	Lenses	Reference(s)	Initial Justification(s)
	concepts across the organization	Lallemand et al. (2014), and Kuusinen (2015)	each person's excitement for the job by developing empathy and a spirit of aiding one another in completing tasks to meet predetermined goals.”
4	Value the UX-related tasks.	Wale-Kolade and Nielsen (2016)	“Not valuing UX-related tasks may lead to inadequate allocation of resources for non-technical-related activities.”

Based on your experience, what other factors affect the readiness of the FOSS community to assess its UX maturity, and how? Kindly provide your feedback in a tabular form, and you may also repeat or delete factors from the list above to indicate your agreement or disagreement.

Appendix IV: Final Analytical Lenses for Evaluating Preparation of the FOSS Community for UX Maturity

This questionnaire aims to establish the final list of analytical lenses for assessing the readiness of UX maturity in the FOSS community. Please submit honest and sincere responses to help enhance UX design in the development process of desktop-based FOSS applications. The information and data obtained from respondents are strictly confidential and will only be utilized for this study.

1. Please select seven (7) Analytical lenses from the last session that you think can be kept or updated and list them in no particular order of importance in the table below.

Item no.	**Factor**	**Justification**
1		
2		
3		
4		
5		
6		
7		

Appendix V: Ranking, Analytical Lenses to Evaluate the Preparation of FOSS Projects for UX Maturity

Dear XXXXX

This questionnaire aims to get experts' consensus on the identified analytical lenses and ranks based on their importance. Please submit honest and sincere responses to help enhance UX design in the development process of desktop-based FOSS applications. The information and data obtained from respondents are strictly confidential and will only be utilized for this study.

Interpretation of scale (1) Extremely Unimportant, (2) Very Unimportant, (3) Unimportant, (4) Moderately Important, (5) Important, (6) Very Important, and (7) Extremely Important										
Sn.	**Analytical Lens**	**Score**							**Your previous score**	**Median score**
1	Availability of UX strategic plans	1	2	3	4	5	6	7		
	Total responses									
2	Accommodate UX professionals as decision-makers.	1	2	3	4	5	6	7		
	Total responses									
3	Clear UX roles for each stakeholder	1	2	3	4	5	6	7		
	Total responses									
4	Value UX-related tasks.	1	2	3	4	5	6	7		
	Total responses									
5	Adopting UCD practices in design	1	2	3	4	5	6	7		
	Total responses									
6	Decision-makers buy-in and support UX assessment initiatives.	1	2	3	4	5	6	7		
	Total responses									
7	Availability of FOSS-exclusive UX maturity assessment tools	1	2	3	4	5	6	7		
	Total responses									
8	Accommodate UX principles in FOSS design.	1	2	3	4	5	6	7		
	Total responses									
9	Uniform understanding of UX concepts.	1	2	3	4	5	6	7		
	Total responses									

Interpretation of scale (1) Extremely Unimportant, (2) Very Unimportant, (3) Unimportant, (4) Moderately Important, (5) Important, (6) Very Important, and (7) Extremely Important										
10	Quantify UX maturity assessment metrics.	1	2	3	4	5	6	7		
	Total responses									

Thank you

Appendix VI: Stakeholders' Perceptions, UX Maturity Assessment Readiness

QUESTIONNAIRE FOR STAKEHOLDERS IN THE FOSS COMMUNITY

Dear FOSS project stakeholders

This questionnaire aims to collect FOSS stakeholders' perceptions regarding preparing FOSS projects for UX maturity. Based on your opinions, the questionnaire examines whether identified analytical lenses impact UX maturity. Kindly provide your honest and sincere responses to improve the design for UX in the development process of desktop-based FOSS projects. Respondents' information and data collected are confidential and will only be used for this study.

1. Date : …../…../20…
2. Gender ☐ Male ☐ Female
3. Role (s) ☐ Project Leader ☐ Core Member ☐ Active Developer ☐ Periphery Developer ☐ Bugs Patcher☐ Active User ☐ Passive user ☐ Others
4. Specialty ☐ Software engineer ☐ Computer scientist ☐ Programmer ☐ Interaction designer ☐ HCI specialist ☐ Others
5. Experience ☐ 0-5 years ☐ 5-10 years ☐ 10-15 years ☐ 15+ years
6. Software skills☐ Extremely experienced and technically knowledgeable ☐ Extremely experienced with no or limited technical knowledge ☐ Can efficiently use most software ☐ Find it difficult to use most software ☐ Others.
7. Opinion: Key: SA = Strong Agree, A = Agreed, N = Not sure, D = Disagree, and SD = Strong Disagree

No.	**Statement (S)**	**SD**	**D**	**N**	**A**	**SA**
S.1	Availability of UX strategic plans influences UX maturity					
S.2	Accommodating UX professionals as decision-makers influences UX maturity.					

S.3	The presence of clear UX roles for each stakeholder influences UX maturity.					
S.4	Value UX-related tasks influence UX maturity.					
S.5	Adopting UCD practices in design positively affects UX maturity.					
S.6	Decision-makers' buy-in and support of UX assessment initiatives contribute to enhancing UX maturity.					
S.7	The availability of FOSS-exclusive UX maturity assessment tools plays a significant role in improving UX maturity.					
S.8	Accommodating UX principles in FOSS design promotes UX maturity.					
S.9	A uniform understanding of UX concepts enhances UX maturity.					
S.10	Quantifying UX maturity assessment metrics promotes improving UX maturity.					
Preparation of FOSS community for UX maturity						
R.1	Organizations must implement institutional UX/HCI practices to achieve higher levels of UX maturity.					
R.2	To promote UX maturity, the FOSS community must allow the integration of UX specialists and other UX/HCI practices.					
R.3	The UX and its maturity is an organizational endeavor and can be improved by hiring experts in UX/HCI.					
R4	Organizations ready to assess their UX maturity are flexible to design change.					
R5	Adopting or creating novel UXCMMs for the FOSS community may not make a difference without the necessary preparations for UX maturity.					

Appendix VII: E-mail Introducing the Fuzzy Delphi Study to Identify Factors Influencing FOSS Community UX Maturity

E-MAIL TO SCHOLAR TO DETERMINE UX MATURITY INFLUENCING FACTORS IN THE FOSS COMMUNITY.

Dear xxxxxxxxx

Thank you for agreeing to participate in a survey for a PhD study entitled **"Optimizing the Design for UX in the Development Process of Desktop-based FOSS Projects."** Your participation will be kept confidential and anonymous throughout the study. The information acquired in this survey will not be made public and will only be used to achieve the study's objectives.

The study aims to identify and rank UX maturity influencing factors exclusive to the FOSS community. This e-mail serves as an introduction to a Fuzzy Delphi questionnaire. As a requirement, the fuzzy Delphi process requires a firm commitment from engaged respondents. It will iteratively involve questioning you to establish a list of possible UX maturity influencing factors in the FOSS community. You and other panelists will also validate the gathered factors.

In different stages, you will be asked to fill in open-ended and 7-point rating scale questionnaires ranging from extremely unimportant (1) to extremely important (7).

Namayala, Phesto.

Appendix VIII: Establishing Possible UX Maturity Influencing Factors

Dear XXXXX

This questionnaire will collect the factors influencing UX maturity in the FOSS community. Please submit honest and sincere responses to help enhance UX design in the development process of the desktop-based FOSS applications. The information and data obtained from respondents are strictly confidential and will only be utilized for this study.

1. Date : …../…../20…
2. Gender ☐ Male ☐ Female
3. Academic rank ☐ Professor ☐ Assistant professor ☐ Senior lecturer ☐ Lecturer ☐ Others
4. Academic working experience in years ………………………………
5. Country or residence ……………………………………………………
6. The completed literature review identified dozens of factors and detailed descriptions of why they might affect UX maturity in FOSS projects. As shown below, twenty-one factors are given to you and other panelists for brainstorming and discussion while discovering the right factors.

Sn.	Factor	Source(s)
1	Allocating adequate financial resources for UX-related works	Van Tyne (2009), MacDonald et al.(2022), Chapman and Plewes(2014), Rukonić et al.(2019), Peres et al.(2014), Sauro et al.(2017), and Young et al.(2020)
2	Dedicate UX staff with specialized UX skills.	Van Tyne (2009), MacDonald et al.(2022), Chapman and Plewes(2014), Rukonić et al.(2019), Peres et al.(2014), Sauro et al.(2017), and Young et al.(2020)
3	UX leadership	MacDonald et al.(2022), Chapman and Plewes(2014), Peres et al.(2014), Sauro et al.(2017), and Young et al.(2020).
4	Include UX professionals as decision-makers	Raza et al.(2012a) and Terry et al. (2010)

Sn.	Factor	Source(s)
5	Adopt UCD techniques	Young et al.(2020) and Rukonić et al.(2019).
6	Consider UX as a professional	Rukonić et al.(2019), Raza et al.(2012a), and Terry et al. (2010)
7	Buy-in of UX ideas	Rukonić et al.(2019), Peres et al.(2014), Sauro et al.(2017), and Young et al.(2020).
8	Clear UX maturity roles for each stakeholder	Chapman and Plewes(2014), and Rukonić et al.(2019).
9	Timing to UX techniques in the development process	Chapman and Plewes(2014) and Peres et al.(2014)
10	Regular and systematic use of UX research methods	Van Tyne (2009), Pernice et al. (2021), Anchahua et al.(2018)and Rukonić et al.(2019),
11	Formalise UX design standards.	MacDonald et al.(2022), Chapman and Plewes(2014), Peres et al.(2014), Sauro et al.(2017), and Young et al.(2020).
12	Participation of stakeholders beyond the UX team during UX design	Young et al.(2020), Rukonić et al.(2019) and Pernice et al (2021)
13	Regular UX training	Peres et al.(2014) and MacDonald et al.(2022).
14	Esteeming UX skill sets similar to technical skills.	Raza et al. (2012a), Peres et al.(2014), and Sauro et al.(2017).
15	Organizational flexibility	Sauro et al.(2017) and Rukonić et al.(2019)
16	UCD goals predicting UX metrics.	Rukonić et al.(2019) and Jokela and Abrahamsson (2000)
17	Regular review and update of UX metrics	Jokela and Abrahamsson (2000), Pernice et al. (2021), and Chapman and Plewes(2014).
18	UX impacts all project stakeholders.	Pernice et al. (2021), Rukonić et al.(2019) and MacDonald et al.(2022).

Sn.	Factor	Source(s)
19	constantly monitoring UX quality	Namayala et al. (2022), Pernice et al. (2021) and Rukonić et al.(2019),
20	Users' needs	Peres et al.(2014), Chapman and Plewes(2014), and Van Tyne (2009).
21	Consistently tracking UX-related tasks.	Van Tyne (2009)and Sauro et al.(2017).

Based on your experience, what other factors affect the UX maturity of the FOSS community, and how? Kindly provide your feedback in a tabular form, and you may also repeat or delete factors from the list above to indicate your agreement or disagreement.

Thank you

Appendix IX: Final Factors Affecting UX Maturity of FOSS Projects

Dear XXXXX

This questionnaire aims to establish the final factors influencing UX maturity. Please submit honest and sincere responses to help enhance UX design in the development process of desktop-based FOSS applications. The information and data obtained from respondents are strictly confidential and will only be utilized for this study.

Please select thirty (30) UX maturity influencing factors that may be kept or updated from the last session and list them in no particular order of importance in the table below.

Item no.	**Factor**	**Justification**
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		

22		
23		
24		
25		
26		
27		
28		
29		
30		

Thank you

Appendix X: Ranking, Factors Influencing UX Maturity in FOSS

Dear XXXXX

This questionnaire aims to get experts' agreement on the UX maturity influencing factors and rank them based on their importance. Please submit honest and sincere responses to help enhance UX design in the development process of desktop-based FOSS applications. The information and data obtained from respondents are strictly confidential and will only be utilized for this study.

Interpretation of scale (1) Extremely Unimportant, (2) Very Unimportant, (3) Unimportant, (4) Moderately Important, (5) Important, (6) Very Important, and (7) Extremely Important										
Sn.	**Factor**	**Score**							**Your previous score**	**Median score**
1	Documentations and support	1	2	3	4	5	6	7		
	Total responses									
2	Ease of use	1	2	3	4	5	6	7		
	Total responses									
3	Speed of use	1	2	3	4	5	6	7		
	Total responses									
4	learnability	1	2	3	4	5	6	7		
	Total responses									
5	Users' feedback	1	2	3	4	5	6	7		
	Total responses									
6	Bugs reporting procedure	1	2	3	4	5	6	7		
	Total responses									
7	UX/usability testing and quality assurance	1	2	3	4	5	6	7		
	Total responses									
8	Meeting users' requirements	1	2	3	4	5	6	7		
	Total responses									
9	User-centered Design (UCD) Methodology	1	2	3	4	5	6	7		
	Total responses									
10	Operability	1	2	3	4	5	6	7		
	Total responses									
11	Accessibility	1	2	3	4	5	6	7		
	Total responses									
12	Attractiveness	1	2	3	4	5	6	7		
	Total responses									
13	Artificial Intelligence (AI)	1	2	3	4	5	6	7		
	Total responses									
14	Adopted FOSS development methods	1	2	3	4	5	6	7		
	Total responses									

Interpretation of scale (1) Extremely Unimportant, (2) Very Unimportant, (3) Unimportant, (4) Moderately Important, (5) Important, (6) Very Important, and (7) Extremely Important										
15	Technical infrastructures, such as software development tools, libraries, and frameworks	1	2	3	4	5	6	7		
	Total responses									
16	FOSS UX leadership	1	2	3	4	5	6	7		
	Total responses									
17	Planning of FOSS UX-related activities	1	2	3	4	5	6	7		
	Total responses									
18	FOSS allocates adequate financial resources for UX-related works	1	2	3	4	5	6	7		
	Total responses									
19	FOSS dedicated UX staff with specialized UX skills	1	2	3	4	5	6	7		
	Total responses									
20	FOSS developers' UX skillset	1	2	3	4	5	6	7		
	Total responses									
21	FOSS Regular training on UX-related activities	1	2	3	4	5	6	7		
	Total responses									
22	FOSS Esteeming UX skill sets similar to technical skills	1	2	3	4	5	6	7		
	Total responses									
23	The organizational level of flexibility in FOSS	1	2	3	4	5	6	7		
	Total responses									
24	FOSS considering UX as a professional	1	2	3	4	5	6	7		
	Total responses									
25	FOSS buy-in of UX ideas or executive understanding	1	2	3	4	5	6	7		
	Total responses									
26	Clear UX maturity roles for each FOSS stakeholder	1	2	3	4	5	6	7		
	Total responses									
27	FOSS includes UX professionals as decision-makers	1	2	3	4	5	6	7		
	Total responses									
28	Regular and systematic use of UX research methods in FOSS projects	1	2	3	4	5	6	7		
	Total responses									
29	FOSS formalizes UX design standards.	1	2	3	4	5	6	7		
	Total responses									
30	Timing to include UX methods, practices, principles, and tools	1	2	3	4	5	6	7		
	Total responses									

Interpretation of scale (1) Extremely Unimportant, (2) Very Unimportant, (3) Unimportant, (4) Moderately Important, (5) Important, (6) Very Important, and (7) Extremely Important										
31	Consistently tracking FOSS UX-related tasks.	1	2	3	4	5	6	7		
	Total responses									
32	Constantly monitoring FOSS UX quality	1	2	3	4	5	6	7		
	Total responses									
33	UCD goals predicting FOSS UX metrics	1	2	3	4	5	6	7		
	Total responses									
34	Regular review/update of FOSS UX metrics	1	2	3	4	5	6	7		
	Total responses									
35	FOSS stakeholders' engagements	1	2	3	4	5	6	7		
	Total responses									
36	Number of errors made by users	1	2	3	4	5	6	7		
	Total responses									

Appendix XI: The Observation Checklist

Research Title: Optimizing the Design for User Experience in the Development Process of Desktop-based Free and Open Source Software Projects:

General objective: To optimize the design for UX in the development processes of desktop-based FOSS projects.

FOSS project: ……………………………… Date: …………………………………

Inclusion/exclusion criteria of projects	
1	Does the project have discussion forums such as a mailing list?
2	Does the project have a weekly downloads count above 100
3	Is the user rating of the project greater or equal to 4.5?
4	Has the project adopted some UX/HCI practices or UX/HCI assessment practices?
5	Is the project recently updated? (Note: the last update must not exceed two years.)
Inclusions/exclusions criteria of stakeholders	
1	Is the stakeholder an active member of mailing lists by regularly contributing to ongoing discussions?
2	Is the e-mail address of an active stakeholder easily accessible?
3	Does the stakeholder indicate knowledge of UX/HCI and their assessment from their contributions to discussion forums?
4	Is the stakeholder willing to participate in other studies?

Appendix XII: List of Publications

DETAILS AND STATUS OF PAPERS

S #	Title	Publisher and Journal	Status
1	The practical applicability of user experience capability/maturity models in the development processes of free and open-source software: a systematic literature review	Taylor and Francis, Behavior & Information Technology Journal	Published (2022)
2	The Comparative Analysis of the Contemporary User Experience Capability/Maturity Models: The Roadmap to an Acceptable Improvement.	Taylor and Francis, The Journal of Quality Management Control	The first revision is under review.
3	Application of Fuzzy Delphi Technique to Identify Analytical Lenses for Determining the Preparation of Free and Open Source Software Projects for User Experience Maturity	Elsevier, The Journal of Science of Computer Programming	The first draft of the manuscript is under review.
4	The Factors Affecting User Experience Maturity in Free and Open Source Software Community: An Empirical Study.	Taylor and Francis, Behavior & Information Technology Journal	Published (2023).
5	Institutionalizing the adoption of Free and Open Source Software in Tanzania: A Case of Selected Higher Education Institutions.	University of Dar es Salaam, Journal of ICT Systems	The first draft is under review.

Appendix XIII: List of Examined FOSS Projects

Sn	Project Name
1	GnuAccounting
2	uniCenta oPOS - dynamically evolving POS project
3	FrontAccounting (FA)
4	webERP Accounting & Business Management
5	TurboCASH Accounting
6	ADempiere ERP Business Suite
7	Compiere ERP + CRM Business Solution
8	Open Tax Solver
9	Turquaz Financial Accounting
10	Applewood Computers Accounting System
11	JFS Accounting
12	PortableApps.com
13	Tiki Wiki CMS Groupware
14	OOoDocs
15	OxygenOffice Professional - Office Suite
16	odt2braille
17	IGSuite
18	GanttProject
19	]project-open[- Project Server
20	Software Process Dashboard
21	Kablink
22	TestLink
23	dotProject
24	Feng Office: Project Management and more
25	Ganib | Project Management Software
26	LAMP Application Server
27	Tux Paint
28	DeSmuME
29	DOSBox
30	Warzone 2100
31	FlightGear - Flight Simulator
32	FCEUX
33	SuperTuxKart
34	Scid vs. PC
35	ChessX
36	The Guide
37	OpenDocMan
38	TemaTres: controlled vocabulary server
39	Dev-C++
40	Tcl

41	libjpeg-turbo
42	GnuWin
43	SWIG
44	libusb
45	PMD
46	Lazarus
47	TurboVNC
48	VirtualGL
49	Logisim
50	Skim
51	gnuplot
52	GeoServer
53	Maxima -- GPL CAS based on DOE-MACSYMA
54	CiteSpace
55	GeographicLib
56	Hibernate
57	Quite Universal Circuit Simulator
58	YAT
59	Ham Radio Control Libraries
60	OpenEMR
61	wxMaxima
62	NetEmul

Appendix XIV: Research Permit

JAMHURI YA MUUNGANO WA TANZANIA

OFISI YA RAIS
TAWALA ZA MIKOA NA SERIKALI ZA MITAA

Anuani ya Simu "TAMISEMI" DODOMA
Simu Na: +255 26 2321607
Nukushi: +255 26 2322116
Barua pepe:ps@tamisemi.go.tz
Unapojibu tafadhali taja:-

Mji wa Serikali – Mtumba,
Mtaa wa TAMISEMI,
S.L.P. 1923,
41185 DODOMA.

Kumb. Na. AB.307/323/01/ 69

27 Septemba, 2021

Katibu Tawala wa Mkoa,
DODOMA.

Yah: **KIBALI CHA KUFANYA UTAFITI KUHUSU *OPTIMISING THE USER EXPERIENCES IN THE DEVELOPMENT PROCESS OF DESKTOP-BASED FREE AND OPEN-SOURCE SOFTWARE PROJECTS***

Tafadhali rejea somo tajwa hapo juu.

2. Ofisi ya Rais –TAMISEMI imetoa kibali kwa **Bw. Festo P. Namayala** Mwanafunzi kutoka Chuo Kikuu cha Dodoma kwa ajili ya kufanya utafiti tajwa katika Mkoa wa Dodoma.

3. Muda wa kufanya utafiti huu ni kati ya mwezi Agosti, 2021 na mwezi Desemba, 2021. Ofisi ya Rais -TAMISEMI kwa kushirikiana na Taasisi nyingine za Serikali itafanya ukaguzi wakati wowote kujiridhisha na utekelezaji sahihi wa kibali hiki. Takwimu zitakazokusanywa kutokana na utafiti huu ni kwa ajili ya matumizi ya ndani tu na iwapo zitatakiwa kuchapishwa na kusambazwa kibali kutoka Mamlaka husika kitapaswa kuombwa.

4. Kwa barua hii, tafadhali waelekeze Wakurugenzi wa Halmashauri za Mkoa wako ili kutoa ushirikiano utakaohitajika na kukamilisha utafiti huu kama ulivyokusudiwa. Kazi hii isimamiwe na Mtakwimu wa Mkoa na Halmashauri husika na kutoa taarifa ya utekelezaji.

Ninakushukuru kwa ushirikiano wako.

Prof. Riziki S. Shemdoe
KATIBU MKUU