Application of quality in use model to assess the user experience of open source digital forensics tools

Manar Abu Talib*

Department of Computer Science,
University of Sharjah, United Arab Emirates
Email: mtalib@sharjah.ac.ae
*Corresponding author

Reem Alnanih

Faculty of Computing and Information Technology,
Department of Computer Science,
King Abdulaziz University,
Jeddah, Saudi Arabia
Email: ralnanih@kau.edu.sa

Adel Khelifi

College of Computer and Information Technology,
American University in the Emirates,
Dubai, UAE
Email: adel.khelifi@aue.ae

Abstract: Open source digital forensics tools are playing an important role for forensics investigations. So, there is a need to assess these tools to ensure they meet users' needs. The existing literature does not satisfy the requirements of assessing their quality-in-use. This paper assesses three most used digital forensics tools, namely autopsy, DFF, and DART using five quality-in use characteristics, namely, effectiveness, productivity, efficiency, error safety, and cognitive load. The results demonstrated that Autopsy, DFF, and DART performances are similar in terms of efficiency and productivity. DDF outperformed the other two in effectiveness. Autopsy was the best in terms of error safety, and DART had the highest cognitive load. The relevant community may consider these findings in selecting solutions to perform its duties. The future researches can involve more studies to encompass additional aspects of software quality, to compare open and closed source digital forensics programs and to enhance testing efforts

Keywords: quality models; open source software; OSS; digital forensics tools; autopsy; DFF; DART; quality-in-use model; ISO/IEC 25010.

Reference to this paper should be made as follows: Talib, M.A., Alnanih, R. and Khelifi, A. (2020) 'Application of quality in use model to assess the user experience of open source digital forensics tools', *Int. J. Electronic Security and Digital Forensics*, Vol. 12, No. 1, pp.43–76.

Biographical notes: Manar Abu Talib is Assistant Dean of College of Sciences at Sharjah University in the UAE. Her research interest includes software engineering, software measurement, software quality, software testing, information security, blockchain, IoT and open source software. She published many scientific articles, involved in more than 200 professional activities and sponsored research activities. She received the Best Teacher Award two times, the Exemplary Faculty Award in 2008 and 2010, Google CS4HS Award in 2014, QCRI ArabWIC and Anita Borg Institute Faculty scholarships in 2015, outstanding University and Community Service Award in 2016 and Exemplary Leader Award in WiSTEM 2016.

Reem Alnanih is an Assistant Professor in the Computer Science Department, Faculty of Computing and Information Technology in King Abdulaziz University. Jeddah, Saudi Arabia. She holds her PhD in Computer Science from Concordia University, Montreal, Canada, 2015. She obtained her Masters' degree in Computer Science from King Abdulaziz University, Jeddah, Saudi Arabia (2008). Her research interests are human computer interaction, designing mobile user interfaces, user experience, software quality measurement and associated evaluation techniques. Most of her publications are in the area of designing user interfaces and assessing the quality of software applications.

Adel Khelifi is the Dean of Computer Information Technology at the American University in the Emirates, Dubai, UAE. With a PhD from the Engineering School of High Technology, Canada (2005), he holds a high level of knowledge and expertise. Currently, he is involved in prompting the open source software paradigm in the region. He has held impressive past careers, previously working as a Lecturer for the Engineering School of Technology in Canada, United Nations MSF in Canada, and Ministry of Relations with Citizen and Immigration in Canada, and Ministry of Finances in Tunisia.

1 Introduction

Digital forensics is a new field, in which computerised forensics tools are being developed to acquire, authenticate, or analyse electronic content. Such tools are needed by investigators in this field to produce results that are both reliable enough to satisfy legal requirements and acceptable in the courts. In the USA, they must meet the four main Daubert criteria, which require that the tools be:

1 testable and accurate

2 peer-reviewed,

3 accepted by a scientific community

4 have acceptable error rates, which also requires intensive testing efforts.

Many of these computerised forensics tools are closed source, which means that only the vendor has access to the code, making it more difficult to apply the Daubert criteria. Consequently, assessing this type of tool will require greater effort on the part of

researchers and practitioners. According to Carrier (2002a), who studies these guidelines, open source tools are clearly better and more precise in terms of meeting the Daubert criteria than closed source tools. Among the many benefits of using open source digital forensics tools (Altheide et al., 2011) are the following:

- education: the tester can run the tool, investigate the options and outputs, and examine the code that produced the output to understand the logic behind the tool's operation
- portability and flexibility: the tester can choose where to use the tool, as well as how to use it without needing to ask for permission from the vendor
- license price: none
- ground truth: the tester can review and change the code.

Standards are developed to build up processes and methodologies intended to guarantee the compatibility of methods, products, and techniques. Standards address a scope of concerns, including different approaches to consensus that would assure product usefulness and similarity, encourages interoperability and bolster products' quality. Accordingly, standards are used in the context of this research to verify the quality requirements of open source digital forensics tools.

The ongoing challenge of assessing the quality of digital forensics tools has become critical, since there are now a large number of tools and the users of these tools are becoming more diverse in their backgrounds and technology interests. Therefore, in this paper we aim to assess the forensics tools Autopsy (2016), digital forensics framework (DFF, 2016), and the digital advanced response toolkit (DART) (DEFT, 2016) by applying the quality-in-use characteristics model (Alnanih, 2015) to obtain feedback from professionals and non-expert users using these tools as part of their regularly assigned duties.

The main contribution of this research paper is to validate the quality-in-use characteristics for the selected open source digital forensics tools used by both non-experts and professional people in order to easily determine the best tool based on the evidence, without compromising the data. In addition, our objective is to conduct a comparative study by measuring effectiveness, productivity, efficiency, error safety, and cognitive load for the tools mentioned above.

In this paper, we apply an existing quality-in-use model (Alnanih, 2015), inspired by the ISO/IEC 25010 (2011) and adapted to the context of the open source software (OSS) paradigm, to assess the selected open source digital forensics tools (see background section).

The paper is organised as follows: in Section 2, a comprehensive set of quality models for assessing OSS is introduced. However, there is little in the general literature on quality- in- use for these open source digital forensics tools. In Section 3, a summary of the literature on open source digital forensics tools assessment is provided. Section 4 explains our methodology for applying the quality-in-use model to three open source digital forensics tools. The experiments are detailed in Section 5 and a discussion is presented in Section 6. Section 7 introduces validation study through expert users. Conclusions and directions for future work are provided in Section 8.

2 Background

Standards frame the central building components for product development by setting up reliable conventions that can be widely understood and embraced. This enables more similarity and interoperability and also streamlines product development. Standards likewise make it less demanding to comprehend and compare competing products. As standards are more broadly embraced and connected in many markets, they additionally fuel global exchange. Thus, a set of International Organization for Standardization (ISO) standards, which are related to the objective of this research paper, are presented in the section below.

ISO/International Electrotechnical Commission (IEC) 25010:2011 defines "A quality-in-use model composed of five characteristics that relate to the outcome of interaction when a product is used in a particular context of use and a product quality model composed of eight characteristics that relate to static properties of software and dynamic properties of the computer system."

Per ISO (1998) 9241-11, usability is "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use." Usability is defined in (ISO, 2001) as "the capability of the software product to be understood, learned, used and attractive to the user when used under specified conditions." However, quality-in-use is defined as "the degree to which a product or system can be used by specific users to meet their needs to achieve specific goals with effectiveness, efficiency, freedom from risk and satisfaction in specific contexts of use" (ISO/IEC 25010: 2011; https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en, accessed 21 April 2016).

ISO/IEC 27037:2012 (2012) provides guidelines for specific activities in the handling of digital evidence, which are identification, collection, acquisition, and preservation of potential digital evidence that can be of evidential value. The limitation of this standard is that it ignores the software applications and only gives guidance for devices such as digital storage media, floppy disks, optical and magneto-optical disks, data devices with similar functions, mobile phones, personal digital assistants (PDAs), personal electronic devices (PEDs), memory cards, etc.

ISO/IEC 27041:2015 (2015a) provides guidance on mechanisms for ensuring that methods and processes used in the investigation of information security (IS) incidents are 'fit for purpose'. It encapsulates best practice on defining requirements, describing methods, and providing evidence that implementation of methods can be shown to satisfy requirements. It includes consideration of how vendor and third-party testing can be used to assist this assurance process. This document aims to:

- provide guidance on the capture and analysis of functional and non-functional requirements relating to an IS incident investigation
- give guidance on the use of validation as a means of assuring suitability of processes involved in the investigation
- provide guidance on assessing the levels of validation required and the evidence required from a validation exercise
- give guidance on how external testing and documentation can be incorporated in the validation process.

ISO/IEC 27042:2015 (2015b) provides guidance on the analysis and interpretation of digital evidence in a manner which addresses issues of continuity, validity, reproducibility, and repeatability. It encapsulates best practice for selection, design, and implementation of analytical processes and recording sufficient information to allow such processes to be subjected to independent scrutiny when required. It provides guidance on appropriate mechanisms for demonstrating proficiency and competence of the investigative team.

Analysis and interpretation of digital evidence can be a complex process. In some circumstances, there may be several methods which could be applied and members of the investigative team will be required to justify their selection of a particular process and show how it is equivalent to another process used by other investigators. In other circumstances, investigators may have to devise new methods for examining digital evidence which has not previously been considered and should be able to show that the method produced is 'fit for purpose'.

Quality-in-use is a more comprehensive concept than usability. Usability refers to the software itself, however, quality-in-use embodies the effects of using the software in a specified context, such as forensics.

According to Alnanih (2015) quality-in-use based on ISO/IEC 25010:2011(E) and ISO/IEC 25022:2014 "is the degree to which a product or system can be used by specific users to meet their needs to achieve specific goals with effectiveness, efficiency, freedom from risk, and satisfaction in specific contexts of use."

In this section, the quality-in-use characteristics are discussed, as an assessment of this type of software has received little attention in the literature. Also, Alnanih (2015) quality-in-use model is measured based on the software, system or tool's usage results, rather than the software properties themselves.

Alnanih's (2015) quality-in-use model and the characteristics of this model are considered in this work (Figure 1).

- Effectiveness: the number of actions required to complete the subtasks of each task in a specified context of use. It is measured in actions per subtask, and is calculated as follows:

$$\frac{Min\ \#correct\ actions}{\#correct\ actions + \#incorrect\ actions}$$

- Productivity: the number of correct actions performed in a specified context of use relative to the time taken by the user to complete the task. It is measured in actions per second, and is calculated as follows:

$$\frac{Min\ \#correct\ actions}{Time\ Period}$$

- Efficiency: the efficiency of the user in completing the task in a specified context of use. It is measured in actions per second, and is calculated as follows:

$$\frac{Effectiveness}{Time\ Period}$$

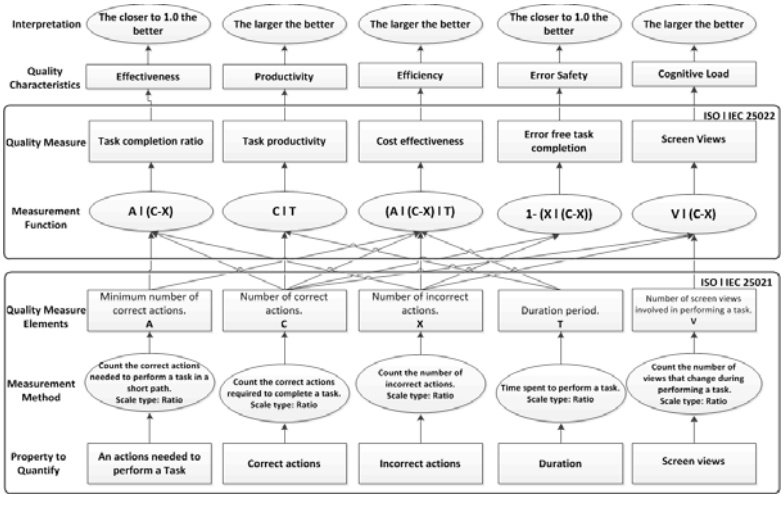
- Error safety (error prevention and recovery from error): the safety of the user, in terms of the number of errors committed in each action of each task performed in a specified context of use. It is measured in errors per action, and is calculated as follows:

$$1-\left(\frac{\#incorrect\ actions}{\#correct\ actions+\#incorrect\ actions}\right)$$

- Cognitive load: to perform a 'user task', the user may have to move from one screen to another, perhaps even back and forth. This will depend on three things: task complexity, screen size or form factor, and the way the designer has packed information on different screens. It is calculated as follows:

$$\frac{\#views}{\#correct\ actions+\#incorrect\ actions}$$

Figure 1 Alnanih's quality in use model



Notes: Alnanih (2015) quality-in-use model is measured based on the software, system or tool's usage results, rather than the software properties themselves. Figure 1 presents Alnanih's (2015) quality-in-use model and its characteristics.

Source: Alnanih (2015)

3 Literature review

There is a large amount of existing research in the area of forensics tools. In this section, we highlight issues related to how other researchers have assessed digital forensic tools and the ways in which our work is different than their work.

There are projects by the National Institute of Standards and Technology (NIST) under the computer forensics tool testing (CFTT) project (NIST, 2017b) with different goals. The goal of this project is to establish a methodology for testing computer forensic software tools by development. However, results of these tests are dependent on people such as toolmakers to improve tools, users to make informed choices about acquiring and using computer forensics tools, and interested parties to understand the tools' capabilities. However, in our work we considered the professional users of forensics tools in order to come up with results to help to improve and assess their domain.

In Pan and Batten, (2009) the authors presented a robust correctness testing method based on the questionnaire methods i that assesses accuracy and precision, but only for digital forensic tools. The focus of their work is on developing a simple way of helping an expert witness in digital forensics to implement some standard software tests in order to be ready with answers to questions. Their methods show how much the functionality of a tool differs from the user's expectations. However, the quality of forensics software tools depends on many parameters such as effectiveness and cognitive load, which we considered in our work.

In an effective work, authors in Flandrin et al. (2014) focused on reviewing papers in order to outline the evaluation and validation methodologies. They found that some of these methodologies, such as Carrier's (2002b) abstraction layers model, are too complex to be used by digital forensics investigators and others do not cover all aspects of the tools (Lempereur et al., 2006). This shows that none of them has yet implemented a model to enable automation of the validation process and indicates that there is a need to find a model that would enable the testing to be performed automatically. Our quality model could be the solution for this gap in the other research work.

The potential error rate generated by the technique needs to be known because it represents the level of uncertainty in the scientific method. This statement around *error rates* is virtually unaddressed at the current time. Only Lyle (2010) attempted to identify basic issues of defining *error rates* for digital forensics tools. Lyle (2010) observed that the error rate needs to be linked with the condition under which they have been produced. These results support our quality-in-use model, which has a complete model, consisting of a complete set of characteristics including the error rate and is adapted to the different context of use. In the following subsections, we highlight the OSS in terms of quality and tools.

3.1 Quality models for OSS

Over the years, many open source assessment models have been introduced and used in various fields. In this section, we introduce a comprehensive set of existing quality models and describe their application to OSS.

In 2003, Capgemini developed the open source maturity model (OSMM) (Duijnhouwer and Widdows, 2003). OSMM is a tool that uses the maturity of the product to compare different software products in order to choose the one that best fits the goals of the organisation. However, OSMM is available under a non-free license agreement, but one that provides for authorised distribution.

Houaich and Belaissaoui (2015) conducted a survey of 200 Moroccan SMEs to identify their needs, as well as their knowledge of open source technology and their abilities to adopt it. For the purposes of matching the various SMEs with the appropriate open source technology, they used the OSMM product category to design a new assessment model: E-OSSEM. Using this model, they could determine the open source technology that best fit each SME.

Akbari and Peikar (2014) studied the free/open source (FLOSS) GIS tools available in the web mapping and spatial database environment. This research shows that UMN MapServer is a totally mature OSS that is compatible with other closed source products, and that PostGIS is highly competitive among closed source products, especially in terms of 3D function features.

In 2007, the open business readiness rating (Open BRR) was sponsored by Carnegie Mellon West Center for Open Source Investigation, CodeZoo, SpikeSource, and Intel (Wasserman et al., 2006). It is designed to help organisations find the most suitable OSS for their needs. In order to speed up the evaluation process, this model uses a systematic approach consisting of four stages. Also, it uses qualitative measures, and so a weighting is assigned to each metric, which is then used to arrive at a score for each category by computing the individual metric scores. However, they did not focus on the quantitative measure, which is the major part of the quality-in-use model.

Das and Wasserman (2011), created a web-based application prototype that guides users to find their appropriate OSS using Open BRR. In their experiment, the user interface (UI) was changed slightly to follow the BRR framework more closely.

Groven et al. (2010) applied the three-security metrics available in open BRR to measure the level of security of Asterisk, a FLOSS implementation framework, for building communications applications. Groven et al. (2010) applied the nine security indicators, along with 30-40 of the security metrics available in QualOSS, to measure the level of security of asterisk.

Samoladas et al. (2008) developed the software quality observatory for OSS (SQO-OSS). This is a platform that performs detailed automatic quality evaluations of OSS, including the source code, in order to help users decide whether or not the target software is suitable for them to use. The model is hierarchical and assesses source code and community processes.

Groot et al. (2006) provided a quality evaluation of the KDE project to allow engineers to select appropriate improvements to be appended to the original solution. The authors concluded that the "SQO-OSS system will ultimately aid OSS developers to write [sic] better software and enable potential users to make better-informed choices."

Soto and Ciolkowski (2009a) developed a model called quality of OSS (QualOSS), which focuses on evolvability and robustness in the evaluation of OSS. This model is made up of two categories of quality characteristics: product-related and community-related, but does not include any quality-in-use characteristics in either category.

Soto and Ciolkowski (2009b) have applied the QualOSS model on up to 20 different FLOSS products, including successful and unsuccessful software. The authors' goal was to compare the results to verify the ability of QualOSS to differentiate between successful and unsuccessful software. Soto et al. (2009) presented their evaluation procedures, along with recommendations and application lessons learned in some OSS projects.

Aversano and Tortorella (2010) designed the evaluation framework for free/open source projects (EFFORT). Its purpose is to provide a framework for evaluating the quality and functionality of the target OSS. In the same year, the authors presented a customised version of EFFORT for evaluating enterprise resource planning (ERP) OSS systems. The customised framework was applied to the evaluation and comparison of five ERP OSS systems (Aversano and Tortorella, 2011). Similarly, Aversano and Tortorella (2011) customised EFFORT to explicitly fit FLOSS for a customer relationship management (CRM) system. EFFORT was applied to four of the most common CRM systems. The results obtained were positive for product quality and product attractiveness, but less satisfying for community trustworthiness.

According to Alnanih (2015) quality-in-use based on ISO/IEC 25010:2011(E) and ISO/IEC 25022:2014 is measured based on the software, system or tool's usage results, rather than the software properties themselves.

Since ISO 25010 characteristics constitute the foundation for software and systems quality requirements (ISO, 2007) and their measurements (ISO, 2011), this research adopts this standard to conduct all experiments for the quality requirements of the tools above. Quality requirements describe the desired quality of a system.

Table 1 shows the characteristics of the models discussed above compared with those in ISO/IEC 25010. It is clear that quality-in-use assessment for this type of software is lacking in the literature.

Table 1 Comparison between ISO/IEC 25010 and OSMM, open BRR, QualOSS, SQO-OSS, and EFFORT

ISO 25010	*Quality characteristics*	*OSMM*	*Open BRR*	*Qual OOS*	*SQO-OSS*	*Effort model*
Product quality	Functional suitability	√	√	√		√
	Reliability	√		√	√	√
	Performance efficiency	√	√	√		√
	Operability	√	√	√		
	Security	√	√	√	√	
	Compatibility	√		√		
	Maintainability	√	√	√	√	√
	Transferability	√		√		√
Quality in use	Effectiveness				√	
	Efficiency					
	Satisfaction					
	Safety					
	Usability	√	√			√

3.2 Open source digital forensics tools

Raza et al. (2011) presented an interesting literature review, starting with Bodker et al. (2007), who highlights the fact that OSS developers need to have a full understanding of user demands, as well as the motivation and determination to address them. It ended with Zaharias and Poylymenakou (2009), who, in addition, consider usability questionnaires

as a fast and cost-effective way to collect user feedback. Raza et al. (2011) analyse industry users' perception of usability factors and of OSS usability from the industrial perspective, citing the importance of understandability, learnability, operability, and attractiveness. They conducted their study on a dataset of 105 industry users. The results of their empirical investigation indicate the significance of the key factors in OSS usability.

In Manson et al. (2007) assessed Sleuth Kit in terms of ease of use, robust functionality, and the reliability and verifiability of its results. They compared Sleuth Kit to EnCase and FTK to determine whether evidentiary data are identified by all three products. They concluded, "Since the aim of performing computer forensics is not to have a duel between two competing technologies, but to prosecute a person for the crimes they have been accused of, therefore it is important that both open and closed source programs work together to validate each other's results, so that justice can be done to those who deserve it" (Manson et al., 2007). They also stated, "This means that closed source users must have an open mind and must try other tools, preferably open source tools, to validate their results" (Manson et al., 2007).

Bennett and Stephens (2008) reviewed the usability of the autopsy forensic browser. Two expert-based usability review techniques were used: cognitive walkthrough and heuristic evaluation. The results of the review indicate that there are many areas where usability could be improved, and these are classified into eight overlapping areas. The authors concluded with a proposal for future work, which would involve affecting the changes suggested and retesting both the current system and the replacement system with a user-based evaluation.

According to Hibshi et al. (2011), "the usability aspect of forensics tools is examined through interviews and surveys designed to obtain feedback from professionals using these tools as part of their regularly assigned duties. The study results highlight several usability issues that need to be taken into consideration when designing and implementing digital forensics tools."

Many usability issues raised in the open source literature remain unresolved (Nurse et al., 2011). In addition, little attention has been paid to quality-in-use assessment for open source digital forensics tools.

4 Methodology

Three examples of the most popular open source digital forensics tools as reported by Kaur et al. (2016) are Autopsy, the DFF, and the DART (DEFT). These tools are selected for conducting this research, because they are easy to use, fast, and the cost-effectiveness is high, since they are free in addition to being the most popular tools within the OSS community.

According to Kaur et al. (2016) "Autopsy tool helps thousands of users around the world and has community-based e-mail lists and forums. Law enforcement, military and corporate examiners use autopsy to examine what appears on a computer." Autopsy is a digital forensics platform. It is based on a GUI program that allows you automatically to investigate what appears on computers and smartphones.

Kaur et al. (2016) stated that "Digital Forensics Framework is an Open Source computer forensics software. It is used both by professional and non-experts in order to quickly and easily gather, conserve and admit digital evidence without compromising systems and data."

According to Digital Forensics Pentest Linux Distributions (2016) "DEFT is paired with DART (an acronym for Digital Advanced Response Toolkit), a Forensics System which can be run on Windows and contains the best tools for Forensics and Incident Response."

In this research, our participant sample was made up of two groups of users: non-expert and professional. Non-expert participants are 30 senior undergraduate students registered in a Software Engineering course (1,411,366) during spring, 2016 at University of Sharjah (UOS) in the UAE. The participants belong to junior stereotype and to the 22 to 32-year-old age group. These users have a good understanding of technology such as using smartphone and computer platforms, however; this group is also usually among the younger students who may have little to no experience in the real domain of the forensics.

For the professional participants, we did the test with nine participants. However, during usability testing, Nielsen and Landauer (1993) found on average $p = 0.31$ for the set of projects studied. Based on that, five users would be expected to find 85% of the usability problems available for discovery in that test iteration. Similarly, Virzi (1992) created a model based on other usability projects, finding p to be between 0.32 and 0.42. Therefore, 80% of the usability problems in a test could be detected with four or five participants. Professional participants are nine expert users who belong to a senior stereotype of PhD holders for three age groups. These users show average to little resistance to using new technology and applications on their smartphones and computer platforms. They have worked in the digital forensics field for at least 1–3 years. Moreover, a usability questionnaire was given to every participant.

As a part of the study, we adopted the following process. First, thirty graduate students were selected for the study and divided at random into three groups (A, B, and C), with each group consisting of ten students. Second, each group received one of the forensics tools randomly. Usability testing for each tool was conducted twice for each participant, in each group. For example, Group A conducted the test once with the group B tool and once with the group C tool. Table 2 demonstrates the pattern of non-expert and expert participants using the three forensics tools.

Table 2 Three forensics tools

Forensics tool	*Participant*	
	Non-expert	*Expert*
One tool among autopsy, DFF, DART	A	Usability questionnaire
Another tool among autopsy, DFF, DART	B	Usability questionnaire
The remaining tool from autopsy, DFF, DART	C	Usability questionnaire

The independent variables in the study were the base measurements (A, C, X, T, and V) of each tool. The dependent variables were the quality-in-use characteristics (Table 3).

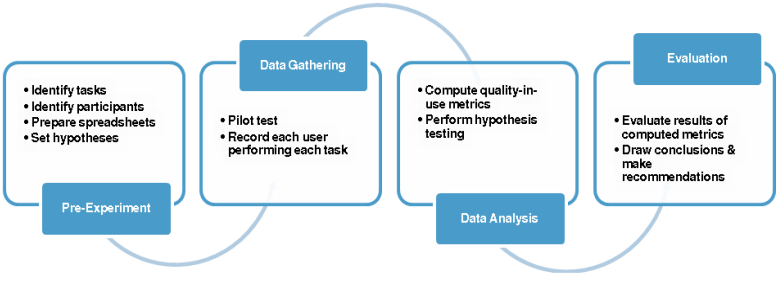
Table 3 The dependent and independent variables

Independent variables		*Dependent variable*
A	# of minimum actions	Effectiveness
C	# correct actions	Productivity
X	# incorrect actions	Efficiency
T	Time duration	Error safety
V	# of views	Cognitive load

4.1 Quality-in-use application

In this section, we conducted an experimental evaluation in which we compared the three software products based on the stages shown in Figure 2.

Figure 2 Our quality in use methodology (see online version for colours)



Note: It shows the stages used to compare the three software products, which is part of this research experimental evaluation.

4.2 Pre-experiment

At this stage, we organised the various parts of the experiment to ensure that we would be fully prepared and would not be interrupted during the experiment. We selected ten different tasks which are available on the three software products and shared these between the three groups. The selected tasks are the following:

1 recover email messages

2 hash lookup and mapping

3 retrieve exit metadata

4 view file hex code

5 extract files

6 view document

7 analyse registry

8 extract web browser history

9 view/play media

10 recover data files.

The above tasks were performed by each of the 'expert' groups of participants to categorise the users, we asked them directly before conducting the experiments. In addition, during the experiment, the functionalities of the tools were discussed with all users individually and then users' answers across these functionalities helped to give a fair idea of their expertise. Users who are using forensics tools and have previous knowledge of our open source forensics tools' functionalities are considered expert users.

We prepared spreadsheets to organise the data gathered by experiment users containing all the attributes that needed to be gathered for each task.

In our experiment, we relied on the data to refute or support our hypotheses, which, as we will explain further, allowed us to compare the three software products. We first defined a null hypothesis which we the tried to refute and an alternative hypothesis that is automatically supported if the null hypothesis is refuted.

a Null hypothesis: there is no relationship between the two measured phenomena (a general statement).

b Alternative hypothesis: the opposite of the null hypothesis.

Below is an example of each of the two hypotheses:

- Effectiveness null-HYP: "There is no significant difference between the effectiveness of autopsy and the effectiveness of DART."
- Effectiveness alt-HYP: "There is a significant difference between the effectiveness of autopsy and the effectiveness of DFF."

4.3 Data gathering

This stage constitutes the first phase of the actual experiment, where we gathered the data used to compare the three software products. In order to ensure that we were ready to conduct the actual experiment, we first conducted some pilot tests in which we performed the steps of the actual experiment. We took a screenshot on our laptop while each participant performed each of the ten tasks on each software product, and altered the order in which the software was used by the participants, in order to minimise the amount of learning that could take place. Then, for each task, we calculated the number of correct and incorrect actions that the user performed to complete the task. Also, we calculated the execution time of each task and counted the number of screenshots taken. We reported all these data in the spreadsheets.

4.4 Data analysis

At this stage, we analysed the data gathered, in order to compare the values obtained for each software product. We computed the metrics using the formulas mentioned above using the data in the spreadsheets. We then computed the student t-test using an excel tool. Because it is useful to compare mean values, we computed the average value for all the tasks in each metric for each user, for each software product. Then, we calculated the t-Test using the average obtained for each product.

4.5 Evaluation

In hypothesis testing, if the P-value of a factor provided in the t-test results is greater than alpha (0.05), then we cannot refute the null hypothesis for that factor. Otherwise, the alternative hypothesis is supported, and we should rely on the mean value to compare the two products.

5 Experiment results

The experiment was conducted in a well-ordered way.

a We tested three subjects who were expert users on open source digital forensics tools (the objects in this scenario) on three designated machines, all running the same OS.

b The response from the users was good overall. There was minimal confusion on the part of the users throughout their exposure to the software, except at the start, when they struggled somewhat to find the task that was the target of the test. After that, they went through the testing process quite quickly.

c The behaviour of the expert users during testing was quite lively. Most did not hesitate to ask questions about the software and why they were being tested in that particular way. Other users shared their ideas and knowledge with us about some of the software functions and why they were designed to work the way they do. One of the expert users we tested, who is retired, gave us a clear idea of how drastically things had changed in recent decades, as such technology was unavailable back in the early 70s. This gave us a clearer picture of how far technology has come and how digital forensics tools have made things easier for the many users involved in the forensics field. Overall, the testing process went well for us all as a group: the testers, and the expert users and us.

d This software has a large number of characteristics, some of which are extremely positive, while others are rather more negative. The reason for the negative ones could be a lack of software updates provided, as this is more of the open source version of the software itself. Some of the appealing characteristics of this software include its vast library of tasks that range from something as simple as a media player or a file compression application to a much more complex feature, yet all are equally valuable, such as a program that can track down the user's browser history

and what files they have downloaded and deleted. In the world of forensics, a few characteristics can make a world of difference, and this software has packed everything a user could ever want into one massive package. On the negative side (and this is also true for DART, unfortunately), some of the software cannot be executed on certain platforms running particular OS versions. This can affect a large variety of users with limited to no Internet access. In our case, Mailcure, a program that is used to track down and recover email sent from the host machine, would not execute on 2/3 of the machines that we used for testing. To counter this problem, we had to download the program itself from outside sources. Another negative characteristic of DART is that some of its software is outdated. This problem could be solved if the program had the ability to upgrade when connected to the internet.

e The independent variables collected from the users during the experiment were the following:

- Time: the time it took for the user to complete the action specified by the tester. We also recorded the overall time taken (in seconds) to complete the tasks, to give us a general idea of how fast the users were at using these open source digital forensics software products.
- The number of actions: the total number of actions taken by the user to complete the task assigned by the tester. This value contains both the correct and incorrect number of actions taken by the user.
- The number of incorrect actions: the incorrect actions were recorded by the tester alongside the total number of actions, and used to determine the correct number of actions.
- The number of views: the number of windows users have for viewing the task they are asked to perform. This is, of course, a set number of views for each program, because all the users have to go through the same number of windows to complete their actions.

5.1 Autopsy as an illustration

The calculations for effectiveness, as shown in Table 4, depend on the minimum number of correct actions divided by the sum of correct and incorrect actions. This yields a perfect result of 1 only if the number of incorrect actions is 0 and the number of correct actions is equal to the minimum number of actions.

Each of the tasks below has its own effectiveness value. Also, each user has his own row of effectiveness values for each task. The average of this row is taken and the average of the effectiveness for a single user who has finished all the tasks is then calculated. For the total average, we have averaged all the final values in each row to obtain the final effectiveness AVG, after observations have been made and the data analysed. Our team concluded that this value can certainly be improved, as it is some distance from 1. The closer to 1 it is, the better (Figure 1).

The calculations for productivity, efficiency, error safety and cognitive load are shown in Tables 5, 6, 7, and 8.

Table 4 Autopsy effectiveness [A | (C + X)]

Users	*Recover email messages*	*Hash lookup and mapping*	*Retrieve exif metadata*	*View file hex code*	*Extract files*	*View document*	*Analyse registry*	*Extract web browser history*	*View/play media*	*Recover data files*	*Average*
User1	0.16	0.25	0.5	0.42	0.28	0.28	1	0.5	0.33	0.27	0.40
User2	0.16	0.15	0.33	0.6	0.4	1	0.55	0.4	0.5	0.33	0.44
User3	0.25	1	0.6	0.42	0.28	0.5	0.45	0.28	0.4	0.5	0.47
User4	0.09	0.125	0.15	1	0.66	0.66	0.38	0.33	0.66	0.2	0.42
User5	0.33	0.66	1	0.75	0.5	0.5	0.35	1	0.5	0.3	0.59
User6	0.09	0.28	1	1.5	1	1	0.625	0.28	0.66	0.33	0.67
User7	0.33	0.5	0.75	1	0.66	0.22	0.55	0.5	0.22	1	0.57
User8	0.25	0.18	0.6	0.5	0.33	0.4	0.62	0.33	0.66	0.42	0.43
User9	1	0.18	1	0.25	0.16	0.18	0.55	0.22	0.4	0.33	0.42
Total average											0.49

Table 5 Autopsy productivity [C | T]

Users	*Recover email messages*	*Hash lookup and mapping*	*Retrieve exif metadata*	*View file hex code*	*Extract files*	*View document*	*Analyse registry*	*Extract web browser history*	*View/play media*	*Recover data files*	*Average*
User1	0.15	0.2	0.16	0.37	0.48	0.21	0.33	0.2	0.36	0.26	0.27
User2	0.08	0.3	0.2	0.5	0.27	0.13	0.35	0.31	0.22	0.26	0.26
User3	0.12	0.03	0.13	0.15	0.2	0.18	0.33	0.41	0.22	0.26	0.20
User4	0.08	0.16	0.2	0.2	0.23	0.12	0.16	0.42	0.11	0.35	0.20
User5	0.07	0.11	0.17	0.5	0.44	0.36	0.33	0.10	1	0.19	0.33
User6	0.05	0.15	0.3	0.6	0.13	0.4	0.5	0.31	0.05	0.17	0.26
User7	0.06	0.4	0.66	0.33	0.35	4.5	0.3	0.43	6.9	0.06	1.403
User8	0.14	0.0	0.06	0.06	0.08	0.16	0.08	0.3	0.11	0.1	0.12
User9	0.02	0.47	0.33	0.39	0.45	0.70	0.2	0.2	0.62	0.21	0.36
Total average											0.38

Table 6 Autopsy efficiency [A | (C + X) | T]

Users	*Recover email messages*	*Hash lookup and mapping*	*Retrieve exif metadata*	*View file hex code*	*Extract files*	*View document*	*Analyse registry*	*Extract web browser history*	*View/play media*	*Recover data files*	*Average*
User1	0.005	0.007	0.013	0.026	0.019	0.010	0.067	0.025	0.02	0.009	0.0204
User2	0.002	0.005	0.008	0.06	0.022	0.066	0.027	0.025	0.027	0.012	0.0258
User3	0.008	0.018	0.020	0.013	0.009	0.023	0.016	0.019	0.018	0.022	0.0170
User4	0.001	0.002	0.002	0.066	0.051	0.026	0.007	0.023	0.025	0.007	0.0214
User5	0.008	0.02	0.058	0.125	0.055	0.0454	0.009	0.052	0.125	0.006	0.051
User6	0.0007	0.007	0.1	0.3	0.066	0.2	0.052	0.012	0.018	0.007	0.076
User7	0.006	0.05	0.125	0.055	0.078	0.111	0.018	0.054	0.170	0.02	0.069
User8	0.009	0.001	0.01	0.008	0.005	0.013	0.008	0.016	0.024	0.011	0.011
User9	0.022	0.010	0.111	0.010	0.008	0.016	0.012	0.004	0.05	0.008	0.025
Total average											0.035

Table 7 Error safety [1 – (X | (C + X)]

Users	*Recover email messages*	*Hash lookup and mapping*	*Retrieve exif metadata*	*View file hex code*	*Extract files*	*View document*	*Analyse registry*	*Extract web browser history*	*View/play media*	*Recover data files*	*Average*
User1	0.83	0.875	1	1	1	0.85	1	1	1	0.72	0.92
User2	0.83	0.69	0.88	0.71	1	1	0.77	1	1	0.77	0.86
User3	1	1	0.8	0.83	0.85	1	0.81	0.85	1	1	0.91
User4	0.72	0.62	0.65	1	1	1	0.61	1	1	0.66	0.82
User5	1	1	1	1	1	1	0.85	1	1	0.9	0.97
User6	0.63	0.85	1	1	1	1	0.75	1	0.66	0.88	0.87
User7	1	1	1	1	1	1	1	1	1	1	1
User8	1	0.54	0.8	0.8	0.83	1	0.75	1	1	0.71	0.84
User9	1	0.72	1	0.69	0.75	0.72	1	1	1	1	0.88
Total average											0.90

Table 8 Cognitive load [V | (C + X)]

Users	*Recover email messages*	*Hash lookup and mapping*	*Retrieve exif metadata*	*View file hex code*	*Extract files*	*View document*	*Analyse registry*	*Extract web browser history*	*View/play media*	*Recover data files*	*Average*
User1	0.66	0.25	0.16	0.16	0.28	0.14	0.4	0.5	0.16	0.18	0.29
User2	0.66	0.23	0.11	0.14	0.4	0.5	0.22	0.2	0.25	0.22	0.29
User3	0.25	1	0.4	0.33	0.71	0.5	0.72	0.14	0.8	0.66	0.55
User4	0.45	0.25	0.15	0.33	0.33	0.66	0.23	0.5	0.33	0.2	0.345
User5	0.66	1	0.66	0.33	1	0.5	0.57	0.5	0.5	0.6	0.633
User6	0.27	0.14	0.33	1	0.5	0.5	0.5	0.57	0.33	0.22	0.43
User7	0.33	0.5	0.5	0.33	0.66	0.55	0.55	0.5	0.55	0.33	0.483
User8	1	0.36	0.4	0.6	0.5	1	0.62	0.83	1.33	0.28	0.69
User9	1	0.18	0.66	0.15	0.25	0.18	0.55	0.55	0.4	0.55	0.450
Total average											0.464

6 Discussions

To represent our findings, we calculated the mean of all the averages for each metric. We did so for all three software products. In addition, we were able to support and refute the hypotheses that we defined at the beginning of our experiment.

- Effectiveness: the P-values (0.02) and (0.004) Tables 9 and 10 obtained are smaller than the value of alpha (0.05), which means that effectiveness null-HYP is refuted. The value of mean was considered to compare the three software products. The mean value for effectiveness in DART (0.59) is greater than the mean value for effectiveness in autopsy (0.49) as shown in Table 9. Similarly, the mean value for effectiveness in DFF (0.62) is greater than the mean value for effectiveness in autopsy (0.49) as shown in Table 10. In other words, there is a difference in the effectiveness values for autopsy, DART, and DFF. We conclude that DFF is more effective than autopsy and DART.
- Productivity: the P-values obtained (0.55) and (0.53) as shown in Tables 11 and 12 are greater than the value of alpha (0.05), which means that productivity null-HYP is not refuted. In other words, we conclude that there is no significant difference in the productivity values for autopsy, DART, and DFF.
- Efficiency: the P-values obtained (0.20) and (0.25) as shown in Tables 13 and 14 are greater than the value of alpha (0.05), which means that efficiency null-HYP is not refuted. In other words, we conclude that there is no significant difference in the efficiency values for autopsy, DART, and DFF suite.
- Error safety: the P-values obtained (0.02) and (0.04) are smaller as shown in Tables 15 and 16 than the value of alpha (0.05), which means that error safety null-HYP is refuted and we rely on the mean to compare the three software products. The mean value for error safety in autopsy (0.90) is greater than the mean value for error safety in DART (0.67), as shown in Table 15. Similarly, the mean value for error safety in autopsy (0.90) is greater than the mean value for error safety in DFF (0.76) as shown in Table 16. In other words, there is a difference in error safety values for Autopsy, DART, and DFF. We conclude that Autopsy has better error safety than DFF and DART.
- Cognitive load: the P-value obtained (5.9459E-5) as shown in Table 17 is smaller than the value of alpha (0.05), which means that cognitive load null-HYP is refuted and we rely on the mean to compare the two software products. The mean value for cognitive load in DART (0.95) as shown in Table 17 is greater than the mean value for cognitive load in autopsy (0.46) as shown in Table 17. However, there is no significant difference in the cognitive load values for autopsy and DFF since the P-value (0.84) is greater than alpha (0.05) as shown in Table 18.

Table 9 The T-test result for effectiveness in DART vs. autopsy

DART vs. autopsy		
Effectiveness (DART)	*Effectiveness (autopsy)*	
t-test: paired two sample for means		
	Variable 1	*Variable 2*
Mean	0.593987583	0.494538702
Variance	0.01066458	0.009231416
Observations	9	9
Pearson correlation	0.36608129	
Hypothesised mean difference	0	
Df	8	
t stat	2.654581262	
P(T <= t) one-tail	0.014524305	
t critical one-tail	1.859548038	
P(T <= t) two-tail	0.029048609	
t critical two-tail	2.306004135	

Note: 2-tail 0.029 < 0.05 so we reject null hypothesis and investigate alternative.

Table 10 The T-test result for effectiveness in DFF vs. autopsy

DFF vs. autopsy		
Effectiveness(DFF)	*Effectiveness (autopsy)*	
t-test: paired two sample for means		
	Variable 1	*Variable 2*
Mean	0.629305945	0.494538702
Variance	0.011242066	0.009231416
Observations	9	9
Pearson correlation	0.478487307	
Hypothesised mean difference	0	
Df	8	
t stat	3.904058637	
P(T <= t) one-tail	0.002259361	
t critical one-tail	1.859548038	
P(T <= t) two-tail	0.004518722	
t critical two-tail	2.306004135	

Note: 2-tail 0.004 < 0.05 so we reject null hypothesis and investigate alternative.

Table 11 The T-test result for productivity in DART vs. autopsy

DART vs. autopsy		
Productivity (DART)	*Productivity (autopsy)*	
t-test: paired two sample for means		
	Variable 1	*Variable 2*
Mean	0.46219551	0.382625373
Variance	0.000910209	0.15155464
Observations	9	9
Pearson correlation	–0.042489738	
Hypothesised mean difference	0	
Df	8	
t stat	0.609353908	
P(T <= t) one-tail	0.279596844	
t critical one-tail	1.859548038	
P(T <= t) two-tail	0.559193688	
t critical two-tail	2.306004135	

Note: 2-tail 0.55 > 0.05 so we accept the null hypothesis.

Table 12 The T-test result for productivity in DFF vs. autopsy

DFF vs. autopsy		
Productivity (DFF)	*Productivity (autopsy)*	
t-test: paired two sample for means		
	Variable 1	*Variable 2*
Mean	0.299015866	0.382625373
Variance	0.001419246	0.15155464
Observations	9	9
Pearson correlation	0.144233839	
Hypothesised mean difference	0	
Df	8	
t stat	–0.650366792	
P(T <= t) one-tail	0.266841617	
t critical one-tail	1.859548038	
P(T <= t) two-tail	0.533683235	
t critical two-tail	2.306004135	

Note: 2-tail 0.533 > 0.05 so we accept the null hypothesis.

Table 13 The T-test result for efficiency in DART vs. autopsy

DART vs. autopsy		
Efficiency (DART)	*Efficiency (autopsy)*	
t-test: paired two sample for means		
	Variable 1	*Variable 2*
Mean	0.226562164	0.035348146
Variance	0.165625404	0.000577262
Observations	9	9
Pearson correlation	#N/A	
Hypothesised mean difference	0	
Df	8	
t stat	1.380638005	
P(T <= t) one-tail	0.102367012	
t critical one-tail	1.859548038	
P(T <= t) two-tail	0.204734025	
t critical two-tail	2.306004135	

Note: 2-tail 0.20 > 0.05 so we accept the null hypothesis.

Table 14 The T-test result for efficiency in DFF vs. autopsy

DFF vs. autopsy		
Efficiency (DFF)	*Efficiency (autopsy)*	
t-test: paired two sample for means		
	Variable 1	*Variable 2*
Mean	0.025775714	0.035348146
Variance	2.93594E-05	0.000577262
Observations	9	9
Pearson correlation	0.246480345	
Hypothesised mean difference	0	
Df	8	
t stat	–1.233006889	
P(T <= t) one-tail	0.126286755	
t critical one-tail	1.859548038	
P(T <= t) two-tail	0.252573511	
t critical two-tail	2.306004135	

Note: 2-tail 0.25 > 0.05 so we accept the null hypothesis.

Table 15 The T-test result for error safety in DART vs. autopsy

DART vs. autopsy		
Error safety (DART)	*Error safety (autopsy)*	
t-test: paired two sample for means		
	Variable 1	*Variable 2*
Mean	0.678701741	0.903593074
Variance	0.062585872	0.003309793
Observations	9	9
Pearson correlation	0.17080189	
Hypothesised mean difference	0	
Df	8	
t stat	–2.73214312	
P(T <= t) one-tail	0.012880139	
t critical one-tail	1.859548038	
P(T <= t) two-tail	0.025760279	
t critical two-tail	2.306004135	

Note: 2-tail 0.025 < 0.05 so we reject null hypothesis and investigate alternative.

Table 16 The T-test result for error safety in DFF vs. autopsy

DFF vs. autopsy		
Error safety (DFF)	*Error safety (autopsy)*	
t-test: paired two sample for means		
	Variable 1	*Variable 2*
Mean	0.769897579	0.903593074
Variance	0.019991634	0.003309793
Observations	9	9
Pearson correlation	–0.2364933	
Hypothesised mean difference	0	
Df	8	
t stat	–2.43423257	
P(T <= t) one-tail	0.020465546	
t critical one-tail	1.859548038	
P(T <= t) two-tail	0.040931092	
t critical two-tail	2.306004135	

Note: 2-tail 0.04 < 0.05 so we reject null hypothesis and investigate alternative.

Table 17 The T-test result for load cognitive in DART vs. autopsy

DART vs. autopsy group		
Load cognitive (DART)	*Load cognitive (autopsy)*	
t-test: paired two sample for means		
	Variable 1	*Variable 2*
Mean	0.955793651	0.464982949
Variance	0.006332596	0.020399411
Observations	9	9
Pearson correlation	–0.448329566	
Hypothesised mean difference	0	
Df	8	
t stat	7.66276173	
P(T <= t) one-tail	2.97297E-05	
t critical one-tail	1.859548038	
P(T <= t) two-tail	5.94595E-05	
t critical two-tail	2.306004135	

Note: 2-tail 0.0000059 < 0.05 so we reject null hypothesis and investigate alternative.

Table 18 The T-test result for load cognitive in DFF vs. autopsy

DFF vs. autopsy		
Load cognitive (DFF)	*Load cognitive (autopsy)*	
t-test: paired two sample for means		
	Variable 1	*Variable 2*
Mean	0.455190371	0.464982949
Variance	0.021503922	0.020399411
Observations	9	9
Pearson correlation	0.477900435	
Hypothesised mean difference	0	
Df	8	
t stat	–0.198585858	
P(T <= t) one-tail	0.423770291	
t critical one-tail	1.859548038	
P(T <= t) two-tail	0.847540581	
t critical two-tail	2.306004135	

Note: 2-tail 0.84 > 0.05 so we accept the null hypothesis.

6.1 DART as an illustration

From the results obtained in this experiment, we know that DART has low efficiency and is lacking in terms of user productivity. To improve this, the developers need to figure out *what* these problems are and *where* they occur. For example, low efficiency can be

related to the fact that most of the programs are outdated and will have to be replaced with newer versions. Moreover, these versions will certainly have increased computer OS complexity and hardware, which will, without a doubt, limit the compatibility of these programs. The newer versions will nevertheless increase the user's efficiency in completing the task in a specified context of use.

Productivity is another factor that can heavily influence the use and purpose of a program. As DART failed to achieve the required value in the t-test, it cannot be considered productive. Productivity is something that really should not be a problem in designing an interactive UI for open software, mainly because users can pitch in and help improve it whenever they wish to. This will increase the productivity of DART by increasing the number of correct actions performed in a specified context of use, relative to the time taken by the user to complete the task. Finally, DART incorporates less error safety than other software. This can be a problem in any UI, whether open or closed source and can affect a program in a wide variety of ways. For example, if you are working with safety-critical software, such as DART, in this case, while you are in the middle of recovering crucial data, errors could occur that might erase the user's progress. This is an important issue that should be addressed by UI designers.

7 Validation study

We did the test with nine professional participants. However, Nielsen and Landauer (1993) found on average $p = 0.31$ for the set of studied projects. Based on that, five users would be expected to find 85% of the usability problems available for discovery in that test iteration. Similarly, Virzi (1992) created a model based on other usability projects, finding p between 0.32 and 0.42. Therefore, 80% of the usability problems in a test could be detected with 4 or 5 participants.

Professional participants are nine expert users who belong to a senior stereotype of Ph.D. holders for three age groups. These users show average to little resistance to using new technology and applications on their smartphones and computers platform. They have worked in the digital forensics field for at least one to three years. Figure 3 summarises the background of expert users.

The expert users' results show no difference when compared with non-expert users. Tables 19–20 explore in more detail their opinion on each OSS: for example, DART has low efficiency and is lacking in terms of user productivity according to the answers of the expert users. One expert user said that "Doing forensic analysis of evidence files is more effective using Autopsy as compared to DFF and DART libraries. Because of its user-friendly GUI, Autopsy helps to be more productive and efficient recovery of required information from the digital devices." In addition, another expert user mentioned that "I consider Autopsy to be easier to use, efficient in showing required results, faster in performing a task and easy to manage account as well. My overall comment on all three-software mentioned above is that I would prefer using Autopsy for my coming projects since it saves time, which helps me in concentrating more on the job in hand instead of spending time learning the tool itself." Most of them have confirmed that they prefer using OSS. However, this software lacks mature documentation.

Figure 3 Expert users' background (see online version for colours)

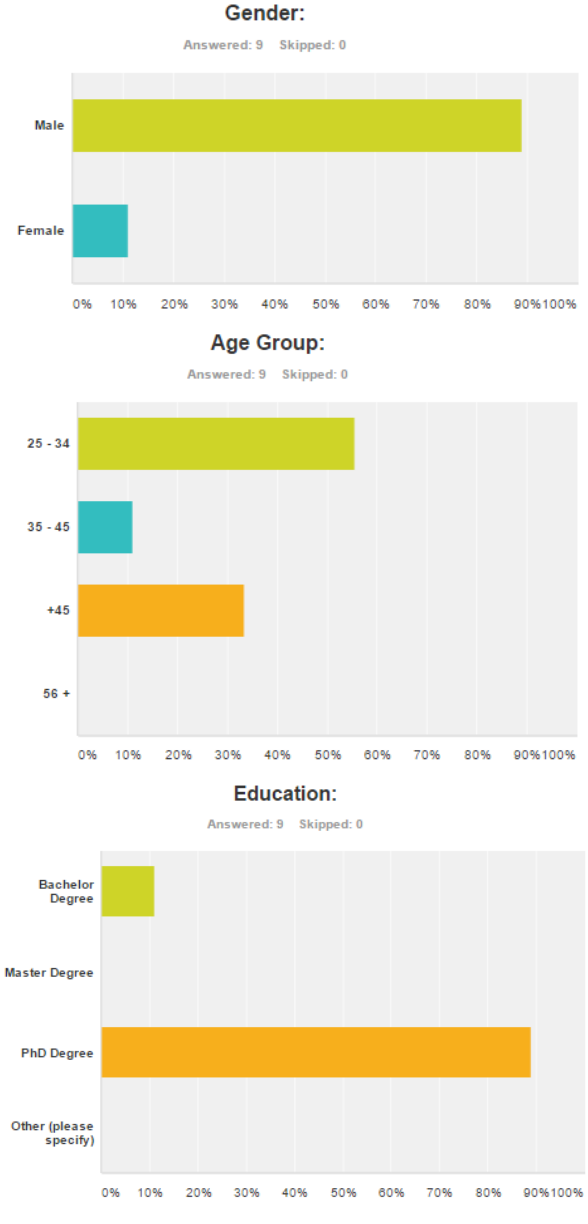


Figure 3 Expert users' background (continued) (see online version for colours)

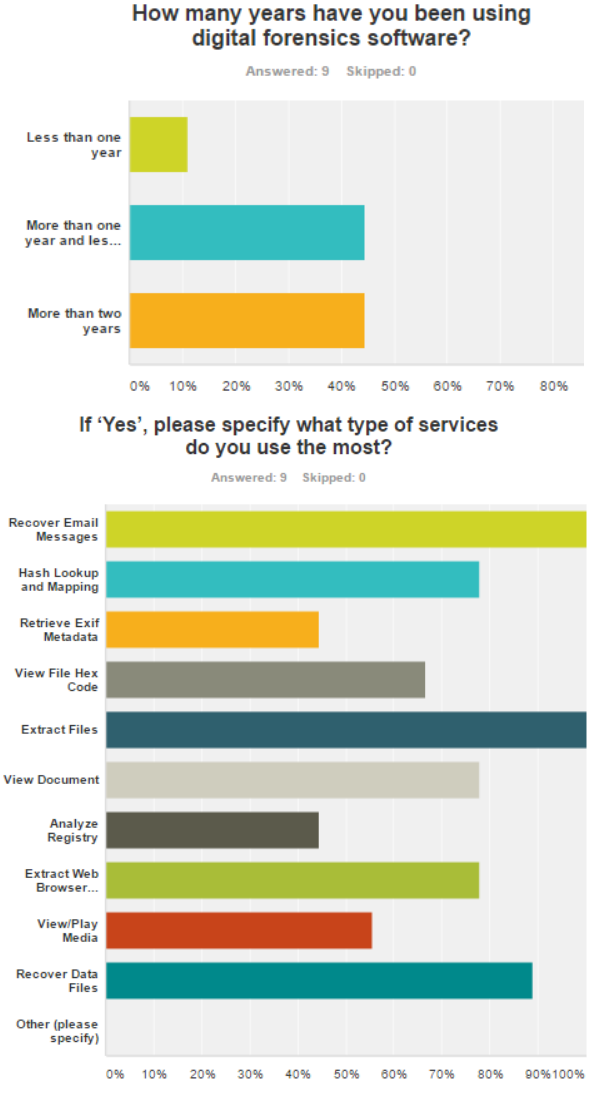


Table 19 Expert users' results – part 1

	Very easy	*Easy*	*Neutral*	*Difficult*	*Very difficult*	*Total*	*Weighted average*
Learning autopsy was	33.33% 3	44.44% 4	22.22% 2	0.00% 0	0.00% 0	9	1.89
Learning DFF was	0.00% 0	55.56% 5	44.44% 4	0.00% 0	0.00% 0	9	2.44
Learning DART was	11.11% 1	11.11% 1	33.33% 3	44.44% 4	0.00% 0	9	3.11
Performing a task on Autopsy was	22.22% 2	55.56% 5	22.22% 2	0.00% 0	0.00% 0	9	2.00
Performing a task on DFF was	0.00% 0	22.22% 2	22.22% 2	44.44% 4	11.11% 1	9	3.44
Performing a task on DART was	11.11% 1	11.11% 1	33.33% 3	22.22% 2	22.22% 2	9	3.33
Finding a feature on Autopsy was	22.22%	44.44%	33.33%	0.00%	0.00%	9	2.11
Finding a feature on DFF was	0.00%	22.22%	22.22%	55.56%	0.00%	9	3.33
Finding a feature on DART was	11.11%	11.11%	22.22%	11.11%	44.44%	9	3.67
Understanding Autopsy navigation was	22.22%	44.44%	33.33%	0.00%	0.00%	9	2.11
Understanding DFF navigation was	0.00% 0	22.22% 2	44.44% 4	33.33% 3	0.00% 0	9	3.11
Understanding DART navigation was	11.11% 1	22.22% 2	33.33% 3	33.33% 3	0.00% 0	9	2.89
Recovering from error for Autopsy was	22.22% 2	44.44% 4	33.33% 3	0.00% 0	0.00% 0	9	2.11
Recovering from error for DFF was	0.00% 0	22.22% 2	33.33% 3	11.11% 1	33.33% 3	9	3.56
Recovering from error for DART was	11.11% 1	11.11% 1	33.33% 3	0.00% 0	44.44% 4	9	3.56
Managing an account in Autopsy was	22.22% 2	55.56% 5	22.22% 2	0.00% 0	0.00% 0	9	2.00
Managing an account in DFF was	0.00% 0	44.44% 4	55.56% 5	0.00% 0	0.00% 0	9	2.56
Managing an account in DART was	11.11% 1	11.11% 1	33.33% 3	44.44% 4	0.00% 0	9	3.11

Table 20 Expert users' results – part 2

	Strongly agree	*Agree*	*Neutral*	*Disagree*	*Strongly disagree*	*Total*
I would prefer to use autopsy to perform my tasks	55.56%	33.33%	11.11%	0.00%	0.00%	9
	5	3	1	0	0	
I would prefer to use DFF to perform my tasks	0.00%	22.22%	77.78%	0.00%	0.00%	9
	0	2	7	0	0	
I would prefer to use DART to perform my tasks	11.11%	11.11%	33.33%	44.44%	0.00%	9
	1	1	3	4	0	
Using autopsy save time	44.44%	33.33%	11.11%	11.11%	0.00%	9
	4	3	1	1	0	
Using DF save time	0.00%	66.67%	22.22%	11.11%	0.00%	9
	0	6	2	1	0	
Using DART save time	11.11%	11.11%	44.44%	33.33%	0.00%	9
	1	1		3	0	

8 Conclusions and future work

In this paper, a comprehensive set of existing quality models and their application to OSS has been introduced. We applied an existing quality-in-use model (Alnanih, 2015), inspired by ISO/IEC 25010 (2011), to assess the user experience of three open source digital forensics tools. Briefly, we can clearly conclude from our results that autopsy, DFF, and DART perform similarly in terms of efficiency and productivity. However, DDF outperforms the other two slightly when it comes to effectiveness. Autopsy outperforms in terms of error safety, and DART outperforms in terms of cognitive load. The goal of this paper is to initiate a dialogue regarding the quality requirements of open source digital forensics tools within its community, with the aim t of improving them.

The evaluation of the quality-in-use of the three tools does not reflect the quality of all open source digital forensics tools. The idea for this research paper is to start a dialog that can ultimately lead to clear and more simply applied set quality requirements for the aforementioned tools. So, the future will involve conducting a more comprehensive study that will encompass more aspects of software quality, and not only the user experience of three main products. In addition, a comparative study of the open and closed source digital forensics programs using standardised software quality requirements will be considered. This will enhance testing efforts and increase the quality of this type of software.

References

Akbari, M. and Peikar, S. (2014) 'Evaluation of free/open source software using OSMM model case study: WebGIS and spatial database', *Advances in Computer Science*, *An International Journal*, Vol. 3, No. 5, pp34–43.

Alnanih, R. (2015) *CON-INFO: A Context-based Methodology for Designing and Assessing the Quality of Adaptable MUIs in Healthcare Applications*, PhD thesis.

Altheide, C. and Carvey, H. (2011) *Digital Forensics with Open Source Tools*, 1st ed., Elsevier, Syngress Publishing Rockland, MA, USA, ISBN-13: 978-1597495868.

Autopsy (2016) [online] http://www.sleuthkit.org/autopsy/ (accessed 21 April 2016).

Aversano, L. and Tortorella, M. (2011) *Applying EFFORT for Evaluating CRM Open Source Systems*, Department of Engineering, University of Sannio, Via Traiano, 82100 Benevento, Italy, DOI: 10.1007/978-3-642-21843-9_17.

Aversano, L. and Tortorella, M. (2010) 'Evaluating the quality of free/open source systems: a case study', July, Vol. 55, No. 7, DOI: 10.1007/978-3-642-19802-1.

Bennett, D.J. and Stephens, P. (2008) 'A usability analysis of the autopsy forensic browser', *Proceedings of the Second International Symposium on Human Aspects of Information Security & Assurance (HAISA 2008)*.

Bodker, M., Nielsen, L. and Orngreen, R.N. (2007) 'Enabling user-centered design processes in open source communities, usability and internationalization', HCI and Culture, in *Proceedings of the 2nd International Conference on Usability and Internationalization held as Part of the HCI International Conference Part I*, pp.10–18.

Carrier, B. (2002a) 'Defining digital forensic examination and analysis tools', in *Digital Forensic Research Workshop 2002*.

Carrier, B. (2002b) 'Open source digital forensics tools: the legal argument', *Stake.*

Das, A. and Wasserman, A.I. (2011) 'Using flossmole data in determining business readiness ratings." Workshop on public data about software development', *The 3rd International Conference on Open Source Systems*, IFIP, Vol. 2.

DEFT [online] http://www.deftlinux.net/ (accessed 21 April 2016).

DFF (2016) [online] http://www.digital-forensic.org/ (accessed 21 April 2016).

Digital Forensics Pentest Linux Distributions (DEFT) (2016) *Digital Forensics Toolkit: DEFT*, CyberPunk, [online] https://n0where.net/digital-forensics-toolkit-deft (accessed 1 December 2016).

Duijnhouwer, F. and Widdows, C. (2003) *Open Source Maturity Model, Capgemini Expert Letter*,. EU QualOSS project, Grant number: 033547, IST-2005-2.5.5.

Flandrin, F., Buchanan, W.J., Macfarlane, R., Ramsay, B. and Smales, A. (2014) 'Evaluating digital forensic tools (DFTs)', in *7th International Conference: Cybercrime Forensics Education & Training*, September, pp.1–16.

Groot, A., Kügler, S., Adams, P.J. and Gousios, G. (2006) 'Call for quality: open source software quality observation', *Open Source Systems*, *IFIP International Federation for Information Processing 2006*, DOI: 10.1007/0-387-34226-5_6.

Groven, A., Haaland, K., Glott, R. and Tannenberg, A. (2010) 'Security measurements within the framework of quality assessment models for free/libre open source software', *Proceedings of the Fourth European Conference on Software Architecture*, New York, NY, USA.

Hibshi, H., Vidas, T. and Cranor, L. (2011) 'Usability of forensics tools: a user study', published in *Proceedings IMF'11*, *Proceedings of the 2011 Sixth International Conference on IT Security Incident Management and IT Forensics*, pp.81–91, IEEE Computer Society Washington, DC, USA ©2011 ISBN: 978-0-7695-4403-8, DOI: 10.1109/IMF.2011.19.

Houaich, Y.A. and Belaissaoui, M. (2015) 'Measuring the maturity of open source software', *6th International Conference on Information Systems and Economic Intelligence (SIIE),* Hammamet, DOI: 10.1109/ISEI.2015.7358735.

International Organization for Standardization (ISO) (2007) *ISO/IEC 25030 – Software Engineering – Software Product Quality Requirements and Evaluation (SQuaRE) – Quality Requirements*, June, ISO/IEC.

International Organization for Standardization (ISO) (2011) *ISO/IEC 25040 – Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – Evaluation Process*, ISO/IEC, February.

ISO (1998) *ISO 9241-11: Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs): Part 11: Guidance on Usability*.

ISO (2001) *ISO/IEC 9126-1:2001*, *Software Engineering: Product Quality*.

ISO (2011) *ISO/IEC 25010:2011 Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – System and Software Quality Models*.

ISO/IEC 27037:2012 (2012) *Guidelines for Identification, Collection, Acquisition and Preservation of Digital Evidence*; *The International Organization for Standardization (ISO)*; *The International Electrotechnical Commission (IEC)*, ISO/IEC, Geneva, Switzerland.

ISO/IEC 27041:2015 (2015a) *Guidance on Assuring Suitability and Adequacy of Incident Investigative Method*; *The International Organization for Standardization (ISO)*; *The International Electrotechnical Commission (IEC)*, ISO/IEC, Geneva, Switzerland.

ISO/IEC 27042:2015 (2015b) *Guidelines for the Analysis and Interpretation of Digital Evidence*; *The International Organization for Standardization (ISO)*; *The International Electrotechnical Commission (IEC)*; ISO/IEC, Geneva, Switzerland.

Kaur, M., Kaur, N. and Khurana, S. (2016) 'A literature review on cyber forensic and its analysis tools', *IJARCCE*, Vol. 5, No. 1, pp.23–28 [online] http://www.ijarcce.com/upload/2016/january-16/IJARCCE%206.pdf (accessed December).

Lempereur, B., Merabti, M., and Shi, Q. (2006) 'Pypette: a framework for the automated evaluation of live digital forensic techniques', in *11th Annual PostGraduate Symposium on The Convergence of Telecommunications Networking and Broadcasting*.

Lyle, J.R. (2010) 'If the error rate is such a simple concept, why don't I have one for my forensic tool yet?', *Journal of Digital Investigation*, Vol. 7, No. 8, pp.135–139.

Manson, D., Carlin, A., Ramos, S., Gyger, A., Kaufman, M. and Treichelt, J. (2007) 'Is the open way a better way? Digital forensics using open source tools', *Proceedings of the 40th Hawaii International Conference on System Sciences*.

NIST (2017a) *Computer Forensics Tool Testing Program: Project Overview* [online] http://www.cftt.nist.gov/ (accessed 25 December 2015).

NIST (2017b) *Digital Data Acquisition Tool Test Assertions and Test Plan*, National Institute of Standards and Technology, Draft 1 [online] http://www.cftt.nist.gov/DA-ATP-pc-01.pdf (accessed 13 October 2018).

NIST (2001) *General Test Methodology for Computer Forensic Tools*, National Institute of Standards and Technology, Tech. Rep. Version 1.9 [online] http://www.cftt.nist.gov/Test%20Methodology%207.doc (accessed 25 December 2015).

Nurse, J.R.C. et al. (2011) 'Guidelines for usable cybersecurity: past and present', *Third International Workshop on Cyberspace Safety and Security (CSS)*, IEEE.

Pan, L. and Batten, L.M. (2009) 'Robust performance testing for digital forensic tools', *Journal of Digital Investigation*, Vol. 6, Nos. 1–2, pp.71–81.

Raza, A., Capretz, L.F. and Ahmed, F. (2011) 'An empirical study of open source software usability: the industrial perspective', *International Journal of Open Source Software and Processes (IJOSSP)*, Vol. 3, No. 1, p.16, DOI: 10.4018/jossp.2011010101.

Samoladas, I., Gousios, G. and Spinellis, D. (2008) 'The SQO-OSS quality model: measurement-based open source software evaluation', *Open Source Development, Communities and Quality, IFIP – The International Federation for Information Processing*, pp.237–248, DOI: 10.1007/978-0-387-09684-1_19.

Soto, M. and Ciolkowski, M. (2009) 'The QualOSS open source assessment model measuring the performance of open source communities', *3rd International Symposium on ESEM 2009*, Lake Buena Vista, DOI: 10.1109/ESEM.2009.5314237.

Soto, M. and Ciolkowski, M. (2009) 'The QualOSS process evaluation: initial experiences with assessing open source processes', *Software Process Improvement: 16th European Conference*, *EuroSPI*, Alcala, Madrid, Spain, DOI: 10.1007/978-3-642-04133-4_9, Source Forge [online] https://sourceforge.net/projects/cyclos/?source=directory (accessed 21 April 2016).

Wasserman, A.I., Pal, M. and Chan, C. (2006) 'Business readiness rating for open source', *Proceedings of the EFOSS Workshop*, Como, Italy.

Zaharias, P. and Poylymenakou, A. (2009) 'Developing a usability evaluation method for e-learning applications: beyond functional usability', *International Journal of Human-Computer Interaction*, Vol. 25, No. 1, pp.75–98, DOI: 10.1080/10447310802546716.

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*, Vol. 34, 10.1177/001872089203400407.

Abbreviations

ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
PDAs	Personal digital assistants
PEDs	Personal electronic devices
IS	Information security
DFF	Digital forensics framework
DART	Digital advanced response toolkit
NIST	National Institute of Standards and Technology
CFTT	Computer forensics tool testing
OSS	Open source software
OSMM	Open source maturity model
FLOSS	Free/open source
BRR	Open business readiness rating open
UI	User interface
SQO-OSS	Software quality observatory for OSS
QualOSS	Quality of open source software
EFFORT	Evaluation framework for free/open source projects
ERP	Enterprise resource planning.