International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7630
***Article Received:** 01 June 2023 **Revised:** 28 July 2023 **Accepted:** 10 August 2023*

A Review on Software Quality Forensics: Techniques, Challenges, and Limitations

Sandeep Kumar Sharma[1], Dr. Mazhar Khaliq[2]
[1]Research Scholar ,Department of Computer Science & IT
Khwaja Moinuddin Chishti Language University,
Lucknow, Uttar Pradesh. India,
e-mail: sandeep24af@gmail.com
[2]Assistant Professor,Department of Computer Science & IT
Khwaja Moinuddin Chishti Language University,
Lucknow, Uttar Pradesh. India,
e-mail: dr.mazharkhaliq@gmail.com

Abstract— Software quality forensics plays a vibrant role related to software quality, security, and integrity. The paper aims to derive a software quality forensics model through existing software quality models and their factors. The papers explore quality models, factors, approaches, tools, techniques, and standards regarding software quality investigation and confine the research area for software quality integrity breach forensics. The explore the deviations of quality attributes, standards, factors, and artifacts, it leads to further investigation of root-cause followed by digital evidence procedure for alleged software quality issues. Therefore, there is a need for a software quality forensics model and dedicated standards to fulfill the digital evidence procedure validation, satisfiable, and prosecution in the court of law in the context of alleged or illegal activity investigation quality of software. The paper has derived the techniques, challenges, and limitations of software quality forensics based on the review of research questions.

Keywords- Software quality model, Software quality forensic, Integrity breach, software forensic.

I. INTRODUCTION

Software quality forensics is a novel approach to enhance and the advent of quality forensics of software with the scope of standardization aspect with emerging technologies. If the application changes its behavior from normal to malicious. It includes software testing and artifacts as part of software quality forensics to test alleged software applications and mapping quality factors, standards, and models with forensics concerns. Software quality forensics illustrates the forensics of software, quality of evidence, validation and verification of forensics tools, and alleged software. As we know that forensic process starts after the events or crimes have occurred. Now we can clarify the need for Software quality forensic reference with quality-related court cases financial or property damage, physical harm, downtime, data loss, or data integrity breach. If a client loses money as a result of a software flaw, the software company may have to pay a steep legal bill. According to software liability law, anyone can sue a software developer for negligence or if the application crashes and causes financial harm. We are exploring some cases that happened regarding software quality-related issues that lead to loss of financial, and physical injuries due to software defects. In the 1980s “Therac 25”, a radiation therapy machine. In that case, software engineering mistakes led to radiation overdose deaths. After discovering a software flaw that might lead to the Prius cars stalling at high speeds in the latter part of 2018, Toyota recalled approximately 2.5 million of the hybrid automobiles. A similar flaw caused the business to recall 1 million vehicles earlier. Numerous lawsuits were brought about by Toyota's alleged failure to fix the software correctly. The Federal Aviation Administration attributes two deadly Boeing 737 MAX aircraft crashes caused by faulty automated control systems to software flaws. the March 2019 Ethiopian Air accident and the Indonesian Lion Air crash. [1,2]. “Model S and Model X from Tesla”, A class action lawsuit was brought against Tesla in May 2020 due to a broken touch screen. [1]. Bhattathiripad in Appendix-1.[3] derived software quality forensic challenges in alleged software, “Such an instance came out in the form of a suit in a court in the southern state of Kerala in India. The owner of a banking business firm filed a civil suit against his software developer complaining against the development and delivery of low-quality bank management software.” This case study has motivated us to point out the need for software-quality forensic and law enforcement directions to handle such types of futuristic cases. To explore the directions. There is a need to design and develop of software quality model for investigation and prosecution allied with quality factors and standards mapped with past and current scenarios.

So, considering this aspect Gillies.[4] explore the quality management systems that draw attention towards the system must be supported through high-quality software that should

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7630
***Article Received:** 01 June 2023 **Revised:** 28 July 2023 **Accepted:** 10 August 2023*

operate "correctly and dependably, be scalable, and meet the needs of all users for Information, knowledge, and wisdom storage, retrieval, and processing". Easy, safe, and suitable for testing, reusing, security, and maintaining, as well as conforming to the demands of stakeholders. [5-8] describe "For qualitative investigation, code smells, refactoring design patterns, quality models, human factors, testing, and quality prediction are used.". To find out pre-identification for investigation purposes. Hynnine, Kasurinen, & Taipale. [9] Describe the run-time framework for assessing software quality characteristics and software quality in use models according to "ISO/IEC 25000". During the software development process, measurement probes are connected to the program and used to gather quality data while it is running. Alim & Purwanti. [10] explore the impact of auditing, to trace the elements that affect the quality. The first is the auditor's competency and the second is the digital forensic support (DFS). Pasquale, Alrajeh, Peersman, Tun, Nuseibeh, & Rashid [11] According to the audit literature, auditing competency is necessary to maintain audit quality. "An independent, manual, and automated application-level review should be performed on every software component used in the application." the examination of patterns that compromise the application's confidentiality, availability, and integrity. According to Eloff and Bella [12] The investigation of software failures makes use of a near-miss management system for the digital forensic process model. In another paper Bella & Eloff. [13] " suggests near-miss analysis can enhance the gathering of valid proof regarding software flaws. According to Ekanem & Meye. [14] the automation of reverse engineering in software forensics could reveal alterations that were made by the alleged infringer during the development of the alleged counterfeit software, empowering the prosecuting team to prove the case beyond reasonable suspicion in court. Software testing is critical, particularly to evaluate the quality. Juniawan.[15] According to the "McCall approach framework", which focuses on software product operation aspects, the five metrics that are examined are "Correctness, reliability, efficiency, integrity, and usability.

To explore software quality, The research has very vast so according to the forensic point of view we confine the area related to quality issues research. Through following the keyword Software quality forensics very less research has existed, now we can relate the term with forensics investigation towards quality issues. So, in addition to being one of the most important, software quality is also a complex feature of computer software. The paper is followed by four sections, the first is the introduction and motivation for the review of software quality forensics second discusses methodology & literature review based on research questions of software quality forensics development and issues. This section has research questions and related work described to answer each question. The third section is Results and Discussion derived the challenges and limitations from the section second. In section four, we have finalized the conclusion and possible future directions.

II. METHODOLOGY

The paper is based on the following research questions.

RQ1 What contributions have been made regarding the Software quality forensic?

RQ2 What are the categories of Software quality forensic?

RQ3 How can map software quality models with quality forensic aspects?

RQ4 How to describe data integrity and software integrity breaches regarding software quality issues?

RQ5 What are tools and standards that can facilitate software quality investigations?

The RQ1 and RQ2 were used for describing the software quality forensic and categories areas, after that RQ3 and RQ4 were derived from RQ1 and RQ2 and finally, to answer the RQ5, identify the tools and standards related to software quality forensics.

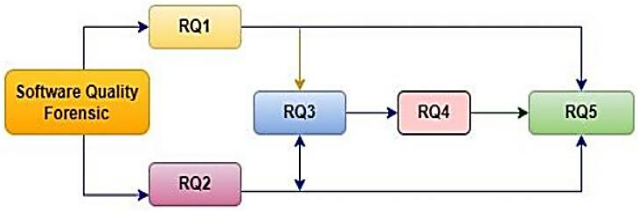

Figure 1. Research questions mapping

III. LITRATURE REVIEW

This section elaborates on existing kinds of literature based on research questions mentioned as RQ1 to RQ5.

RQ1 What contributions have been made regarding the Software quality forensic?

Significant research on software quality already exists, but there are specific requirements of software quality sub-areas, domains based on new trends, and best practices in inclusive software, it would need to be modified and expanded [15-18]. Oveisi, Farsi, Nadjafi, & Moeini, [19] Introduces a method for gathering engineering evidence that analyses software throughout its lifecycle following system safety engineering and software safety engineering concepts. This strategy makes sure that software risks are recognized and recorded throughout the program lifecycle, after which they are diminished to a level that is acceptable in terms of safety using the suggested techniques. Software development must take software safety and defect detection into account. Johnson & Hatton. [20,21] The task of forensic software engineering entails planning, "executing search operations, collecting, and preserving evidence of software failures and faults for proximate analysis". Rowley & Ramakrishnan [22], "The examination of software-induced accidents and software-facilitated crimes will be included in this

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7630
***Article Received:** 01 June 2023 **Revised:** 28 July 2023 **Accepted:** 10 August 2023*

paper's review of conventional crime and accident scene reaction, evidence gathering, and preservation procedures."

The following table summarizes the related research with specific contributions toward software quality investigation:

TABLE I. DERIVATION OF SOFTWARE QUALITY FORENSIC

Title of Research Paper	The outcome of the paper	Source
"Appendix 1: Challenges in Software Quality Forensics and Litigation-A Case Study."	Through this case study, we have extracted the idea for software quality forensics.	[3]
"Forensic Software Engineering: are Software Failures Symptomatic of Systemic Problems?"	The creation of safety-critical applications, systemic analysis of failures, and accident investigation were the main topics of this article.	[22]
"Towards early software reliability prediction for computer forensic tools (case study)"	Markov chains-based and architecture-based software dependability prediction model for computer forensic tools.	[23]
"Computer Source Code: A Source of the Growing Controversy Over the Reliability of Automated Forensic Techniques"	This paper explains the utility of validation studies for evidentiary privilege in trade secrets.	[24]
"A model of qualitative factors in forensic-ready software systems. In: Lecture Notes in Business Information Processing."	The reference model is used in the study to outline the prerequisites for forensic readiness.	[25]
"Application of Quality in Use Model to Assess the User Experience of Open-Source Digital Forensics Tools"	Outlines software quality methodology and digital forensics tools testing efforts.	[26]
"Quality standards for digital forensics: Learning from experience in England & Wales"	Addresses the challenges of implementing the standards, "ISO/IEC 17025, quality issues in digital forensics".	[27]
"Accrediting digital forensics: what are the choices?"	This paper focuses on Quality failures in accredited organizations' evidence quality standards.	[28]

RQ2 What are the categories of Software quality forensic?

Through the literature survey software quality forensics is categorized into seven wings to fulfill the development of the ontological framework for quality forensics. After reporting quality issues or alleged software the software quality forensics approach as mentioned; Incident Investigation, Root Cause Analysis, Evidence Collection, Analysis Tools & Techniques Software quality factors and metrics, Software Quality Standards, Reporting, and Presentation as digital evidence to the court of law.

The following fig. explores each approach:

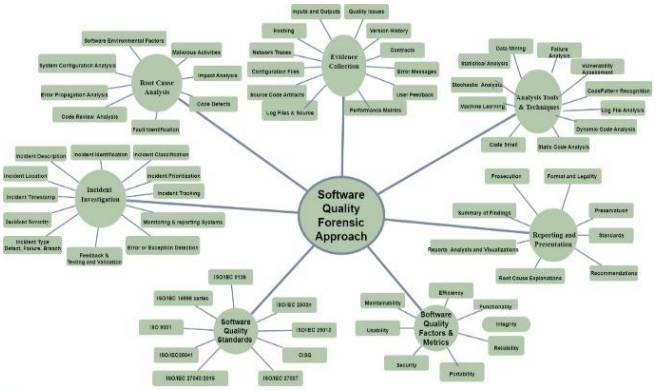

Figure 2. Software Quality Forensics Approach

RQ3 How can we map software quality models with Software quality forensic aspects?

The measuring quality forensics is not only dedicated to software but also performing the verified forensics laboratory's software. Mapping software quality models with quality forensic aspects involves identifying the relevant factors and dimensions from software quality models and aligning them with the specific requirements and considerations of quality forensics.

The following approach to mapping these aspects:

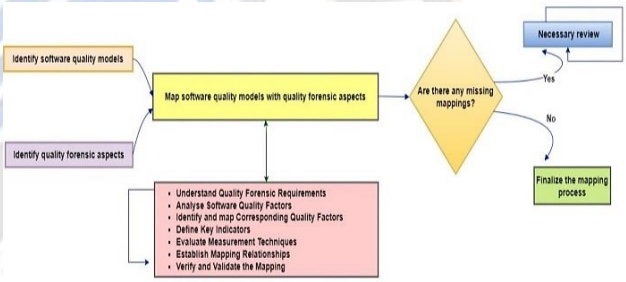

Figure 3. Conceptual framework of software quality forensics

A. Identify Software Quality Models

We start by finding the models for software quality that are widely accepted or used in the sector. comparable to the "ISO/IEC 25000 series".[29], "ISO/IEC 25010" [30] The "SQuaRE" (Software Product Quality Requirements and Evaluation model) series. "Boehm's Quality Model" [31] "McCall's Quality Model" [32] forensic best practices in focus. According to Miguel & et al. [33] a quality model is defined as "the set of characteristics, and the relationships between them, that provides the basis for specifying quality requirements and evaluation" by "ISO/IEC IS 9126-1". The models for evaluating software quality have been developed, and each model includes the basic characteristics as well as the sub-factors or sub-characteristics that fall under each of them. Each sub-factor is given a metric for the actual evaluation. As shown in the comparison table, tailored quality models have essentially identical features to entry-level ones. It should be distinguished, nevertheless, that A model is not unsuccessful because it lacks a certain feature. The sub-features of fundamental software

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7630
***Article Received:** 01 June 2023 **Revised:** 28 July 2023 **Accepted:** 10 August 2023*

quality models and customized-oriented models are investigated to assess the existing position regarding the identification of software quality forensic model identification and ontological development of quality forensic investigation.

According to Glott [34] Models used to assess the quality of products created with free software are modified by models. like "ISO-9126" and include some context-specific information. It is interesting that despite the distinction between models of the first and second generations there is yet no perfect model that accurately depicts every facet of quality in a piece of free software. There are various models for assessing the quality of open-source software, according to Madaehoh and Senivongse [35]. A good example of one of these is the "Cap Gemini Open-Source Maturity Model," which evaluates the product's maturity using maturity indicators. The "Open BRR Model, or Business Readiness Rating framework," was inspired by "the Cap Gemini and ISO 9126 Models" and identifies seven characteristics that are essential for judging open-source software. These categories can be further broken down for greater clarity and to include subjects that have not yet been thoroughly explored. Miguel & et al. [33] said that another model is the "SQO-OSS Model", which evaluates the source code and community process to allow for the automatic calculation of metrics. Madaehoh & Senivongse.[35] Finally, the "Qual-OSS Model "asserts that the environment in which it is used and the goals that an organization or individual pursues with it have a significant impact on quality. A full collection of quality metrics is available from "OSS-AQM," along with an automation tool that gathers data from "SonarQube, Stack Exchange, GitHub, and other sources. By utilizing this information, it quantitatively evaluates the general level of "open-source software quality" and offers insightful information on its functionality and dependability.

Miguel, Mauricio & Rodriguez [33] Basic Models, which choose the attributes and sub-attributes to evaluate based on a certain domain, post Tailored Quality Models. This type of model is adapted for a certain product or viewed from the viewpoint of a user realm. hence, imposed limitations. The three subcategories of the software quality model are "Basic quality model," "Tailored Quality Models," and "Open-Source Quality Models." As mentioned in Table 2. and Table 3. below:

TABLE II. BASIC QUALITY MODEL, [33]

Characteristic	**"A"**	"B"	**"C"**	**"D"**	**"E"**	**"F"**
Accuracy					X	X
Adaptability			X			X
Analysability					X	X
Attractiveness					X	X
Changeability					X	X
Correctness	X					X
Efficiency	X	X		X	X	X
Flexibility	X					
Functionality			X	X	X	X
Human Engineering		X				
Install ability					X	X
Integrity	X					X
Interoperability	X					X
Maintainability	X			X	X	X
Maturity					X	X
Modifiability						X
Operability					X	X
Performance			X		X	X
Portability	X	X		X	X	X
Reliability	X	X	X	X	X	X
Resource utilization					X	X
Reusability	X			X		X
Stability					X	X
Suitability					X	X
Supportability			X		X	X
Testability	X	X			X	X
Transferability						X
Understandability		X			X	X
Usability	X		X	X	X	X

A. "McCall", B. "Boehm", C. "FURPS", D. "Dromey", E. "ISO-9126", F. "ISO-25010".

TABLE III. TAILORED MODELS [33]

Characteristic	**"Bertoa"**	**"Gecuamo"**	**"Alvaro"**	**"Rawashdeh"**
Accuracy	X		X	X
Adaptability		X	X	
Analyzability				
Attractiveness				
Changeability	X		X	X
Compliance	X	X	X	X
Configurability			X	
Compatibility				X
Correctness		X		
Efficiency			X	X
Fault Tolerance			X	
Flexibility				
Functionality	X	X	X	X
Human Engineering				
Install ability				
Integrity				
Interoperability	X		X	X
Learnability	X	X	X	X
Maintainability	X		X	X
Manageability				X
Maturity	X	X	X	X
Modifiability				
Operability	X			X
Performance				
Portability	X		X	

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7630
***Article Received:** 01 June 2023 **Revised:** 28 July 2023 **Accepted:** 10 August 2023*

Recoverability	X			X
Reliability	X		X	X
Replaceability	X		X	
Resource utilization	X	X	X	X
Reusability	X		X	
Scalability			X	
Stability			X	
Security	X		X	X
Self-Contained			X	
Suitability	X		X	X
Supportability				
Testability	X	X	X	X
Time Behavior	X		X	X
Understandability	X	X	X	X
Usability	X	X	X	X

B. Identify Quality Forensic Requirements

Milošević et-al. [36] Gain a clear understanding of the specific requirements and considerations of quality forensics.[37], Doyle.[38] explained that the system includes factors such as objectivity, evidence handling, legal and ethical compliance, confidentiality, professionalism, admissibility in legal proceedings, and documentation/reporting practices. According to Daubner & et-al. [25] The report recommends using qualitative factor reference model preparedness of model as a starting point for formulating standards that can be confirmed. The model improves the current "forensic-ready risk management method", by defining the Forensic readiness requirements, and non-Disputability in specific terms. "FR-ISSRM" The two strategies have a similar goal that can be achieved by combining them to form sub-factors. Applying measurements and enabling alternative implementations both benefit from such hierarchical organization. Based on the outcomes of “forensic-ready risk management”, the reference model is utilized to describe the requirements for forensic readiness.

C. Analyze Software Quality Factors

Review the different quality factors outlined in the selected software quality models. According to Behnamghader & et-al. [39] “Reliability, maintainability, usefulness, efficiency, portability, and security are often included in these characteristics.” Evaluate each factor to identify its relevance and alignment with quality forensic requirements. Analysis of software quality improvements over releases ignores the minute modifications that each commit makes. Commit-impact analysis, which compares software quality before and after each commit, can, in our opinion, disclose a plethora of information about how the software develops and how each change affects its quality. The impact of each commit on the source code is investigated in this research, along with the compilability of each significant commit and how source code modifications affect software quality metrics.

D. Identify Corresponding Quality Factors

Identify the software quality factors that directly correspond or overlap with the requirements and considerations of quality forensics. For example, the integrity factor in software quality models may align with the need for objectivity and evidence handling in quality forensics. As focused with integrity quality factor derived the subquality derivation, The three parameters that make up the integrity factor are security, instrumentation, and audibility.

Therefore, the equation is used to calculate the value of integrity:

$$\text{Integrity} = \frac{Auditability + instrumentation + security}{3}$$

Cavano & McCall [32] The term "audibility" refers to how easily software and data can be examined for standard compliance. The metric represents “instrumentation” aspects of the devices that are present in the system and can detect errors. To maintain data security in the system, “security investigate metrics” looks at the security that is already in place.

E. Define Key Indicators

Define specific indicators or metrics within the selected software quality factors that are relevant to quality forensics. For instance, within the integrity factor, indicators may include adherence to legal and ethical standards, evidence-handling protocols, or compliance with rules of evidence. [30,32] The aspect of communication quality factors has not been incorporated in all the models. Integrity factor unified in only “McCall and ISO-25010 quality model”. So, the chain of custody evidence flow communication is an important factor to deal with forensics at all levels of investigation.

F. Evaluate Measurement Techniques

Assess the measurement techniques or methods used in software quality models to evaluate the identified quality factors. Consider how these techniques can be adapted or extended to measure the corresponding factors concerning higher-quality forensics. Alvaro’s software quality model proposes a framework for evaluating and certifying software components to establish their quality. [30] Based on the “ISO 9126 standard”, the suggested model gives a collection of quality characteristics and sub-characteristics for software components. Alvaro, Almeida, Vasconcelos, & Meira [40] The model build on tests that developers provide in a portable standard format. The methodology is used to effectively assess software components' quality. To assure quality, this study offers a “Software Component Quality Framework” based on clearly defined modules that work in tandem.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7630
***Article Received:** 01 June 2023 **Revised:** 28 July 2023 **Accepted:** 10 August 2023*

G. Establish Mapping Relationship

Establish the linkages between the software quality criteria and the quality forensic aspects by creating a mapping document or matrix. The alignment between each quality element and the requirements of quality forensics should be highlighted by this mapping. The legal system presents challenges for the "Digital Forensics and Incident Response (DFIR)" area, which relies on software tools to capture and analyze digital evidence. The evidence must be complete, accurate, reliable, and accessible through repeatable procedures to be admitted in court. The absence of acceptable software quality, on the other hand, is a renowned problem in this situation. Madaehoh & Senivongse.[35] The availability of tools across channels has expanded due to the popularity of OSS-based development, but their inconsistent quality is a cause for concern. More lengthy software supply chains have added new variables that affect tool quality and give users control over using specific software versions. Prior research on the tool's quality level tended to concentrate more on the tool's core codebase than on its underlying dependencies. According to Saari. [41]. There has not been any research on how the software supply chain affects quality factors in the realm of "DFIR".

H. Verify and Validate the Mapping:

Validate the mapping by consulting with experts in both software quality and quality forensics. Gather feedback and insights to ensure that the mapping accurately represents the relationship between software quality models and quality forensic aspects. According to Daubner, Matulevičius, & Buhnova [25] However, the legal situation makes it difficult to confirm if the software is “forensic.” To be utilized in court, digital evidence "must be comprehensive, accurate, reliable, and acquirable in reproducible methods." The absence of acceptable software quality and the proposed "Model of Qualitative Factors in Forensic-Ready Software Systems" are recognized problems in this regard, according to Daubner, Matulevičius, & Buhnova. [25]. The following are some resources that discuss the challenges of developing and validating forensic software. Brecht. [42] This paper "suggests a paradigm of forensic-ready software system quality criteria that can be utilized to create verifiable requirements. Brunty.[43] This resource discusses the importance of the three as of computer forensics: Acquire, Authenticate, and Analyze. According to Guo, Slay, Beckett, & Watson. [44] It also highlights the need for forensic analysts to examine digital evidence and provide key discoveries to identify those responsible for a crime. Brunty.[43] and Guo, Slay, Beckett, & Watson. [44] focused on forensic tools and procedures validation.[45] It would be a resource for digital forensics investigators. The Guo, Slay, Beckett, & Watson.[44] To verify that forensic procedures and instruments work properly and as intended, the paper highlights the significance of thorough testing as well as systematic validation and verification. software tools for computer forensic validation and verification. Brunty, [43] as well as Guo, Slay, Beckett, & Watson.[44] The significance of validating and confirming computer forensic software tools is discussed in this article to make sure they work properly and as intended. [45] New Methods for gathering and examining digital Evidence. This article examines the difficulties of gathering digital proof and the need of creating fresh strategies for obtaining and analyzing digital media. Rowley & Ramakrishnan. [46] This paper demonstrates that a top-down approach to quality analysis, more specifically, verification and validation, facilitates forensic software profiling. Organizations can relate the software development process to the unique demands of quality forensic investigations using the proposed mapping of software quality models with quality forensic aspects. This mapping helps ensure that software systems are developed and evaluated with considerations for maintaining integrity, objectivity, adherence to legal standards, and the ability to withstand forensic scrutiny necessary review assurance.

RQ4 How to describe data integrity and software integrity breaches regarding software quality issues?

Applications are prone to malicious infection and act as hostile malicious code that redirects personal data to the dedicated server or hijacking apps depriving software quality. According to the problem definition data integrity breaches over incompetence to track the quality behavior of applications, what gets stolen, how it happens and where is the destination of data. It is a very complex task to discriminate app behavior for bypassing breached integrity data. Kreitzberg & et al. [47] has defined that a program's integrity is determined by its capacity to respond correctly to various sets of input. Data integrity breach deficient quality system, data integrity is not only intentional but includes application fault also. The paper focuses on a Software quality forensics approach to identify data integrity breaches. “Data integrity is defined as the consistency, accuracy, completeness, and dependability of data throughout its life. Data integrity violations, poor quality control, and application errors are all examples of data integrity issues”. The review on software quality forensics draws attention to research work to identify data integrity breaching approaches and mapping quality factors referenced with the forensics approach. It includes software quality factors that impact breaches in data integrity procedures due to software failure. This strategy concentrates time and resources on regions where data integrity breaches are most likely to be discovered.

TABLE IV. DATA INTEGRITY BREACH ISSUES

Intentional Data Integrity Breach	Unintentional Data Integrity Breach	Potential Data Integrity Breach
Falsified Data	Unsecured systems code and lack of versions control	Has the potential to be compromised
Reprocessed Data without controls	Lack of Audit Trails	Unsecured repository

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7630
***Article Received:** 01 June 2023 **Revised:** 28 July 2023 **Accepted:** 10 August 2023*

Operating outside of validation or registration parameters	Broken Security & software failure	Inappropriate user privileges
Misleading label claims	Unprocessed Data	Malicious code and acts

Modern software development is defined by rapid release and upgrade cycles, which are encouraged by agile development approaches. The foundational components of the Agile methodology, according to Gray [48], necessitate strict integrity checks; otherwise, attackers might insert malicious inputs that could potentially disrupt every step of the deployment pipeline. Page, Horsman, Sarna and Foster. [49] Unsafe design is one of the most frequent root causes of a vast class of application security vulnerabilities, sometimes known as "software and data integrity" issues. According to Horsman [50], due to improper validation brought on by software and data integrity problems, applications are susceptible to malicious code insertion, system compromise, and unauthorized information exposure. Johnson [51] explained that the use of outdated or unsupported third-party software, erroneous assumptions made in the server-side and client-side components being used, and software data integrity failure were to culpability, insufficient vulnerability screening procedures, and inadequate input validation across the pipeline are all examples of poor security practices in DevOps environment. Sengupta elaborates that software and data Integrity Failures [52] the absence of framework/platform patches, Lack of unit tests, and Insecure configurations of components leads to integrity breach. According to Hatton [53] Integrity refers to preserving the reliability of the evidence collection and analysis technique alongside the forensic investigators' moral character and professional integrity. It considers things like impartiality, objectivity, confidentiality, and respect for ethics and rules of the law. According to Guo, Slay, Beckett, & Watson. [44] System software confidence is increased by the application of technologies and methodologies such as software validation and verification.[45] The scientific rigor of the forensic investigator, the caliber of the analysis, and the correct use and operation of the computer forensic tools all contribute to the credibility of digital evidence. Software verification and validation and software inspection and testing are two possible ways, according to Fisher [53]. The "Computer Forensic Tool Testing (CFTT)" project at the "National Institute of Standards and Technology" designed and develop a methodology for digital forensics tools testing [53,54]. The capacity to ensure that forensic software tools consistently produce reliable, impartial test results that are admissible in court is a requirement [54,55]. Kebande and H. S. Venter raised the cloud platform challenges. [56] The integrity of the evidence is upheld. No research has up till now considered the broader possibilities or implications of careful software engineering on the creation of forensically sound systems. According to Ekanem [57] must focus on the software maturity index using Reusability, extensibility, dependability, portability, and innovativeness. These are the primary quality qualities that are required in the quality assessment of modern assets. Through its reliability of software depicted and useful to consider in the investigation. However, the effectiveness of the modernized applications, the difficulties involved with the procedure, and potential areas for development have not been investigated. There has been a significant gap found that needs to be filled.

Ouriques & et al. [58], raised the confidentiality and integrity issues of data in the software testing stages of the SDLC. Before running any test instances, it is crucial to consider a foundation examination measure. Arafeen &et al. [59] The average percentage of faults identified, “APFD”, would be a straightforward metric to evaluate a test case. Faster rates of fault or bug identification are associated with higher “APFD”. Kim & Kim [60] It has been stated that "Software quality-in-use" is challenging to quantify because it depends heavily on user perception and performance. Yongfeng [61] The software quality-in-use application service quality indicator (ASQI) has been created. The fundamental, limited, and optional "SRMs from ISO/IEC 9126" can be chosen for different software integrity levels using this method. Radhakrishnan, Shin &Ogale [62] This paper describes the data integrity spot detection and “Integrity Breach Conditions (IBCs)” that programmers employ to identify security holes that amend the values of sensitive data.

Banday [63] This article addresses how to distribute open-source software while maintaining its authenticity and integrity. A hacker might alter the software and add malicious codes that could result in a variety of security breaches. The recommendation according to Guveiy & Aktas.[64] At present the harmful consequences are to violate its confidentiality, integrity, and accessibility. These risks are caused by software flaws, which might allow third parties to access data or expose confidential information. First, a software life cycle analysis was done to identify the stages of software development to address this issue. Second, potential dangers to information were identified while considering the phases attained. Tarigopula [65] discussed the RADIUM (Race-free on-demand Integrity Measurement Architecture) architecture, an application-level integrity measurement solution that extends the trusted computing capability to the application layer. This is built on the idea of program invariants, which can be used to determine an application's proper behaviour. Using hypervisor-based introspection on RADIUM and a proof of concept for application-level measurement capability. Naks, Aiello, Taft, & Zheng [66] What is the software quality forensics model, and how is it derived? What are the techniques, challenges, and limitations of software quality forensics? What are the past and current scenarios related to software quality issues that require software-quality forensic and law enforcement direction? Describe in their case study the System-to-Software Integrity (SSI) process, which shows that the software developed complies with restrictions outlined in system requirements work

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7630
***Article Received:** 01 June 2023 **Revised:** 28 July 2023 **Accepted:** 10 August 2023*

in progress on a workflow. Zacchiroli and lamb.[67] To strengthen software supply chain integrity, this article investigates reproducible builds, a method that can evaluate whether created binaries match the source code. Khalid [68] This study will present a consensus model of software safety criteria that can be used to dissect and assign levels of quality to software safety integrity. Stephen &et al. by the study's [69] blockchain ledger is built on a graph database can ensure the integrity of code. Habli, Hawkins, Kelly [70] To assure software through the reliability of code quality that is commensurate with the requisite integrity. In this work, Zheng, Chunlin, Zhengyun & Naks. [71] investigate the software system's reliable boot and integrity measures. Data integrity and software integrity breaches about concerns with software quality were explained, nonetheless, after stating and discovering the associated literature review study.

RQ5 What are tools and standards that can facilitate software quality investigations?

The tools can be used for functional testing, non-functional testing, test management, performance testing, quality assurance, and API testing valuable in forensic investigation. It is important to consider the specific requirements mapping when choosing tool kits. Bogen & et al. [72] draw attention to the rapidly changing technological landscape, no single tool can fulfill all the requirements of a specific investigation.

A. Tools that can be used in software quality investigations

There are various tools, techniques, and standards can be utilized in software quality investigations. [73,74] Some of them Like “IBM QRadar SIEM”: QRadar SIEM will track each tactic and technique being used like threat actors trigger multiple detection analytics. Linting: Utilizing lint tools, you can fix source code flaws and enhance the caliber of your product. Static code analysis tools Find errors, weaknesses, and legal compliance problems in source code. “Kualitee”: Best for test case management and bug tracking. "Xray" The Manual & Automated Test Management App "TestRail" Effortlessly generates tests from requirements and bugs from tests. "Selenium" Open-source tool for automating web browsers. “Cucumber” A tool for behavior-driven development. "Testing Whiz" is An all-in-one test automation tool for web, mobile, cloud, and desktop applications. "Helix QAC & Klocwork" Makes it easy to ensure that your code is safe, secure, and high quality to analyze code. “Imperva Attack Analytics” is a tool that helps IT businesses respond to actual threats by being utilized in application security event investigations. “Cyber Triage” is an automated incident response tool used to quickly examine endpoints, gather pertinent data, and scan for malware and ominous behavior. “Malware Analysis (AX series)" offers a secure setting for characterizing, testing, and documenting sophisticated harmful actions.

B. Standards related to Quality Forensics

TABLE 5. STANDARDS RELATED TO QUALITY FORENSICS

Standards	Usage	Resources
“ISO/IEC 25010:2011”	To evaluate the quality of software systems in the context of software forensics, the SQuaRE (Software Product Quality Requirements and Evaluation) paradigm is used. a model of quality in usage.	[29,30,75]
“IEEE (1012–1998)”	Evaluation of a system or component through validation and verification.	[76]
“IEEE (1012-2016)”	Include product and process analysis, evaluation, review, inspection, assessment, and testing.	[77]
“ISO/IEC/IEEE 15288:2015”	outlines the specifications and recommendations for using the integration process, as well as how it relates to other system and software life cycle procedures.	[78]
“ISO/IEC/IEEE 12207:2017”	Presents system and software integration in full, considering interface, verification, and validation.	[78]
“Consortium for Information and Software Quality (CISQ)”	Proposes a set of standards that addresses software quality at the source code level, focusing on four factors: Reliability, Performance Efficiency, Security, and Maintainability.	[78]
“ISO (17025E)”	Defines validation as the verification through inspection and the supply of impartial proof that the conditions for a certain intended usage are met.	[78,79]
“2021-06-17 SWGDE”	Establishing a Quality Management System for a Multimedia and Digital Organization.	[80]
“ISO/IEC 25020-24”	Measures of software product quality, as well as quality in usage and life cycle, are common to quality measure aspects.	[81,82]
“ISO/IEC 27043:2015”	Include a process model to get ready for digital forensic investigations.	[82,83]

Software quality factor models can be used in quality forensics to evaluate and assess the quality of software systems that are involved in forensic investigations. Although there is not a specific model designed exclusively for quality forensics to adapt and apply established software quality factor models and mapping with related standards to accommodate quality investigation with context.

IV. RESULTS AND DISCUSSION

A. Software Quality Forensics Challenges Followed by the research questions discussed we have derived the following challenges

• Software Quality Model Mapping: The specific requirements mapping of quality factors in the rapidly changing technological landscape So no single Software quality model

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7630
***Article Received:** 01 June 2023 **Revised:** 28 July 2023 **Accepted:** 10 August 2023*

and framework can handle all the requirements of a specific and complex investigation.

• Standard Development Dedicated to Quality Forensics: It is a well-known issue that there is no adequate model for software quality. The evidence must be "complete, accurate, trustworthy, and attainable in a replicable manner" to be admissible in court as required by software quality forensics.

• Verification & Validation: Verifying that current software systems meet forensic readiness standards is a substantial task. Maintaining the validity and integrity of open-source software distribution is difficult.

• Integrity breach identification: Code Integrity is the accuracy, consistency, completeness, and dependability of code throughout the lifecycle. Handling breaches at each level is a challenging task, and as a result, susceptibility results in software and data integrity failures. The data integrity spot detection and Integrity Breach Conditions that investigators employ to identify security holes that modify sensitive data values.

• Underlying Dependencies: Previous studies on the quality level have mostly focused on the tool's primary codebase rather than its underlying dependencies.

• Track quality behavior of application: The data integrity breaches over the inability to track the quality behavior of the application, what gets stolen, how it happens and where is the destination of data. It is a very complex task to discriminate app behavior for bypassing breached integrity data.

• Software quality testing: It is a very challenging task to maintain test cases and their process with considering quality standard mapping with alleged software investigation.

• Evidence-based model: Evidence-identified model for software quality failures investigation, gathering, and root-cause analysis are tough processes. The versions of alleged software are dynamic so must focus on the history of software and the change that has been made and must follow the digital evidence process and prosecution.

B. Limitation

Through literature review we can point out the following limitations as follows:

• Dedicated to software quality forensics very less research and guidance existed.

• Cross-platform investigation lawsuit deficiency.

Forensics validation of Tools and techniques dedicated to quality forensics.

• Existence of standards closely related to quality forensics.

• High cost and time occupied for investigation of alleged software.

• Less legally verified automation tools to satisfy the law of court.

V. CONCLUSION

This review paper derived the keyword Software quality forensics and discusses the need for a Software quality forensics model is a cutting-edge method to build forensic models to deal with legal challenges relating to alleged software. This research paper provides a comprehensive examination of software quality forensics, encompassing its techniques, challenges, and limitations. The paper is based on research questions in the novel realm of software quality forensics. The paper explores approaches, tools, techniques, and standards regarding software quality investigation and confines the research area for software quality integrity breach forensics as guidance to futuristic model development for software quality forensic cross-platform.

ACKNOWLEDGMENT

Not Applicable

CONFLICT OF INTEREST

There is conflict on interest

REFERENCES

[1] "What you need to know about software liability," Insureon.com. Available:https://www.insureon.com/blog/what-you-need-to-know-about-software-liability.

[2] Archive.org. [cited,2023 July 14]. Available from: https://web.archive.org/web/20071212183729/http://neptune.netcomp.monash.edu.au/cpe9001/assets/readings/www_uguelph_ca_~tgallagh_~tgallagh.html.

[3] V. Bhattathiripad, Judiciary-Friendly Forensics of Software Copyright Infringement, Appendix 1: Challenges in Software Quality Forensics and Litigation-A Case Study. IGI Global, 2014.

[4] Alan Gillies Gillies, "Software Quality: Theory and Management," Morrisville, NC, USA: Lulu Press, 2011.

[5] A.J. Suali, S. S. M. Fauzi, Mhmwawm Nasir, and I. K. Raharjana, "Software Quality Measurement in Software Engineering Project: A Systematic Literature Review," Journal of Theoretical and Applied Information Technology, vol. 97, pp. 918–929, 2019.

[6] D. Winkler, S. Biffl, D. Mendez, and M. Wimmer, "Software Quality: Future Perspectives on Software Engineering Quality: 13th International Conference, SWQD 2021," J. Bergsmann, Ed. Vienna, Austria; New York, NY, USA: Springer International Publishing, 2021.

[7] G. Lacerda, F. Petrillo, M. Pimenta, and Y. G. Guéhéneuc, "Code smells and refactoring: A tertiary systematic review of challenges and observations," Journal of Systems and Software, vol. 167, p. 110610, 2020. Available: http://dx.doi.org/10.1016/j.jss.2020.110610.

[8] I.Atoum, "A novel framework for measuring software quality-in-use based on semantic similarity and sentiment analysis of software reviews," Journal of King Saud University: Computer and Information Sciences, vol. 32, no. 1, pp. 113–125, 2020.

[Online]. Available: http://dx.doi.org/10.1016/j.jksuci.2018.04.012.

[9] T. Hynninen, J. Kasurinen, and O. Taipale, "Framework for observing the maintenance needs, runtime metrics and the overall quality-in-use," Journal of Software Engineering and Applications, vol. 11, no. 04, pp. 139–152, 2018. [Online]. Available: http://dx.doi.org/10.4236/jsea.2018.114009

[10] M. N. Alim, T. Hapsari, and L. Purwanti, "The Effect of Competence and Independence on Audit Quality with Auditor Ethics as Moderating Variables," in National Accounting Symposium X, Makassar, Indonesia, 2007.

[11] L. Pasquale, D. Alrajeh, C. Peersman, T. Tun, B. Nuseibeh, and A. Rashid, "Towards forensic-ready software systems," in Proceedings of the 40th International Conference on Software Engineering: New Ideas and Emerging Results, New York, NY, USA: ACM, 2018.

[12] Jan H. P. Eloff &Madeleine Adrienne Bihina Bella, "Methodology for Investigating Software Failures Using Digital Forensics and Near-Miss Analysis," Software Failure Investigation. pp.39–56, 2018. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-61334-5_4

[13] M. A. B. Bella and J. H. P. Eloff, "A near-miss management system architecture for the forensic investigation of software failures," Forensic science international, vol. 259, pp. 234–245, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.forsciint.2015.10.007.

[14] A.Bassey and J. Ekanem, "Application of Reverse Engineering Technique in Software Forensic Analysis to Detect Infringements," in Proceedings of the World Congress on Engineering 2021, WCE 2021, 2021.

[15] P. Juniawan, "E-Voting Software Quality Analysis with McCall's Method," in 8th International Conference on Cyber and IT Service Management (CITSM), 2020, pp. 1–5.

[16] G. L. Saini, D. Panwar, S. Kumar, and V. Singh, "A systematic literature review and comparative study of different software quality models," Journal of Discrete Mathematical Sciences and Cryptography, vol. 23, no. 2, pp. 585–593, 2020. [Online]. Available: http://dx.doi.org/10.1080/09720529.2020.1747188.

[17] E. Guveyi, M. S. Aktas, and O. Kalipsiz, "Human factor on software quality: A systematic literature review," in Computational Science and Its Applications - ICCSA 2020, Cham: Springer International Publishing, 2020, pp. 918–930.

[18] S. S. Rathore and S. Kumar, "An empirical study of ensemble techniques for software fault prediction," Applied Intelligence, vol. 51, no. 6, pp. 3615–3644, 2021. [Online]. Available: http://dx.doi.org/10.1007/s10489-020-01935-6

[19] S. Oveisi, M. A. Farsi, M. Nadjafi, and A. Moeini, "A New Approach to Promote Safety in the Software Life Cycle," Journal of Computer & Robotics, vol. 12, no. 1, pp. 77–91, 2019.

[20] C. Johnson, "Forensic software engineering: Are software failures symptomatic of systemic problems?" Computer science, vol. 40, no. 9, pp. 835–847, 2002. [Online]. Available: http://dx.doi.org/10.1016/s0925-7535(01)00086-8.

[21] L. Hatton, "Forensic Software Engineering: An Overview," 2004. [Online]. Available: https://www.leshatton.org/Documents/fse_Dec2004.pdf

[22] D. Rowley and S. Ramakrishnan, "Forensic applications of software analysis," in Law and Technology, Taipei, 2002, pp. 7–9.

[23] M. Abu Talib, "Towards early software reliability prediction for computer forensic tools (case study)," Springerplus, vol. 5, no. 1, p. 827, 2016. [Online]. Available: http://dx.doi.org/10.1186/s40064-016-2539-0

[24] E. Imwinkelried, "Computer Source Code: A Source of the Growing Controversy Over the Reliability of Automated Forensic Techniques," 2002.

[25] L. Daubner, R. Matulevičius, and B. Buhnova, "A model of qualitative factors in forensic-ready software systems," in Lecture Notes in Business Information Processing, Cham: Springer Nature Switzerland, pp. 308–324,2023.

[26] R. Khelifi, R. Alnanih, and M. Abu Talib, "Application of quality in use model to assess the user experience of open-source digital forensics tools," International Journal of Electronic Security and Digital Forensics, vol. 12, no. 1, pp. 1–19, 2020. [Online]. Available: http://dx.doi.org/10.1504/ijesdf.2020.10025165.

[27] G. Tully, N. Cohen, D. Compton, G. Davies, R. Isbell, and T. Watson, "Quality standards for digital forensics: Learning from experience in England & Wales," Forensic Science International: Digital Investigation, vol. 32, p. 200905, 2020. [Online]. Available: http://dx.doi.org/10.1016/j.fsidi.2020.200905

[28] P. Sommer, "Accrediting digital forensics: What are the choices?" Digital Investigation, vol. 25, pp. 116–120, 2018. [Online]. Available: http://dx.doi.org/10.1016/j.diin.2018.04.004

[29] ISO/IEC/IEEE Approved Draft Standard - Systems and Software Engineering -- Life Cycle Management -- Part 6: Systems and Software Integration. ISO/IEC/IEEE P24748-6/FDIS. 2023; 9:1–56.

[30] ISO 25010, "Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE) - System and software quality models," 2011.

[31] B. W. Boehm, J. R. Brown, and M. Lipow. quantitative evaluation of software quality. International Conference on Software Engineering, Proceedings of the 2nd international conference on Software engineering(2nd):592– 605, 1976.

[32] J. P. Cavano and J. A. McCall, "A framework for the measurement of software quality," Perform Eval Rev, vol. 7, no. 3–4, pp. 133–139, 1978. [Online]. Available: http://dx.doi.org/10.1145/1007775.811113

[33] J. P. Miguel, D. Mauricio, and G. Rodriguez, "A review of software quality models for the evaluation of software products," arXiv [cs.SE], 2014.

[34] R. Glott, "Quality models for Free/Libre Open-Source Software-towards the 'Silver Bullet'," in EUROMICRO Conference on Software Engineering and Advanced Applications IEEE Computer Society, 2010, pp. 439–446.

[35] A. Madaehoh and T. Senivongse, "OSS-AQM: An open-source software quality model for automated quality measurement," in 2022 International Conference on Data and Software Engineering (ICoDSE), IEEE, 2022.

[36] M. MiloševićIvana BjelovukTanja Kesić,"Quality management system in forensic laboratories," [Online]. Kriminalističko-policijska akademija, Beograd,2009, Available: https://core.ac.uk/download/pdf/196617864.pdf

[37] "Forensic Science research and development technology working group: Operational requirements," National Institute of Justice. [Online]. Available:https://nij.ojp.gov/topics/articles/forensic-science-

research-and-development-technology-working-group-operational.

[38] S. Doyle, "A review of the current quality standards framework supporting forensic science: Risks and opportunities," WIREs Forensic Science, vol. 2, no. 3, 2022. [Online]. Available: http://dx.doi.org/10.1002/wfs2.1365

[39] P. Behnamghader, R. Alfayez, K. Srisopha, and B. Boehm, "Towards a better understanding of software quality evolution through commit-impact analysis," in 2017 IEEE International Conference on Software Quality, Reliability and Security (QRS), IEEE, 2017.

[40] A. Alvaro, E. S. de Almeida, A. M. L. de Vasconcelos, and S. R. de Lemos Meira, "Towards a software component quality model," Available:https://www.cin.ufpe.br/~if723/seminarios-2005-1/Component%20Quality%20Model-v2

[41] N. Saari, "Software Supply Chain Monitoring in Containerised Open-Source Digital Forensics and Incident Response Tools," 2022.

[42] D.Brecht, "Computer crime investigation using forensic tools and technology, "Available:https://resources.infosecinstitute.com/topic/computer-crime-investigation-using-forensic-tools-and-technology/

[43] J. Brunty, "Validation of forensic tools and methods: A primer for the digital forensics' examiner," WIREs Forensic Science, 2022. [Online]. Available: http://dx.doi.org/10.1002/wfs2.1474

[44] Y. Guo, J. Slay, J. Beckett, and T. Watson, "Validation and verification of computer forensic software tools—Searching Function," Digital Investigation, vol. 6, pp. S12–S22, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.diin.2009.06.015

[45] "New Approaches to Digital Evidence Processing and Storage," Grants.gov announcement number NIJ, 2014.

[46] D. Rowley and S. Ramakrishnan, "Forensic applications of software analysis," in Law and Technology, ACTA Press, pp. 7–9, 2002.

[47] B.C. Kreitzberg, B. Shneiderman. "FORTRAN Programming: A Spiral Approach. FORTRAN Programming: A Spiral Approach." 1982.

[48] A.Gray, "Software Forensics: Extending Authorship Analysis, Techniques to Computer Programs", the Information Science Discussion. 1997.

[49] H. Page, G. Horsman, A. Sarna, and J. Foster, "A review of quality procedures in the UK forensic sciences: What can the field of digital forensics learn?" Science Justice, vol. 59, no. 1, pp. 83–92, 2019. [Online]. Available: http://dx.doi.org/10.1016/j.scijus.2018.09.006

[50] G. Horsman, "Tool testing and reliability issues in the field of digital forensics," Digital Investigation, vol. 28, pp. 163–175, 2019. [Online]. Available: http://dx.doi.org/10.1016/j.diin.2019.01.009.

[51] Johnson C. Forensic software engineering: are software failures symptomatic of systemic problems? Computer science.2002;40(9):835–47.Avail,http://dx.doi.org/10.1016/s0925-7535(01)00086-8.

[52] S. Sengupta, "Software and Data Integrity Failures - examples & prevention," Crashtest Security, 2022. [Online]. Available: https://crashtest-security.com/owasp-software-data-integrity-failures/

[53] M. S. Fisher, "Software verification and validation: An engineering and scientific approach," Springer Science & Business Media, 2007.

[54] "Contacting NIST | NIST," 2019. [Online]. Available: https://www.nist.gov/about-nist/contact-us

[55] "Computer forensics tool testing program (CFTT),", Available: https://www.unodc.org/e4j/data/_university_uni_/computer_forensics_tool_testing_program_cftt.html?lng=en&match.

[56] Pratikso, Abdul Rochim, Rachmad Mudiyono, Adolf Situmorang. (2023). Stabilization of Expansive Soil with Lime, Fly Ash and Cement. International Journal of Intelligent Systems and Applications in Engineering, 11(4s), 491–497. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2706

[57] V. R. Kebande and H. S. Venter, "On digital forensic readiness in the cloud using a distributed agent-based solution: issues and challenges," Aust J Forensic Sci, vol. 50, no. 2, pp. 209–238, 2018. [Online]. Available: http://dx.doi.org/10.1080/00450618.2016.1194473.

[58] A.B. Ekanem, "Assessment of Components Stability for Modernization Using Software Maturity Index," International Journal of Scientific Research and Engineering Studies (IJSRES), vol. 2, no. 12, 2015.

[59] J. Ouriques, S. Felipe, G. Emanuela, and P. D. Cartaxo, "On the Influence of Model Structure and Test Case Profile on the Prioritization of Test Cases in the Context of Model-Based Testing," in Software Engineering (SBES), 2013.

[60] J. M.Arafeen, H.Do., "Test case prioritization using requirements-based clustering", In 2013 IEEE Sixth International Conference on Software Testing, Verification, and Validation. IEEE; 2013.

[61] S.H. Kim and W.J. Kim, "Evaluation of software quality-in-use attributes based on analysis network process," Cluster Computing, vol. 22, no. S1, pp. 2101–2114, 2019. [Online]. Available: http://dx.doi.org/10.1007/s10586-018-2309-6

[62] Y. Yongfeng, "Software Reliability Metrics Selecting Method Considering Software Integrity Level," Computer Science, vol. 38, no. 1, pp. 145–149, 2011.

[63] Robert Roberts, Daniel Taylor, Juan Herrera, Juan Castro, Mette Christensen. Enhancing Collaborative Learning through Machine Learning-based Tools. Kuwait Journal of Machine Learning, 2(1). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/177

[64] R. P. Radhakrishnan, M. Shin, and P. Ogale, "Data integrity security spots detected by object reference," in 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), IEEE, 2021.

[65] M. Banday, "Ensuring Authentication and Integrity of Open-Source Software using Digital Signature," International Journal of Computer Applications NSC, vol. no. 4, pp. 11–14, 2011.

[66] V. Guveiy and T. Aktas, "Secure software developing recommendations," in 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science, and Technology (PIC S&T), IEEE, 2019.

[67] M. Tarigopula, "Integrity Verification of Applications on Radium Architecture," 2015.

[68] T. Naks, M. A. Aiello, S. T. Taft, and N. Zheng. "Using SPARK to ensure system to Software Integrity: A case study" ACM SIGAda Ada Letters [Internet]. 2020;40(1):74–8. Available from: http://dx.doi.org/10.1145/3431235.3431241.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7630
***Article Received:** 01 June 2023 **Revised:** 28 July 2023 **Accepted:** 10 August 2023*

[69] C.Lamb, S.Zacchiroli "Reproducible builds: Increasing the integrity of software supply chains. IEEE Software". 2022;39(2):62–70. Available from: http://dx.doi.org/10.1109/ms.2021.3073045.

[70] T. Khalid, "Dependability Model for Decomposition and Allocation of System Safety Integrity Levels of Software Quality International Review on Computers and Software". 2015; 10:1110–9.

[71] D.Stephen, H.Kim, A.Kim, "A Study of Blockchain based on Graph Database for Software Quality Measurement Integrity".Information and Communication Technology Convergence. 2018.

[72] I. Habli, R. Hawkins, T. Kelly. "Software safety: relating software assurance and software integrity", International Journal of Critical Computer-Based Systems.2010;1(4):364. Available from: http://dx.doi.org/10.1504/ijccbs.2010.036605.

[73] Y. Zheng, Y. Chunlin, F. Zhengyun, Z.Na " Trust chain model and credibility analysis in software systems." In: 2020 5th International Conference on Computer and Communication Systems (ICCCS). IEEE; 2020.

[74] Bogen and D. Dampier, "Unifying computer forensics modeling approaches: a software engineering perspective," in Proceedings of the first international workshop on systematic approaches to digital forensic engineering, Taipei, pp. 7–9, 2005.

[75] "Guide to software quality," Perforce Software. [Online]. Available: https://www.perforce.com/resources/guide-to-software-quality.

[76] WWW.g2.com. [cited 2023]. Available from: https://www.g2.com/categories/digital-forensics

[77] ISO/IEC 25010:2011 [Internet]. ISO. 2019 [cited 2023 Jul 14]. Available from: https://www.iso.org/standard/35733.html

[78] IEEE (1012–2004), the Draft standard for software verification and validation IEEE P1012/D12.

[79] IEEE 1012-2016 [Internet]. IEEE Standards Association. IEEE SA; [cited 2023 June 14]. Available from: https://standards.ieee.org/ieee/1012/5609/

[80] B.Geary. Software quality testing: The parameters, methodologies, and tools [Internet]. Andplus.com. [cited 2023 Jul 14]. Available from: https://www.andplus.com/blog/software-quality-testing.

[81] ISO.org. [cited 2023 Jun 11]. Available from: https://www.iso.org/ISO-IEC-17025-testing-and-calibration.

[82] SWGDE "Establishing a Quality Management System for a Digital and Multimedia Organization under ISO," 2021.

[83] ISO 25000 standards [Internet]. Iso25000.com. [cited 2023 Jul 14]. Available from:https://iso25000.com/index.php/en/iso-25000-standards.

[84] ISO - ICS [Internet]. ISO. 2019 [cited 2023 Jul 14]. Available from: http://www.iso.org/iso/iso_catalogue/

[85] www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=44407.