Journal of
Information Systems Engineering
and Business Intelligence

Vol.11, No.2, June 2025
Available online at: http://e-journal.unair.ac.id/index.php/JISEBI

Exposing Causative Factors on Software Discontinuity using an Elaborative Qualitative Method

Arfive Gandhi[1)*], Dana Sulistiyo Kusumo [2)], Indra Lukmana Sardi [3)]

[1)2)3)] *School of Computing, Telkom University, Bandung, Indonesia*

[1)]arfivegandhi@telkomuniversity.ac.id, [2)] danakusumo@telkomuniversity.ac.id, [3)]indraluk@telkomuniversity.ac.id

Abstract

Background: Software discontinuity due to the inability to accommodate the needs of users is a the significant challenge facing the software development life cycle. This implied that the development team must be capable of producing software with extended lifespan, including the ability to detect outages early, to maintain continuity. Organizations need to determine the contributing and inhibiting factors responsible for discontinuity usage.

Objective: This research aimed to explore the factors that contribute and inhibit the discontinuation of software use in organizations as well as the prevention strategies.

Methods: The summative content analysis technique was used to capture, codify, and classify statements from respondents to discover usage pattern. Data were collected through interview and questionnaire techniques with 10 respondents from various Indonesian companies. The respondents had various sectoral backgrounds in software usage for more than a year. The data collected were compared, contrasted, and synthesized to deliver a holistic pattern among respondents.

Results: The result showed that 10 key factors contributed to software discontinuity, namely Loss of Perceived Usefulness (LUS), Loss of Perceived Ease of Use (LEU), Decreased Effort Expectancy (DEX), Decreased Performance Expectancy (DPX), Social Influence (SOI), Lack of Facilitating Conditions (LFC), Decreased Price Value (DPV), Lack of Habit (LHB), Hedonic Motivation (HDM), and Loss of Perceived Behavioral Control (LBC). The factors were further categorized into three big issues, including Software Usability (LUS, LEU, DEX, and DPX), External Triggers (DPV, SOI, and LBC), and Risk Management after Discontinuity (LFC, LHB, and SOI). Furthermore, the results indicated that nine factors contributed to software discontinuity except HDM with LEU and LUS having weak significance since most respondents stated partial agreement and disagreements.

Conclusion: This research employed a rigorous qualitative method to validate the factors in the proposed software discontinuity model with 10 causative factors. The acquired knowledge is expected to aid organizations or related development units to build software that accommodates user needs, including meeting long-term business targets.

Keywords: Software, Software Discontinuity, Influencing Factors, Qualitative Method

Article history: Received 19 March 2024, first decision 23 August 2024, accepted 14 February 2025, available online 22 July 2025

I. Introduction

Discontinuity, described as termination of software usage, is a significant challenge encountered in the software development life cycle. It signified pressure on continuity during the development and maintenance phases [1] as well as outlined the crucial role of software in satisfying stakeholders' needs [2] through process optimization, value creation, and knowledge-making. In addition, discontinuity was referred to as a natural constituent of a product's lifecycle, indicating the inability of the software to accommodate user's needs. It is also an inadequate part of Information Technology (IT) investment [3], posing a threat to the software development organization/unit as users may no longer adopt related services, thereby impacting business continuity. Software discontinuity also increases user churn, regarded as the numbers of users who abandon the application [2], due to its ineffectiveness [4]. Although Turel reported that information systems (larger context than software) discontinuances were prevalent situations [5], it distinguished major failure to fulfilling users' requirement after the adoption process. Liang et al. also reported the easiness of smartphone users to delete or uninstall software [6]. In line with these findings, software discontinuity reduced stakeholders' satisfaction, hampering the positive contribution of IT investments. Several academicians explored software adoption processes without reporting a detailed discontinuity procedure. This research outlined a major gap since many previous investigations focused on the urgency and importance but failed to identify the causative factors.

[*] Corresponding author

ISSN 2443-2555 (online) 2598-6333 (print)
doi: http://dx.doi.org/10.20473/jisebi.11.2.103-114

Software discontinuity must be detected as early as possible during maintenance, to ensure the survival of business on software sales [7], [8]. The experience should be elaborated to generate new knowledge aimed at maintaining continuity, including updating the software. Learning from this experience plays a valuable role in producing or developing software in the future. The mandated qualities must pay attention to long-term sustainability, monitored through quality assessment [9] while anticipating the risk of being discontinued immediately.

This research aimed to explore the contributing and inhibiting factors responsible for influencing software discontinuity usage. The exploration of both factors enabled organizations to adopt software that met user needs and in line with long-term business targets. An elaborative qualitative method, requiring the gathering of textual data from practitioners was adopted. The method elaborated on ideas expressed based on experiences about software discontinuity processes, and the identification of the influencing factors from an empirical perspective.

Considering the description above, the current research has two main objectives first, it aimed to explore the factors contributing to the discontinuation of software used in organization, including improvement strategies. Second, the factors inhibiting software discontinuity usage in organizations, as well as exploitation strategies, were investigated. Moreover, Olsson, Sentilles, and Papatheocharous [10] conducted a research focusing on the importance of evaluating software quality in diverse phases, including Mishra et al. [11]. These research inspired the assessment of factors influencing software discontinuity based on quality fulfillment. Several prospective theoretical benefits were also realized through the purposes, namely (1) Expansion of software use factors dominated by identification in the adoption and retirement phases, (2) Enrichment of software quality theory through identification of factors detected during operation, and (3) Enrichment of requirements engineering theory through recommendations for factors anticipated in identifying software requirements.

The novelty of this discussion focused on filling the following gaps, firstly, previous research conducted by Almaiah et al. [12], Na et al. [13], and Qader et al. [14] centered on influencing factor identification dominated by the adoption phase. These research reported how people were determined to use the software for the first time, outlining guidelines for product managers to attract new users but failed to maintain continuity. Secondly, the research by Pilliang and Munawar [15], focused on expanding risk management in software engineering, which played a crucial role in the development phase but lacked maintenance concern. It further established awareness of risk management for the maintenance phase in anticipation of software discontinuity as a significant risk. Thirdly, previous research associated software failure with technical issues in related testing processes. This extended the concept of software failure as the initial point of discontinuity, considered an enormous possibility by identifying both technical and non-technical factors. Fourthly, Furneaux et al. (2010, 2020), proposed the Information System discontinuance theory, which was later updated [16]. The theory focused on why Information System was discontinued by the users, in respect to data, and technological related processes. However, the theory failed to specify how users handled software discontinuity since it was vast. Finally, Berger and Kompan used churn prediction as a typical example [8], relying on the quantitative value from behavior recorded on the website. The generated value missed the churn users' opinions, causing the analysis to lack causative factors identification. This prompted the current research to exploit the primary data gathered based on experiences using profound collection methods.

Asides from the description above, two promising practical implications were offered first, the software house was used to portray the factors influencing discontinuity as guidelines by ensuring the functional and non-functional requirements of the developed software. In line with this description, lack of functionality drove users to abandon the software [6], opting for another designed by competitors. The requirements also minimized certain factors through feasible, and viable features, thereby enabling the qualification of the software [10]. Additionally, the developer team was expected to carefully review the Software Requirement Specification document. Second, the factors described was used to monitor and control the adopted software, ensuring continuity.

II. Literature Review

A. Technology Acceptance Model

The Technology Acceptance Model (TAM), is a theory that focused on the acceptance of IT systems [13] by users. TAM was developed by Davis et al. based on the TRA (Theory of Reasoned Action) model. This theory reportedly included two main constructs influencing behavioral intention to the TRA model, namely Perceived Usefulness and Perceived Ease of Use. Both were regarded as the first and second suspected factors through the following label adjustment, Loss of Perceived Usefulness (LUS) and Loss of Perceived Ease of Use (LEU). According to Qader et al. [14], Perceived Usefulness referred to the degree or extent people believed the adoption of a particular technology would improve job performance. In line with the definition, TAM outlined the use of information systems due to its usefulness, and reverse reported if otherwise. The premises were applicable for Ease of Use in this context, with the intent to adopt effortless technology. The other components of TAM included Attitude Toward Using Technology, Behavioral Intention to Use, and Actual Technology Use.

B. Unified Theory of Acceptance and Use of Technology

UTAUT (Unified Theory of Acceptance and Use of Technology), refers to a conceptual framework used to understand the factors influencing individual acceptance and use of technology [17]. This theory was developed to explain why people accepted and adopted new technologies. In addition, the theory combined elements from several previous technology acceptance theories, including TAM and other concepts [18], [19]. UTAUT is characterized by the following suspected factors (1) Effort Expectancy (individual easiness in using technology), (2) Performance Expectancy (the belief that using technology would improve performance or efficiency in specific tasks), (3) Social Influence (external pressure from the social environment to use specific technology), and (4) Facilitating Conditions (external factors that support or hinder the use of technology, such as the availability of technical support). This also included (5) Price Value, (6) Habit, (7) Hedonic Motivation, (8) Behavioral Intention to Use (the intention to use technology), and (9) Previous Experience. Building upon these, Decreased Effort Expectancy (DEX), Decreased Performance Expectancy (DPX), Social Influence (SOI), Lack of Facilitating Conditions (LFC), Decreased Price Value (DPV), Lack of Habit (LHB), and Hedonic Motivation (HDM) were selected as a reflection of reverse factors from existing theories.

C. Theory of Planned Behavior

The Theory of Planned Behavior (TPB) was used to explain certain factors responsible for influencing the intent to engage in specific tasks [20], [21], including adopting software [22] and information systems [5]. In 2016, Turel [5] used the TPB to disclose factors influencing information system discontinuance through quantitative method. These factors positively contributed [20] to Attitudes, Subjective Norms, and Perceived Behavioral Control. This research reported that a negative contribution would reduce the intention to continue using the software. Liang et al. featured SOI as a contributing factor of user smartphone's application abandonment [6]. Therefore, any lack affected the decision to either discontinue or stop using the software. Attitude and Subjective Norms were coherent with Habit (eighth suspected factor) and SOI (fifth one), respectively. In with these findings, this research considered Loos of Perceived Behavioral Control (LBC) as the tenth suspected factor.

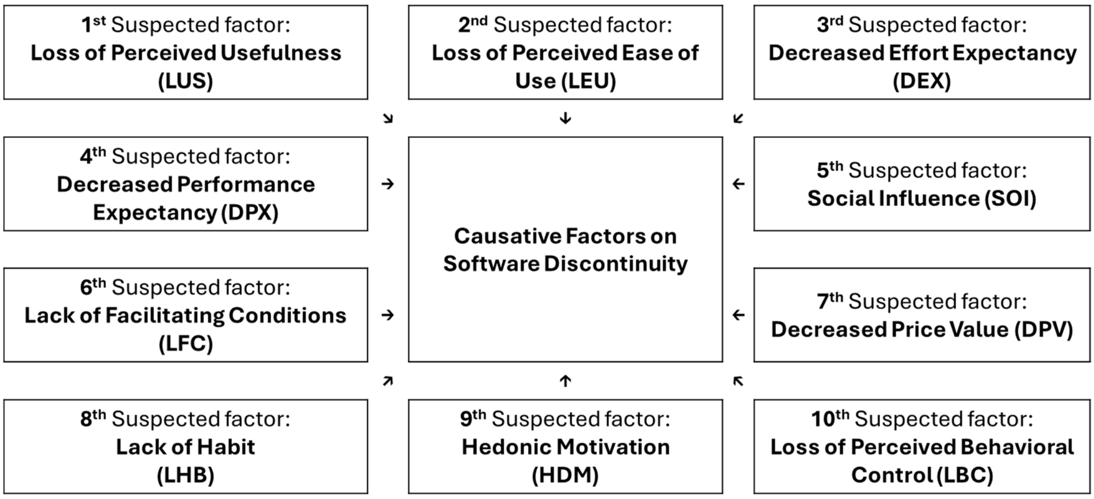

Fig. 1 Suspected factors in software discontinuity.

D. Adoptable Standards on Software Engineering

The research adopted related standards on software engineering, to improve the proposed discontinuity model. The selected standards included ISO/IEC 25010:2023 and ISO/IEC 9126-1:2001. Furthermore, the ISO/IEC 25010:2023 was referred to as Systems and Software Engineering Systems, constituting of Software Quality Requirements and Evaluation (SQuaRE) Product quality model [23]. The research relied on UTAUT, TAM, ISO/EIC 25010:2023, and ISO/EIC 9126-1 as baselines to generate instruments considered as manifestation of each factor or variable. ISO/IEC 9126-1:2001 depicted software engineering - product quality - part 1 quality model [24]. It comprised the following criteria for software quality Effectiveness, Efficiency, Satisfaction, Freedom of Risk, and Context Coverage.

Effectiveness and Satisfaction were strongly related to Perceived Usefulness (first suspected factor) and HDM (ninth one) from TAM and UTAUT2, respectively. Simultaneously, Efficiency was coherent with Perceived Ease of Use (second one) and Effort Expectancy (third one) from TAM and UTAUT2. Freedom of Risk was associated with Perceived Behavioral Control (tenth one).

Finally, 10 suspected factors influencing software discontinuity in businesses were obtained. This included Loss of Perceived Usefulness (LUS), Loss of Perceived Ease of Use (LEU), Decreased Effort Expectancy (DEX), Decreased Performance Expectancy (DPX), Social Influence (SOI), Lack of Facilitating Conditions (LFC), Decreased Price Value (DPV), Lack of Habit (LHB), Hedonic Motivation (HDM), and Loss of Perceived Behavioral Control (LBC). Fig. 1 shows its existence as suspected factor following the extraction process in literature review.

III. METHODS

A. Research Paradigm and Classification

The actualization of a specific research type followed the classification process proposed by Saunders et al. [25]. This subsection served as the baseline for explaining how the research should be performed in line with specific characteristics. Each layer defined the research type from Philosophies to the Time-Horizon. However, in Philosophies, constructivism and interpretivism were adopted because the results were compiled based on respondents' understanding [26], [27], with the truth obtained from multiple viewpoints [28], [29]. The findings were summarized at the end, depicting the use of an inductive method, alongside the survey strategy and categorization shown in Fig. 2. Qualitative method was also adopted, following the research conducted by Creswell [30]. This method enabled profound data collection since respondents were expressive, [31]. The prompted the comparison of the collected data, in order to capture an established pattern.

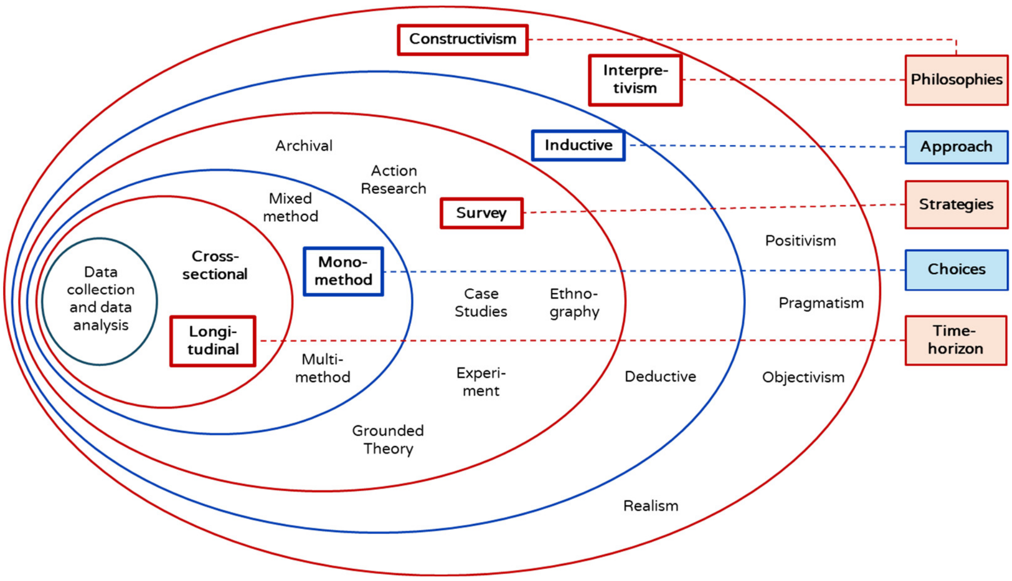

Fig. 2 Research classification following its paradigm.

Based on the research objectives, the causative factors were explored through a survey strategy. This included identifying the contributing and inhibiting factors associated with software discontinuity in an office environment. The qualitative method was applied in respect to main phases, namely (1) Instrument generation from related theories in literature review, (2) Primary data collection (including respondent recruitment and data capturing), and (3) Results elaboration and interpretation. The codification technique was further used to carry out the content analysis, elaborating the primary data into identified patterns. The statement of each respondent was classified under similar categories, and extracted to determine the causative factors by comparing, contrasting, and synthesizing these attributes.

B. Instrument Generation

TAM, UTAUT2, and TPB were elaborated on, and served as the baseline since its qualified components were empirically proven as causative factors affecting organizations in respect to technology adoption. This was based on the inference that people stopped the use of a particular software, after neglecting the causative factors. The instruments were consisted of two main parts, namely demography persona and experience capturing. In the first part, the demographic persona focused on the respondents' profile, showing respective diversity. The demographic attributes were captured for personal information only, not as influencing factors. Moreover, the demographic attributes aided in obtaining insight into the respondents' background suitability and eligibility, enabling further interpretation of the generated results. The second part comprised 13 open-ended questions which elaborated the respondents' experience reflecting individual perception following the instruments. It actualized the instruments cascaded from suspected factors as stated in the Literature Review and in Fig. 1. This second part enabled data collection by matching required information from the instrument with respondents' point of view. A pivot testing of the questionnaire was conducted before being distributed to the respondents to ensure its readability. In line with these findings, Table 1 shows the generated instruments with sources.

TABLE 1
LIST OF INSTRUMENTS

Code	Question Statement	Source
LUS.01	When your office stopped using software, did the perceived benefit decrease so that work goals became less achieved? Give examples of the reduced benefits.	[13][14]
LEU.01	When your office stopped using software, how far did the software's usage complexity influence the decision to stop? Give an example of the complexity.	[13][14]
LEU.02	When your office stopped using software, was it influenced by a lack of software usability that made your work inefficient or unproductive? Give examples of inefficient/unproductive work due to using software.	[13][14]
EFE.01	When your office stopped using software, how far was the gap between the ease of use of the software that you expected and the reality?	[17]
DEX.01	When your office stopped using software, how far did software have a decrease in performance? Give examples of performance degradation.	[17]
SOI.01	How far did your boss influence your/your unit's decision to stop using software? Give a concrete example!	[17]
SOI.02	After you/your unit stopped using software, how did your boss provide a solution for its impacts? Give a concrete example!	[17]
LFC.01	When your office stopped using software, did the related parties stop its technical or moral support? Give examples of missing technical or moral support.	[17]
DPV.01	When your office stopped using software, how far did the price factor influence this decision? Were there any price increases that made you/your unit unable to continue using it?	[17]
LHB.01	When your office stopped using software, did experience using technology or other software influence the decision to stop using the software?	[17]
HDM.01	When your office stopped using software, was that decision influenced by a lack of pleasure during its usage? Give an example of lost pleasure.	[17]
LBC.01	How confident/confident were you (was your unit) when you/your unit stopped using the software? Give a concrete example!	[20]
LBC.02	How confident were you (was your unit) in finding a solution after stopping software usage? Give a concrete example!	[20]

C. Data Collection and Processing Technique

The primary data were directly assembled by combining interview and questionnaire techniques based on the time availability of the respondents. More importantly, the respondents were Indonesian software users in office environments from various sectoral backgrounds. The respondents were selected through convenience sampling conducted in March 2024, based on the following criteria worked more than a year, had experience in stopping software usage during a particular case in the office, and was willing to participate in this research. Additionally, respective experiences in software discontinuity were outlined as mandatory requirements considering it reflected the qualification as respondents. The research did not specify the numbers of software discontinuity cases since it explored only one experience. In line with this finding, convenience sample selection was conducted by considering sector diversity. The selection process was also anonymized to ensure the protection of personal data, as shown in Table 2. The selection of diverse respondents aimed to extract various knowledge following the generated instruments. Additionally, the extracted knowledge was compared, contrasted, and synthesized to deliver a general pattern among respondents.

Building upon the description above, content analysis was conducted by capturing, codifying, and classifying statements from the entire respondents to discover a general pattern. This was aimed to organize and discover meaningful information from the data collected, including drawing realistic inferences [32] using the systematic process of coding (codification) [33]. Content analysis comprised four stages, namely decontextualization, recontextualization, categorization, and compilation [32]. It enabled the interpretation of the collected textual data in respect to reliable qualitative methods. Therefore, this research adopted elaborative quantitative method, as a manifestation of content analysis. Summative content analysis was also conducted, instead of the conventional or directed type due to the use of keywords as basis of codification, including the identification before and after extracting the literature review [34].

TABLE 2
RESPONDENTS' PERSONA

Identifier	Role	Job Expertise	Location	Gender	Age (years)
R.01	Audit Head	Financial service	Great Jakarta	Male	31 – 40
R.02	Vice Head of Public Relation Division	Education	Bandung	Male	31 – 40
R.03	Head of Information Security Management	Government	Great Jakarta	Female	41 – 50
R.04	Assistant Manager of IT Division	Education	Bandung	Male	31 – 40
R.05	Head of Performance Monitoring	Education	Bandung	Female	31 – 40
R.06	Sales Engineer	Manufacturing	Great Tangerang	Male	31 – 40
R.07	Finance Staff	Financial service	Bandung	Male	31 – 40
R.08	Lecturer	Education	Magelang	Female	31 – 40
R.09	Researcher	Research and Development	Great Jakarta	Male	41 – 50
R.10	Software Developer	Software Engineering	Great Jakarta	Male	21 – 30

IV. RESULTS

Several statements were obtained through interviews and questionnaires following the respondents' time availability. The explicit statements were labeled with specific codes, and then verified to ensure similar interpretation. The respondents' statements on the same question and topic were compared through codification. This comparison showed a convergence pattern due to similar statements and agreement. However, the comparison also disclosed divergence patterns for contrasting statements. Divergence patterns were perceived as logical consequences considering diverse experiences with different software in distinct case studies. The codification process reflected the recontextualization phase in line with the research conducted by Bengtsson [32]. The collected statements were checked to ensure whether each fulfilled the question context. In this context, Table 3 showcases samples of the statement about SOI.02 question.

TABLE 3
SAMPLE OF RESPONDENTS' STATEMENT CODIFICATION FOR QUESTION CODE SOI.02 (SEE TABLE 1)

Identifier	Statement	Researchers' Interpretation
R.01	The direct impact was the understanding process for new functions, so the leader created policies *[SOI.02-AGREE-001]* for intensive training for software usage.	The supervisor's solutive action was required positively through his commitment by facilitating the training.
R.04	Supervisors must be committed *[SOI.02-AGREE-002]* and consistent [*SOI.02-AGREE-003]* with the decisions they have made, including finding the root cause of the problem.	The supervisor's solutive action was required positively through his commitment and consistency.
R.05	Supervisors provided manuals and support *[SOI.02-AGREE-004]* for migration.	The supervisor's solutive action was required positively by facilitating manual and technical support for migration.
R.06	There is no direct effect *[SOI.02-DISAGREE-001]*, the process is already underway, and accommodated with training etc.	The supervisor's solutive action was weakly supportive by facilitating the training.
R.07	Superiors provide options for new application solutions that can be used *[SOI.02-AGREE-005]*	The supervisor's solutive action was required positively by facilitating training.
R.08	Superiors provide development directions that reflect the previous system *[SOI.02-AGREE-006]*	The supervisor's solutive action was required positively by facilitating training.

This research adopted qualitative and inductive methods, ensuring that the gathered textual data was codified following thematic analysis and classified using a general pattern. The answers given by the respondents, led to the discovery of a pattern that constructed convergence theme. Following the description, three themes were identified as classification from the entire annotated or codified findings. This included Software Usability, External Triggers, and Risk Managements. The categorization, including the labels were obtained from the synthesized interpretation of the generated results and reviewed theories and standards. For example, the Software Usability evolved from similarities among usefulness, ease of use from LUS and LEU issues, combined with ISO 25010:2023 as the baseline. The categorization aimed to simplify the mentioned issue into convergence understanding and interpretation. Furthermore, the categorization prompted the organization of various collected data into related themes, and efficient exploration of the proposed theories. The categorization made the results more concise and learnable to the readers. It manifested in the third stage of content analysis in line with the theory proposed by Bengtsson [32]. This research categorized several respondents' statements following the related issues as classified theme. Despite the consideration as samples, its existences reflected similar statements made by other respondents. The contrasted statements revealed the divergence pattern, while similar and redundant ones indicated the convergence pattern. Every theme revealed the respondents' statements, including those representing both similar and contrasting meanings. The existence represented respondents' preferences proportionally because similar or contrasting statements were shown. Therefore, the readers are able to determine whether the collected data were convergent or divergent following the available sample.

A. Issues on Software Usability

The theme comprised issues related to software usability and performance, referred to by several previous research, such as Fürst, Pecornik, and Hoyer [35] including Nyameino et al.[36], as antecedents for the adoption process. The antecedents also considered it as suspected factors that influenced software discontinuity. Usability in this context included software's effectiveness, efficiency, and ease, in line with the research conducted by Sagar and Saha [37] Kobyliński [38], Nyameino et al. [36], and also ISO 25010:2023 [23]. Effectiveness was also associated with Perceived Usefulness, while efficiency and ease represented Effort Expectation and Perceived Ease of Use, respectively.

Regarding the LUS, a divergent pattern was found among the respondents, with majority feeling that the discontinued software was not caused by decreasing or disappearance benefit. According to R.01 the decision to implement the software was not caused by benefits reduction *[LUS.02-DISAGREE-001]*. The situation occurrred because the development needs could not be accommodated by previous software, unless it was upgraded or replaced. R.05 made a similar statement, no *[SOI.02-DISAGREE-002]*, because the software change was mandatory and top-down. R.06, reported that there were no benefit reduction since it did not meet only the needs. The software was still running, with the features appearing inadequate. Furthermore, R.04 stated that the discontinued software has added value lacked by the replacement. For example, the respondent claimed more detailed asset management regarding location as an added value on features. Similar disagreements about LUS as contributing factors were expressed by R.08 and R.09. However, R.02 and R.07 stated that the previous software reduced work performance, because it was slower and required more report-generating time. The contradictory opinion depicted that the LUS did not automatically influence the decision to discontinue the software. It was suspected that the intervention of another factor caused the rejected allegations for most case studies on software performance.

Another suspected factor in usability issues is Effort Expectation, with R.06 outlining it as a causative factor that influenced the organization to discontinue software usage. Moreover, R.04 shared a personal experience ensuring software interoperability with database management systems. This played a crucial role in improving software quality. R.01 and R.02 also reported respective experience regarding learning about the new software, due to lack of familiarity. In line with this finding, familiarity was also perceived as a determinant that ensured people used new software. R.05 reported difficulties operating the new online spreadsheet software, which was interestingly more complex than the previous one.

The next factor in usability issue is LEU with two questions shown in Table 1. Meanwhile, through question LEU.01, this research found that more complex or complicated business processes running in software were the supporting factors in discontinuity. R.01 stated that the old software in an audit system could not generate infographics automatically but manually. R.04 also reported that the complexity of technological changes forced alteration *[LEU.01-DISAGREE-004]*. Fixing the running software requires recoding rather than a new feature. R.06 believed that software complexity was due to its many features and difficult usage. Additionally, lack of familiarity was considered an inhibiting factor ensuring software continuity followed R.02's opinion, who believed it reduced software easiness. Considering the experience of R.07, the office software was terminated after assessing the efficiency and security.

LEU.02 asked the respondents to give an example of respective experiences with terminating software usage because the work was either inefficient or unproductive. R.01 reported that the software was terminated because it could not support the growing daily work needs in the agile audit processes, including tracking and documenting similar application. Additionally, R.06 stated that monitoring and reporting processes in software were complicated alongside its needs. In agreement R.02, added that the processing time was more prolonged *[LEU.02-DISAGREE-*

002], and the results less optimal. Q.07 also focused on similar situation with manual processes on quarterly cash flow, which made the software appear inefficient.

B. Issues on External Trigger

The external triggers were from the management, other staff, and outer entity from the company. Most respondents reported that management also called boss is a decision maker in software discontinuity, occasionally, these are referred to as initiators. Respondent R.04 stated that the important key, from an organizational perspective is management, responsible for determining the needs of the organization, as well as deciding whether the software could be used based on certain considerations.

R.06 shared another experience associated with the SOI instruments. The staff were initiators when the software was stopped because majority encountered complex technical issues and spoke to management. Difficulties were also encountered with process measurement, although the final decision maker was management, the staff also played a role in software discontinuity. The staff also engaged in risk reduction because the technical issues required responsive ability since the software was operational in user experience, not managerial. Another respondent, R.10 shared the personal experience in an organization that determined software discontinuity and its substitute through an official internal letter. Furthermore, the letter reflected a legal decision in the management made, becoming a legal certainty for the employee to stop software use. This implied the management had legitimation and the power to determine software discontinuity. The respondent also expressed optimism that organizations can search for substitute software in specific cases such as for human resources business processes. The statement also implied that the control of previous software was lost since the management focused on it directly.

Finally, SOI and LBC contributed to software discontinuity empirically. These were actualized by external triggers through the consideration of the management and staff as decision maker in software governance and role played due to technical experiences, respectively.

C. Issues on Risk Management After Discontinuity

Risk management after software continuity was crucial since problems were always detected and required a reduction plan directly from the management. This was in line with the IT governance concept that outlined the importance of managing risk at the entire level. Interestingly, all respondents recognized that the top management paid great attention. As R.04 stated, the boss was committed for any consequence, including tracing the root cause. R.05 also reported that the boss provided technical and manual support for migration after the previous software had been terminated. In another case, R.06 revealed that the top management had performed the technical review before making decisions on software discontinuity.

Based on the description above, the respondents reported that commitment was a keyword related to finding solutions if any impacts occurred when the corporation discontinued the software. R.05 stated that the keyword should be actualized technically by the IT service from related department. This finding proved that LBC and SOI strongly contributed to convincing people about software discontinuity, due to the robustness when the bosses ensured commitment, including facilitating the IT service.

Another interesting finding on risk management was reported by R.01, after sharing a story about discontinuity, and facing new software with lack of ease (scoring 4 out of 10). Familiarity was an issue in user experience that should be reduced at the user level. However, after a month of using the new software, people became more familiar with it, indicating that the J-curve was a potential risk requiring training and agile response during implementation. In a separate interview, R.09 also shared personal worries about technical support that could disappear if the IT employees had been retired from the company or related unit.

After exploring the generated pattern among respondents' statements, this research manifested the fourth phase in content analysis [32], namely compilation. Based on the generated findings, captured patterns, and classified issues, including software usability, external triggers, and risk management after discontinuity, all 10 factors were defined as contributing factors to software discontinuity. In Software Usability, LUS, Decreased Effort Expectation (EFE), LEU, and Decreased Performance Expectancy (DPX) were categorized as contributing factors. Meanwhile, DPV, SOI, and LBC were considered as contributing factors to External Triggers. Risk Management comprised LFC, LHB, and SOI (appeared twice in two categories). Although these attributes were declared them as contributing factors, LEU and LUS had partial significance since most respondents preferred agreement without strong trustiness while others expressed disagreements about the contribution. HDM did not contribute to people's intention in software discontinuity, due to the disproval of its influence following based on experience. Fig. 3 shows the causative factors on software discontinuity following issue clustering and relationship strength.

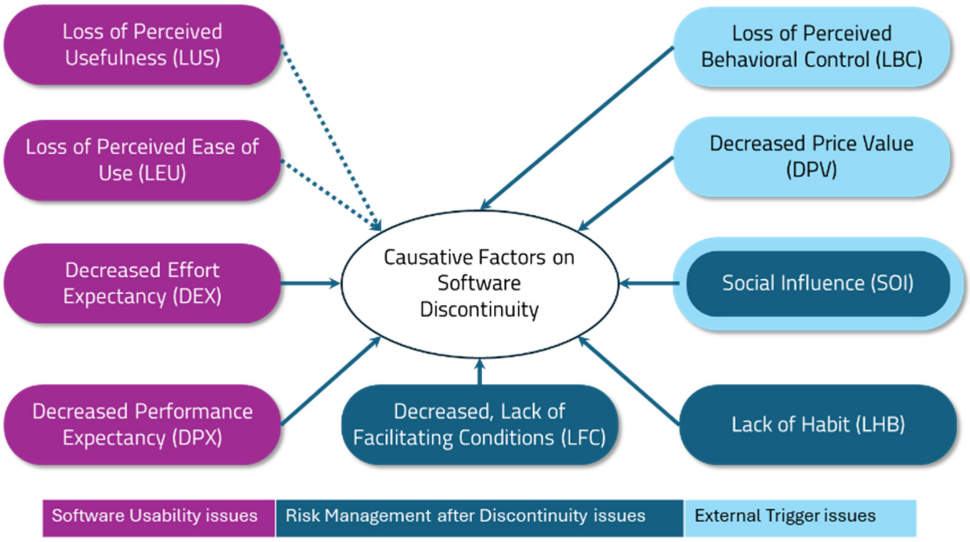

Fig. 3 Causative factors on software discontinuity.

All causative factors on software discontinuity were represented in the diagram with different color codifications. The purple factors referred to Software Usability issues, while the blue represented Risk Management after Discontinuity issues. Finally, cyan declared the factors as external trigger issues and the SOI was associated with two colors, implying all were covered. This figure used a straight line to expose the causative factors with strong significance following the convergence among respondents. However, the dotted line showed the causative factors had a weak significance because not all respondents agreed with the influence of those factors.

D. Validity and Reliability Compliance

Considering the research paradigm were constructivism and interpretivism produced knowledge from the combined respondents' experience and interpretation. Statistical checking was not conducted since it manifested the qualitative method stated in the beginning as shown in Fig. 2. However, to control the subjectivity, validity and reliability testing were conducted, following the characteristics of qualitative method-based research. The validity aspect referred to the consistency among generated results, discovering that the respondent's statements were convergent, despite being revealed and captured in different locations. Similarly, member checking and peer debriefing were performed to ensure the result interpretation was more objective and valid. Reliability testing was actualized through cross-checking independently for the derived codes in content analysis.

V. Discussion

The importance of factors identification was reviewed as the main issue in the Introduction as outlined in the second and third paragraphs. This research centered on a theoretical implication following several literature pieces, leading to the summary of the findings. First, lack usability including effectiveness, efficiency, and easiness led to software discontinuity due to reduced users' satisfaction. The findings were in line with the research conducted by Fürst, Pecornik, and Hoyer [35], Mishra et al. [11], and Nyameino et al. [36] It was reported that usability had positively relation with the intention to use or adopt. The current research reported a positive relations between lack of usability and the intention to terminate software usage.

Second, management played a significant role in decision-making since it was responsible for evaluating, directing, and managing all IT governance aspects, including software. The management also made a budget related to investing in software, aimed to ensure that business objectives could be achieved rationally through business process optimization, increased user satisfaction, and knowledge creation. These factors caused the management to ensure organizational accountability by reviewing software continuity. However, if the software lacked performance, satisfaction, and investment, the management must stop the usage, including setting up the mitigation plan.

SOI and Perceived Behavior Control contributed to the software discontinuity empirically. These factors were actualized by external trigger through management and staff considered as decision maker in software governance and role played due to technical experiences, respectively. The contribution of SOI was in line with the findings of Jo and Bang in 2023 [39] , as well as Fürst et al. [35].

Finally, the findings outlined that risk is a required agenda in software engineering. The systematic literature review by Pilliang and Munawar in 2022 [15], stated that software engineering performed risk management in respect to related assessment, categorization, prediction, and mitigation. This research was in line with Masso et al. [1] in 2022, which outlined several feasible frameworks and standards for adopting risk management in software engineering ISO 31000:2018, PMBOK, COBIT 5, and CMMI V 2.0. The frameworks and standards relied on risk management in the following life cycle identification, assessment, mitigation, and control. However, neither mentioned risk management in discontinuity nor software change case studies. This research contributed to enlarging knowledge about risk management for the entire software development life cycle. More importantly, it motivated software developer team to be proactive in using software defect algorithm as reported by Chennappana and Vidyaathulasiraman [40] and Nagaruju et al. [41]. The research also outlined the prioritization on risk management during software development as reported by Wirtz and Heisel [42]. Following the discussion, organizational support had a strong coherent with the research by Furneaux et al [43] in 2020 who stated that support availability was a requirement for system upgrade preferences. These reduction strategies had strong relevance to the research objectives stated in the fourth paragraph of Introduction, fulfilling the offered practical implications (fifth paragraph of Introduction).

Several limitations were identified first, the results were generated following the gathered respondents, therefore the extracted and elaborated knowledge was limited to specific respondents. Future research should select different respondents to obtain diverse results. Second, the respondents' profiles was limited to Indonesians, thereby affecting the results and interpretation. Finally, the discontinuity was limited to software used in a company or business context, not individually.

VI. CONCLUSIONS

In conclusion, this research presented a novel approach by proposing a software discontinuity model. It started by identifying factors that could lead to software discontinuation, drawing from three established theories, namely TAM, TPB, and UTAUT2. The model incorporated 10 main factors, such as LUS, LEU, DEX, DPX, SOI, LFC, DPV, LHB, HDM, and LBC. In addition, a rigorous qualitative method was adopted to validate the factors in the proposed software discontinuity model. The method prioritized the depth of primary data, ensuring a comprehensive understanding of the diverse respondents' perspectives. The analysis considered both the divergence and convergence of the answer patterns, further enhancing the reliability of the findings.

After proofing those 10 factors, this research contributed theoretically to expanding the software engineering field, especially related processes, by including factors the challenge. As reported in the Background, the software adoption as dominant topic in previous research had been elaborated with discontinuity explanation to reveal a general software usage journey. Moreover, the risk management concept was also manifested in software engineering context. In practical contribution, the 10 crucial factors were outlined for the software development team to ensure the product in the planning and design phases led to anticipated maintainability and sustainability.

Several recommendations were proposed, first, the results obtained using a qualitative method was expanded to strengthen the software discontinuity model's maturity and validity by including more diverse sectors and organizational sizes. Second, results were centered on the locals, therefore it should be retested in other countries to ensure the quality of global discontinuity software models by distinguishing local and global characteristics. Lastly, the results should be followed up with a comparative experimental model relating the anticipated factors discontinuity model, aimed to prove the significance.

Author Contributions: *Arfive Gandhi*: Conceptualization, Formal Analysis, Funding Acquisition, Investigation, Methodology, Project Administration, Software, Supervision, Writing – original draft, and Visualization. *Dana Sulistiyo Kusumo*: Data collection, Data curation, Investigation, Resources, Validation, and Writing - review & editing. *Indra Lukmana Sardi*: Data collection, Data curation, investigation, Resources, Validation, and Writing - review & editing.

Funding: This work was supported by Directorate of Research and Community Service, Telkom University, under the Basic Internal Grant "Exploring the Motivating Factors on Software Discontinuity" in fiscal year 2023-2024.

Conflicts of Interest: The authors declare no conflict of interest.

Data Availability: Dataset link will be available upon request.

Informed Consent: Informed Consent was obtained.

Institutional Review Board Statement: Not applicable.

Animal Subjects: There were no animal subjects.

ORCID:
Arfive Gandhi: https://orcid.org/0000-0001-6488-4341
Dana Sulistiyo Kusumo: https://orcid.org/0000-0002-7451-2653
Indra Lukmana Sardi: https://orcid.org/0009-0006-2617-8810

REFERENCES

[1] J. Masso, F. García, C. Pardo, F. J. Pino, and M. Piattini, "A Common Terminology for Software Risk Management," *ACM Transactions on Software Engineering and Methodology*, vol. 31, no. 4, pp. 1–47, Oct. 2022, doi: 10.1145/3498539.
[2] O. El Zarif, D. A. Da Costa, S. Hassan, and Y. Zou, "On the Relationship between User Churn and Software Issues," in *Proceedings of the 17th International Conference on Mining Software Repositories*, New York, NY, USA: ACM, Jun. 2020, pp. 339–349. doi: 10.1145/3379597.3387456.
[3] N. Haile and J. Altmann, "Evaluating investments in portability and interoperability between software service platforms," *Future Generation Computer Systems*, vol. 78, pp. 224–241, Jan. 2018, doi: 10.1016/j.future.2017.04.040.
[4] S. Ickin, K. Petersen, and J. Gonzalez-Huerta, "Why Do Users Install and Delete Apps? A Survey Study," 2017, pp. 186–191. doi: 10.1007/978-3-319-69191-6_13.
[5] O. Turel, "Untangling the complex role of guilt in rational decisions to discontinue the use of a hedonic Information System," *European Journal of Information Systems*, vol. 25, no. 5, pp. 432–447, Sep. 2016, doi: 10.1057/s41303-016-0002-5.
[6] S. Liang, Z. Wei, and L. Zang, "An exploratory study of factors influencing user app abandonment on smartphones," *Library Hi Tech*, Jan. 2024, doi: 10.1108/LHT-07-2022-0349.
[7] A. Kolomiiets, O. Mezentseva, and K. Kolesnikova C A, "Customer Churn Prediction in the Software by Subscription models IT business using machine learning methods," 2018.
[8] P. Berger and M. Kompan, "User Modeling for Churn Prediction in E-Commerce," *IEEE Intell Syst*, vol. 34, no. 2, pp. 44–52, Mar. 2019, doi: 10.1109/MIS.2019.2895788.
[9] S. Bernardo *et al.*, "Software Quality Assurance as a Service: Encompassing the quality assessment of software and services," *Future Generation Computer Systems*, vol. 156, pp. 254–268, Jul. 2024, doi: 10.1016/j.future.2024.03.024.
[10] T. Olsson, S. Sentilles, and E. Papatheocharous, "A systematic literature review of empirical research on quality requirements," *Requir Eng*, vol. 27, no. 2, pp. 249–271, Jun. 2022, doi: 10.1007/s00766-022-00373-9.
[11] A. Mishra and Z. Otaiwi, "DevOps and software quality: A systematic mapping," *Comput Sci Rev*, vol. 38, p. 100308, Nov. 2020, doi: 10.1016/j.cosrev.2020.100308.
[12] M. A. Almaiah, A. Al-Khasawneh, and A. Althunibat, "Exploring the critical challenges and factors influencing the E-learning system usage during COVID-19 pandemic," *Educ Inf Technol (Dordr)*, vol. 25, no. 6, pp. 5261–5280, Nov. 2020, doi: 10.1007/s10639-020-10219-y.
[13] S. Na, S. Heo, S. Han, Y. Shin, and Y. Roh, "Acceptance Model of Artificial Intelligence (AI)-Based Technologies in Construction Firms: Applying the Technology Acceptance Model (TAM) in Combination with the Technology–Organisation–Environment (TOE) Framework," *Buildings*, vol. 12, no. 2, p. 90, Jan. 2022, doi: 10.3390/buildings12020090.
[14] K. S. Qader *et al.*, "The impact of Technological acceptance model (TAM) outcome on implementing accounting software," *International Journal of Engineering, Business and Management*, vol. 6, no. 6, pp. 14–24, 2022, doi: 10.22161/ijebm.6.6.3.
[15] M. Pilliang and M. Munawar, "Risk Management in Software Development Projects: A Systematic Literature Review," *Khazanah Informatika : Jurnal Ilmu Komputer dan Informatika*, vol. 8, no. 2, Sep. 2022, doi: 10.23917/khif.v8i2.17488.
[16] B. Furneaux, S. Mannina, and L. Rieser, "Responding to Information System Obsolescence: Should We Upgrade or Replace?," *Journal of Computer Information Systems*, vol. 62, no. 2, pp. 372–383, Mar. 2022, doi: 10.1080/08874417.2020.1830006.
[17] V. Venkatesh, S. M. Walton, J. Y. L. Thong, and X. Xu, "CONSUMER ACCEPTANCE AND USE OF INFORMATION TECHNOLOGY: EXTENDING THE UNIFIED THEORY OF ACCEPTANCE AND USE OF TECHNOLOGY," 2012. [Online]. Available: http://ssrn.com/abstract=2002388
[18] U.-K. Lee and H. Kim, "UTAUT in Metaverse: An 'Ifland' Case," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 17, no. 2, pp. 613–635, Apr. 2022, doi: 10.3390/jtaer17020032.
[19] A. Raman and Y. Don, "Preservice Teachers' Acceptance of Learning Management Software: An Application of the UTAUT2 Model," *International Education Studies*, vol. 6, no. 7, Jun. 2013, doi: 10.5539/ies.v6n7p157.
[20] I. Ajzen, "The theory of planned behavior: Frequently asked questions," *Hum Behav Emerg Technol*, vol. 2, no. 4, pp. 314–324, Oct. 2020, doi: 10.1002/hbe2.195.
[21] I. Ajzen and P. Schmidt, "Changing Behavior Using the Theory of Planned Behavior," in *The Handbook of Behavior Change*, Cambridge University Press, 2020, pp. 17–31. doi: 10.1017/9781108677318.002.
[22] C. Hiranrat, A. Harncharnchai, and C. Duangjan, "Theory of Planned Behavior and the Influence of Communication Self-Efficacy on Intention to Pursue a Software Development Career," *Journal of Information Systems Education*, vol. 32, no. 1, pp. 40–52, 2021.
[23] "Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE)-Product quality model," 2001.
[24] "Software engineering-Product quality-Part 1: Quality model," 2001.
[25] M. Saunders, P. Lewis, and A. Thornhill, *Research Methods for Business Students*. Research Methods for Business Students, 2023.
[26] M. Burns, J. Bally, M. Burles, L. Holtslander, and S. Peacock, "Constructivist Grounded Theory or Interpretive Phenomenology? Methodological Choices Within Specific Study Contexts," *Int J Qual Methods*, vol. 21, Apr. 2022, doi: 10.1177/16094069221077758.

[27] W. D. Roth, E. G. van Stee, and A. Regla-Vargas, "Conceptualizations of Race: Essentialism and Constructivism," *Annu Rev Sociol*, vol. 49, no. 1, pp. 39–58, Jul. 2023, doi: 10.1146/annurev-soc-031021-034017.
[28] H. H. Alharahsheh and A. Pius, "A Review of key paradigms: positivism VS interpretivism," *Global Academic Journal of Humanities and Social Sciences*, vol. 2, no. 3, pp. 39–43, 2020, doi: 10.36348/gajhss.2020.v02i03.001.
[29] N. Pervin and M. Mokhtar, "The Interpretivist Research Paradigm: A Subjective Notion of a Social Context," *International Journal of Academic Research in Progressive Education and Development*, vol. 11, no. 2, Apr. 2022, doi: 10.6007/IJARPED/v11-i2/12938.
[30] J. Creswell, "Qualitative Inquiry & Research Design: Choosing Among Five Approaches," *SAGE Publications*, vol. 11, Mar. 2013.
[31] J. W. Creswell and J. D. Creswell, *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2017.
[32] M. Bengtsson, "How to plan and perform a qualitative study using content analysis," *NursingPlus Open*, vol. 2, pp. 8–14, 2016, doi: 10.1016/j.npls.2016.01.001.
[33] A. Assarroudi, F. Heshmati Nabavi, M. R. Armat, A. Ebadi, and M. Vaismoradi, "Directed qualitative content analysis: the description and elaboration of its underpinning methods and data analysis process," *Journal of Research in Nursing*, vol. 23, no. 1, pp. 42–55, Feb. 2018, doi: 10.1177/1744987117741667.
[34] H.-F. Hsieh and S. E. Shannon, "Three Approaches to Qualitative Content Analysis," *Qual Health Res*, vol. 15, no. 9, pp. 1277–1288, Nov. 2005, doi: 10.1177/1049732305276687.
[35] A. Fürst, N. Pecornik, and W. D. Hoyer, "How product complexity affects consumer adoption of new products: The role of feature heterogeneity and interrelatedness," *J Acad Mark Sci*, vol. 52, no. 2, pp. 329–348, Mar. 2024, doi: 10.1007/s11747-023-00933-7.
[36] J. N. Nyameino, B.-R. Ebbesvik, F. Rabbi, M. C. Were, and Y. Lamo, "Model-Driven Automatic Question Generation for a Gamified Clinical Guideline Training System," 2020, pp. 227–245. doi: 10.1007/978-3-030-40223-5_11.
[37] K. Sagar and A. Saha, "A systematic review of software usability studies," *International Journal of Information Technology*, Dec. 2017, doi: 10.1007/s41870-017-0048-1.
[38] A. Kobyliński, "The Relationships between Software Development Processes and Software Product Quality," 2013, pp. 161–169. doi: 10.1007/978-3-642-40823-6_13.
[39] H. Jo and Y. Bang, "Analyzing ChatGPT adoption drivers with the TOEK framework," *Sci Rep*, vol. 13, no. 1, p. 22606, Dec. 2023, doi: 10.1038/s41598-023-49710-0.
[40] R. Chennappan and Vidyaathulasiraman, "An automated software failure prediction technique using hybrid machine learning algorithms," *Journal of Engineering Research*, vol. 11, no. 1, p. 100002, Mar. 2023, doi: 10.1016/j.jer.2023.100002.
[41] V. Nagaraju, V. Shekar, J. Steakelum, M. Luperon, Y. Shi, and L. Fiondella, "Practical software reliability engineering with the Software Failure and Reliability Assessment Tool (SFRAT)," *SoftwareX*, vol. 10, p. 100357, Jul. 2019, doi: 10.1016/j.softx.2019.100357.
[42] R. Wirtz and M. Heisel, "Model-Based Risk Analysis and Evaluation Using CORAS and CVSS," 2020, pp. 108–134. doi: 10.1007/978-3-030-40223-5_6.
[43] B. Furneaux, S. Mannina, and L. Rieser, "Responding to Information System Obsolescence: Should We Upgrade or Replace?," *Journal of Computer Information Systems*, vol. 62, no. 2, pp. 372–383, Mar. 2022, doi: 10.1080/08874417.2020.1830006.

Publisher's Note: Publisher stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.