Article

Securing Software Development Through People Maturity: A Fuzzy-AHP Decision Making Framework

Rafiq Ahmad Khan *, Hussein A. Al Hashimi, Hathal S Alwageed, Ismali M Keshta, Alaa Omran Almagrabi, Saraa Ayouni

Posted Date: 4 September 2025

doi: 10.20944/preprints202407.0669.v2

Keywords: secure software development; human success factors and vulnerabilities; practices; decision-making framework; empirical study; fuzzy-AHP

Article

Securing Software Development Through People Maturity: A Fuzzy-AHP Decision-Making Framework

Rafiq Ahmad Khan [1],*, Hussein A. Al Hashimi [2], Hathal S. Alwageed [3], Ismail Keshta [4], Alaa Omran Almagrabi [5] and Sarra Ayouni [6]

[1] Software Engineering Research Group, Department of Computer Science and IT, University of Malakand, Pakistan
[2] College of Computer and Information Sciences, King Saud University, Saudi Arabia
[3] College of Computer and Information Sciences, Jouf University, Sakaka, Saudi Arabia
[4] Department of Applied Science, College of Computer Science, Almaarefa University, Riyadh, Saudi Arabia
[5] Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
[6] Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
* Correspondence: Rafiq Ahmad Khan, Email: rafiqahmadk@gmail.com

Abstract

The increasing complexity of software development processes has heightened the need for robust security measures. While technical safeguards are essential, the role of human factors in securing software development remains underexplored. This paper presents a novel approach that integrates people's maturity with a Fuzzy Analytic Hierarchy Process (Fuzzy-AHP) decision-making framework to enhance the security in software development. The framework provides a systematic method for evaluating and prioritizing human factors that influence an organization's security posture, such as team-expertized communication and adherence to security protocols. Using the decision-making model allows the project managers and stakeholders to determine the appropriate areas for improvement and develop the right strategies and actions to nurture a secure and mature development culture. The paper identifies 24 human success factors (HSFs) and human security vulnerabilities (HSVs) and 38 practices for addressing these HSFs and HSVs through systematic literature review (SLR) and empirical survey. Furthermore, we discuss the local and global ranks of each HSF and HSV practice and categorize the identified practices into nine categories to determine the ranks and weight of each category. Based on collected data, Fuzzy-AHP prioritized these practices; the category "C4: Skill Development and Stakeholder Engagement" is ranked highest at rank-1 and possesses the most significant weight of 0.12435. Similarly, the highest global weight is 0.051506, and the global ranked (rank-1) HSF and HSV practice is "P15: Hands-On Practice and Stakeholder Communication". The proposed approach complements existing technical methods by addressing the human element of security, making it adaptable to diverse organizational environments. Through this integration of people maturity and Fuzzy-AHP, the paper contributes a new dimension to securing software development, emphasizing the critical role of human factors in achieving comprehensive security.

Keywords: secure software development; human success factors and vulnerabilities; practices; decision-making framework; empirical study; fuzzy-AHP

I. Introduction

The continuing progress of technology systems has raised the importance of software in various aspects of daily life, including socioeconomic activities. As the reliance on such systems increases, the vulnerability of the software that runs these organizations also increases, making secure software development (SSD) an essential goal for organizations worldwide. These are significant risks associated with security vulnerabilities that are the root causes of financial woes, reputational losses, and unwanted penetration of sensitive data [1]. Therefore, It is evident that software systems' protection is paramount in developing and deploying software systems locally and globally [2]. Recent trends in approaching software security issues involve consideration of the human element in the improved security of the application [3]. Organizational maturity of the development team, best known as people maturity, has been cited as an essential factor that influences the effectiveness of SSD practices. Person maturity includes factors like the behavioral aspect, communication, interaction between the team members, expertise, and experience they possess that either facilitates the security of the software projects or not [4].

However, making people mature and developing such a culture, even in a software development environment, is still tricky [5]. Previous methods of identifying and enhancing team thriving tend to avoid an entirely systematic approach considering the framework concept when analyzing human factors. Within this context, a research gap is identified that focuses on developing a systematic framework that may be used to systematically assess and facilitate improvement in people's maturity concerning the SSD context. This paper's first and foremost purpose is to bring us to the development of the framework as a guide to help organizations follow a structured approach in assessing the maturity levels of the development team and pinpointing the significant issues. By so doing, it wants to establish a culture of having people continuously learn while adopting changes to ensure that security measures are still effective, given the fast changes and growing technology.

A. Human Success Factors (HSFs) and Human Security Vulnerabilities (HSFs) in Secure Software Development

HSFs and HSVs are two sides of the same coin in secure software development (SSD) [6]. This paper aims to establish that when these factors are effectively addressed, they play a central role in enhancing secure and dependable software systems. People success factors denote characteristics and behaviors that contribute positively to the capabilities of the personnel and subgroups for the development of secure software. These are technical capabilities, communication, problem-solving, security sensitivity, and learning orientation [7].

The HSFs and the HSVs have central functions in SSD projects [8]. The most vital of HSFs are team experience, knowledge, communication, and learning [9]. Expertise is defined as the technical know-how of the personnel in a team assigned to trace and prevent threats to security. This forms part of communication within the team and with stakeholders to ensure the security requirements are understood and implemented [10]. It helps team members to be up-to-date with security trends, thus making it easier to counter them quickly.

Human security risks arise from the weaknesses and flaws that people and groups may demonstrate, creating threats to software security [6,8]. Such risks include the user's security knowledge, insufficient training, poor communication, and ignoring security policies [11]. Most insecurity issues likely originate from carelessness, a negligent act, or even deliberate provocation. Hence, it is vital to comprehend and minimize them to strengthen the protective measures of software development projects.

Despite the recognized importance of human factors, there is still a dearth of a framework that systematically examines all of these facets and offers systematic incorporation of these characteristics into the security paradigms of software development [8]. The nature of human interactions and the variability inherent in teamwork complicates defining and identifying all relevant success parameters and potential weaknesses in one complete strategic profile [12].

On the other hand, security threats result from people's weaknesses like insufficient training, communication breakdown, and strained cooperation [11]. Lack of training exposes the programmer to a lack of information on new security threats and security measures that can lead to the ease of vulnerabilities in the software being developed [13]. Lack of communication may lead to non-coordination; thus, misinterpretation of the security necessities and inefficient implementation can appear, and inadequate collaboration may cause disintegrated work in securing the software [14]. Also, coding errors and incorrect system configurations contribute to security threats [15].

B. Aims of the Study

Mitigating software development risks can only be possible by embracing a systems approach in the development of people through acquiring appropriate skills, coordination of efforts in implementing security measures, and effective continuous learning processes to boost the security outcome of software development projects [16].

This paper intends to fill this gap by proposing a comprehensive SSD model that integrates human success factors and security threats. In this way, using methods that consider the complexity of human factors, such as the fuzzy analytical hierarchy process (Fuzzy-AHP), we will carry out a detailed analysis of the human components of software security to identify ways for improvement. Fuzzy-AHP integrates the multilevel decision-making ability of AHP into the concept of fuzzy, which offers versatility in presenting human factors frequently in vague conditions.

C. Significance of the Study

This study addresses a critical gap in the software development lifecycle by focusing on the role of people's maturity in enhancing security. As cybersecurity threats continuously evolve, software development organizations must strengthen their development processes to safeguard against potential risks.

This research contributes by introducing novel decision-making that integrates Fuzzy-AHP to assess and prioritize human factors-skills, experience, and teamwork maturity-within software development. By incorporating fuzzy logic, the model accounts for uncertainty and subjectively evaluates people-related aspects, making it adaptable to various organizational settings.

The significance of this study lies in its potential to guide software development organizations toward more secure software development by systematically evaluating and improving people's maturity levels. It offers a structured approach for decision-makers to identify weak areas in their teams and take proactive measures, ultimately leading to more robust, secure, and resilient software systems. This focus on the human element complements existing technical security measures, paving the way for more comprehensive security strategies in software engineering.

D. Organization of the Paper

The paper unfolds as follows: Section 2 delves into the motivation and background of the study. A detailed examination of the study's methodology is presented in Section 3. Section 4 conducts an in-depth analysis and evaluation of the acquired results—and section 5 offers securing software development through a people maturity decision-making framework for SSD organizations. Section 6 addresses the study's implications, Section 7 summarizes the study and its limitations, and Section 9 concludes by highlighting avenues for further research.

II. Background and Related Work

This section describes the motivations behind this study, defines software security, and discusses HSFs and HSVs in SSD-related work.

A. Motivations Behind the Study

The motivation behind this work is multi-layered, resting on the growing intricacy of software systems and the rising risks that modern enterprises contend with. First, hacking attacks and data leakage are critical issues in software applications and result from poor software development practices [17]. While the traditional security practices and mechanisms are still pertinent, they are insufficient if used individually [18,19]. It has emerged that project aspects, such as the human facets of development teams and especially their skills, collaborations, and maturity levels, have a significant bearing on the security levels of software systems [20].

Secondly, the current approaches and methods to increase software security mainly use technical disciplines while not considering people's maturity enough [21]. They concluded that there are no well-defined paradigms of human factors' consideration in the decision-making related to SSD [22]. This gap in the current state of knowledge influenced the investigation of a holistic framework to examine and advance the maturity of development teams and, consequently, improve overall security [23].

Thirdly, the dynamic condition of technology and cyber threats requires decision-making criteria that can be effective in uncertain and changing situations [24,25]. The Fuzzy Analytical Hierarchy Process (Fuzzy-AHP) provides a reliable solution by clearly categorizing AHP and the elasticity of fuzzy logic [20]. Thus, this hybrid approach enables them to consider other factors of workers' maturity besides the objective since people and technology aspects require such flexibility in their assessment.

Besides, this research appears to be an attempt triggered by the real-world need to encourage organizations to embrace efficient security strategies. In turn, with the assistance of the created framework that enables considering and improving the maturity of development teams systematically, this work will contribute to creating more secure and reliable software systems. The long-term goal is to create a more sustainable approach to software security and encourage the spirit of aspiring to be bigger and better across development teams, decreasing threats to an application's security and improving the software systems' solidness.

In sum, the engineering motivations for undertaking this work are driven by the increasing awareness of the human aspects of software security, the lack of fit of existing paradigms to realize it, and the empirical reality of an organization's struggle to achieve secure software development. Following the proposed Fuzzy-AHP decision-making framework, this research will contribute a valuable framework to improve individuals' maturity and, in turn, increase the security of software development projects.

B. Software Security

Software security aims to build programs that can continue to run even when faced with hostile attacks [26]. The best way to reduce software vulnerabilities and problems is to incorporate security and non-functional requirements throughout the Software Development Life Cycle (SDLC) [3]. "Secure software" is an application that follows secure development standards and practices during its design or engineering process [20]. That way, even if the program is attacked, its operations and functions will keep running smoothly. One of the main goals of secure software is to prevent unauthorized individuals from viewing or changing data [27]. Preventing development and maintenance costs requires strict adherence to security standards throughout the design and implementation processes [28].

C. People's Maturity Toward Secure Software Development

Analyzing the concept of people maturity in the context of secure software development (SSD), it is possible to identify the competency level and the degree of preparedness and activity of individuals and teams regarding security measures within SDLC. Thus, failure to realize high people maturity hinders the development of dependable software systems for responding to emerging threats and risks. Following are the dimensions of people's maturity toward SSD [15,29–35]:

1). Dimensions of People Maturity

i. Technical Proficiency and Skill Development: The identification and confirmation that the members of the team have the required competencies is perhaps the most essential element of people's maturity. Enlisting a professional training and education program on the best practices, instruments, and techniques of contemporary securities is also crucial.

ii. Experience: The professional members of the team need to be able to use their expertise and analyze the probable problem related to security and then implement the principles learned most appropriately.

iii. Process Adherence and Standardization: Excellent teams operate with standard methodologies and guidelines that contain security since the beginning of the development stage, for instance, SSDLC models.

iv. Compliance: Thus, following the best practices of the industry and the legally compliant acts guarantees that the security will be the best and most updated.

v. Communication and Collaboration and Interdisciplinary Coordination: Communication is essential because it helps the developers, security specialists, and other related stakeholders to ensure that the security requirements are understood and reflected in the development process.

vi. Feedback Mechanisms: The development of sound feedback mechanisms enables the development team to quickly note or report on security issues as the project progresses.

vii. Proactive Security Culture and Security Awareness: It is compelling to make security a part of a team culture where everyone understands the organization’s threats and actively seeks to prevent them.

viii. Responsibility and Accountability: This gives them ownership of security tasks; in the process, individuals feel responsible for the overall security tasks.

ix. Problem-Solving and Analytical Skills: Most well-established teams employ very formal and rational strategies like threat modeling and risk analysis to analyze the weaknesses in security systems.

x. Decision Frameworks: By applying decision-support tools like Fuzzy-AHP, the teams can calculate security measures considering various factors and the fuzziness of the decision-making environment.

2). Benefits of People's Maturity Toward Secure Software Development

The following benefits of people's maturity towards SSD are identified through literature [15,29–35]:

i. Enhanced Security Posture: Originally, higher people's maturity meant a better ability to understand and eliminate security threats, thus improving security maturity.

ii. Reduced Risk: Consequently, the secure maturity level of the formative team means the ability to independently assess the threats and threats handling, which leads to a decrease in the number of and the impacts due to security breaches.

iii. Improved Compliance: Compliance with security standards and meeting regulatory standards increases, meaning the software is legal and complies with industry standards.

iv. Increased Efficiency: That way, the practices are standardized, and communication is efficient in establishing means of providing security without overloading developers for a long time.

v. Continuous Improvement: By practicing open acknowledgment that is not confined to learning through experience but goes further to embrace proactivity, the teams remain informed on the current security trends and hence make constant enhancements to security.

3). Achieving People's Maturity Toward Secure Software Development

To achieve people maturity in secure software development, organizations should [15,29–35]:

i. Invest in Training: Develop and implement continuing formal training sessions and seminars about new means and protection methods.

ii. Implement Standard Processes: First, implement and apply the set of requirements for security across the entire project, not just the separate ones.

iii. Foster a Security Culture: Encourage the organization's personnel, teams, and departments to embrace security matters.

iv. Facilitate Communication: Promoting the free flow of information and patronizing the establishment of rapport between all the participating stakeholders, especially in designing the software.

v. Utilize Decision-Making Tools: Utilize well-formalized decision-making approaches like Fuzzy-AHP in security-related decision-making.

Thus, people maturity is a significant aspect of securing software processes, covering a development team's competencies, practices, information sharing, and decision-making elements. Overall, by methodologically increasing these dimensions, it is possible to create high performance for developing and maintaining secure software systems and, thus, to guard against the constant growth of potential threats in information security.

D. Related Work

People's maturity related to (SSD) has emerged as the focus of attention in recent years. This section presents a literature review and the current frameworks for dealing with human factors in software security. Based on the literature review of previous studies, this section seeks to establish the differences and similarities of the current study regarding their contributions, shortcomings, and the existing research gaps. This research will focus on the following subsections to cover the related work:

1). Human Factors in Software Security

Human factors, at times, determine the security of software systems. Many papers have stressed the importance of awareness, training, and skills developers must possess to avoid security threats. For instance, a recent study by Acar et al. (2017) assessed the effects of security training on developers' performances in developing secure code, and the authors noted that specific training boosts security solutions. It was also implemented by Xie et al., 2011 the study of the developer's expertise to identify and mitigate security flaws, proving that developed developers can manage security issues.

2). Secure Software Development Lifecycle

The accommodation of security practices into the Software Development Lifecycle (SDLC) has been popularized for quite some time through several models and frameworks. Another model worth mentioning is the Secure Software Development Lifecycle (SSDLC), which was developed by Microsoft and can be described as the approach that focuses on the idea of security as the integral characteristic of the development process and should be considered and implemented at each stage, starting from the stage of requirements definition and ending with maintenance. This has been found helpful in improving the security of software projects, thus qualifying for improvement. Still, compared with SSDLC, which offers a clear approach, there is not enough emphasis on the human aspect if the security plan is effective.

3). Maturity Models

Other frameworks and management models, like CMMI and SAMM, are designed to present organizations with checklists to evaluate and advance their processes. These models include aspects that are related to security. Still, many of these models do not inherently address the level of maturity of team members in terms of security. For example, CMMI addresses improving and advancing an organization's capabilities. At the same time, SAMM offers a framework for implementing security, especially in software development, but does not delve into specific skills and team aspects.

4). Decision-Making Frameworks

AHP and FAHP have been used to solve many decision-making problems in software project management and risk analysis. Fuzzy AHP has mainly been applied to manage the vagueness and subjectivity of human perception. A literature review done by Wang and Elhag (2006) and Lee et al. (2008) illustrated that Fuzzy AHP is effective in ranking the criteria that cannot be clearly defined with regularity, and the decision-making is made with multitudinous factors under conditions of uncertainty. However, the exact application to measure people's maturity, specifically in secure software development, is still a question mark.

5). Roles of Human Factors and Security in Integration

Recent studies focus on the interaction of the human element with typical security frameworks. For instance, Assal and Chiasson (2019) developed a framework to examine the effects of various factors attributable to developers in security practices, suggesting that adequate training and awareness should be conducted frequently. Moreover, works such as Green et al. (2016) have also pointed out the need to improve communication and cooperation among the development teams to improve the available security.

6). Gaps and Opportunities

While the literature on human factors for software development demonstrates an appreciation of people in secure software development, there is a lack of consistent approaches that systematically evaluate and improve people's maturity. Few previous models and frameworks are either technical-oriented or too general and lack a clear structure for measuring the maturity of individuals and teams. Furthermore, there is a lack of research on applying decision-making frameworks such as Fuzzy AHP specifically to health organizations in this context.

7). Conclusion

Summarizing, it is essential to move a step forward, stressing that even though considerable progress is being made in the acknowledgment of human factors regarding software security, it is crucial to develop a systematic approach to carry out the measurement and improvement of people's maturity within the frame of secure software development. This research will fill this gap using the Fuzzy AHP decision-making approach, which incorporates human factors into the security appraisal. It will offer a single solution for organizations to strengthen security by addressing people's maturity.

III. Research Methodology

To know the impact of identified HSFs and HSVs for SSD projects, a three-phase methodology (depicted in Figure 1) was employed. First, a Systematic Literature Review (SLR) identified HSFs and HSVs and recommended practices for SSD projects. Subsequently, an empirical study involving SSD experts was carried out in the second step to assess whether the HSFs, HSVs, and practices identified in the SLR influence the security processes of SSD projects.

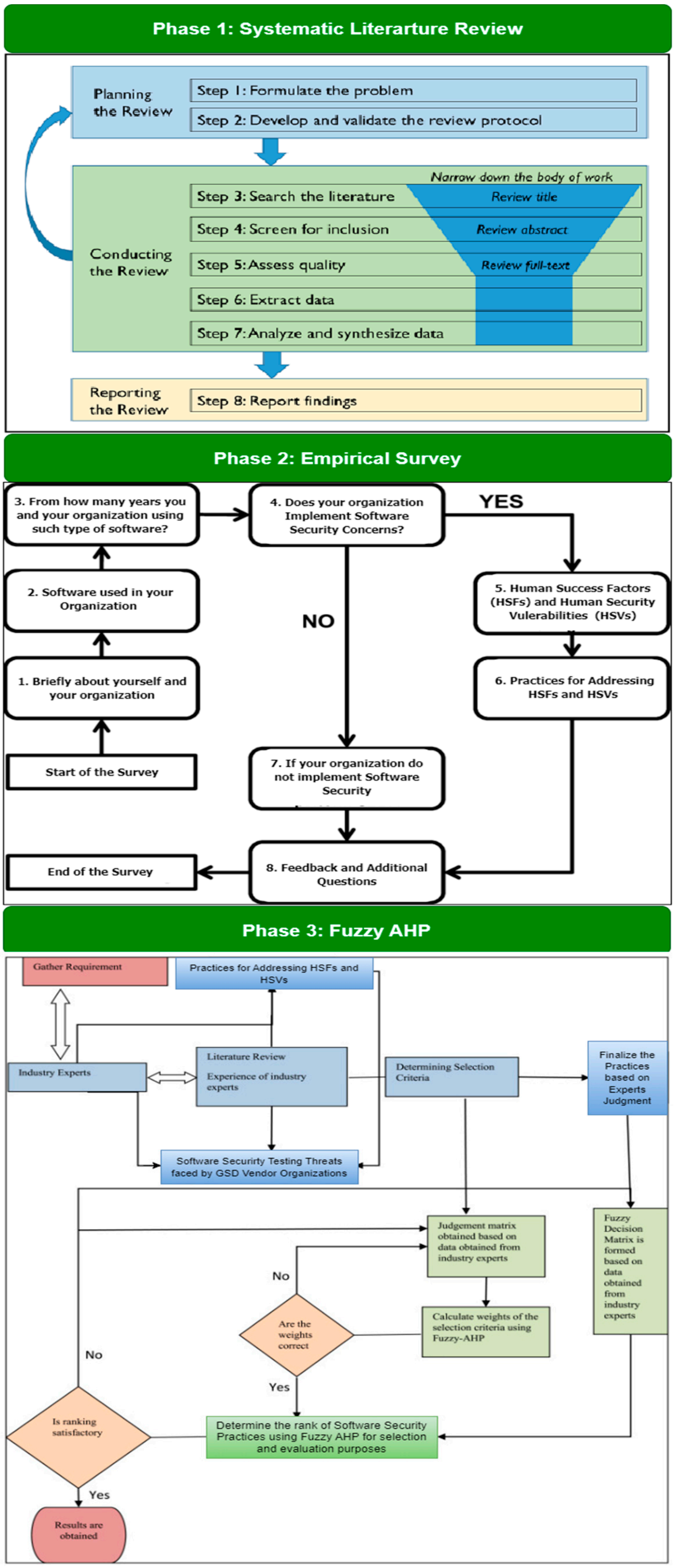

Phase 1: Systematic Literarture Review
Planning the Review
Step 1: Formulate the problem
Step 2: Develop and validate the review protocol
Narrow down the body of work
Conducting the Review
Step 3: Search the literature
Review title
Step 4: Screen for inclusion
Review abstract
Step 5: Assess quality
Review full-text
Step 6: Extract data
Step 7: Analyze and synthesize data
Reporting the Review
Step 8: Report findings
Phase 2: Empirical Survey
3. From how many years you and your organization using such type of software?
4. Does your organization Implement Software Security Concerns?
YES
NO
2. Software used in your Organization
5. Human Success Factors (HSFs) and Human Security Vulerabilities (HSVs)
1. Briefly about yourself and your organization
6. Practices for Addressing HSFs and HSVs
Start of the Survey
7. If your organization do not implement Software Security
End of the Survey
8. Feedback and Additional Questions
Phase 3: Fuzzy AHP
Gather Requirement
Practices for Addressing HSFs and HSVs
Industry Experts
Literature Review
Experience of industry experts
Determining Selection Criteria
Finalize the Practices based on Experts Judgment
Software Securirty Testing Threats faced by GSD Vendor Organizations
Judgement matrix obtained based on data obtained from industry experts
Fuzzy Decision Matrix is formed based on data obtained from industry experts
No
Are the weights correct
Calculate weights of the selection criteria using Fuzzy-AHP
No
Yes
Is ranking satisfactory
Determine the rank of Software Security Practices using Fuzzy AHP for selection and evaluation purposes
Yes
Results are obtained

Figure 1. Research Methodology.

Finally, the identified HSFs and HSVs practices were ranked using fuzzy-AHP, considering the significance of these practices in the context of the SSD domain.

The outlined phases are further detailed and discussed in the following subsections:

A. Phase 1: Systematic Literature Review (SLR)

A systematic literature review (SLR) was conducted initially, and it was used to investigate the realm of HSFs and HSVs and the practices that impact SSD projects. SLRs involve a meticulous and unbiased examination of primary studies, iteratively defining, interpreting, and discussing evidence relevant to research inquiries [36–39]. The guidelines outlined by Kitchenham and Charters [37] were meticulously followed to complete this SLR. SLRs are known to yield more comprehensive and reliable results, as asserted by Kitchenham [40].

SLR's objective is to provide a systematic and comprehensive research study of the literature to evaluate people's maturity contribution to SSD and ensure that potential research gaps are uncovered and addressed effectively while developing the fuzzy-AHP decision-making framework. According to Kitchenham and Charters [37], the following steps are involved in SLR conduction:

1). Step 1: Identification of Research Questions

The first step in designing an SLR involves developing specific and directed questions that the study hopes to answer. In this step, the research questions are clearly stated. It identifies human factors that play crucial roles in the security of software development projects—how people's maturity is measured in the context of SSD.

The following research questions help us get closer to our goals:

RQ1: What are the human factors (success factors and security vulnerabilities), as reported in the literature and real-world industry, that influence secure software development projects?
RQ2: Are there any differences between the success factors and security vulnerabilities identified in the literature and real-world industry?
RQ3: What are the best practices, as reported in the literature and real-world industry, that address the human factors (security vulnerabilities) in secure software development projects?
RQ4: How can a decision-making framework be developed to identify the weights of human success factors and security vulnerability practices and evaluate the effectiveness of secure software development projects?

2). STEP 2: Reviewing or Drafting Review Protocol

Similarly, we create a Review Protocol. A review protocol is, therefore, a plan prepared before carrying out the SLR to determine the most appropriate approach.

- Search Strategy: Specified keywords, databases, and search engines will be used.
- Inclusion and Exclusion Criteria: Selection of literature.
- Data Extraction: The type of data that can be mined from each study.
- Quality Assessment: The quality of the studies included in the review is also considered.

3). STEP 3: Search String Designing and Implementing

The next step we followed was designing and implementing an effective literature search. Here, the following defined search strategy is applied, and a broad search is performed on several databases and search engines. Some possible keywords could be phrases like 'people maturity software security,' 'secure software development,' 'human factor,' 'Fuzzy AHP', and 'decision-making frameworks.'

4). STEP 4: Screening and Selecting Studies

Screening and selecting studies is one of the most critical steps in a systematic literature review and can take up to two-thirds of the time required to conduct a review. The first step we conducted in the screening process was identifying those studies that might be relevant according to the article's title and abstract. The shortlisted studies are then reviewed in full text to ensure they fit the inclusion/exclusion criteria. Research that examined human aspects that play a role in secure software development. Studies discussing people's maturity from the perspective of software security. Research articles discussing frameworks/ models for evaluating and enhancing people's maturity. Scholarly journals, research conferences, and technical reports which have passed through peer review. This includes all research that did not focus on software development or security. Some of the articles are not indexed for full-text access. Traditional magazines generally do not contain peer-reviewed articles, op-eds,s or editorials. Figure 2 presents the primary and final selection of papers from different digital libraries.

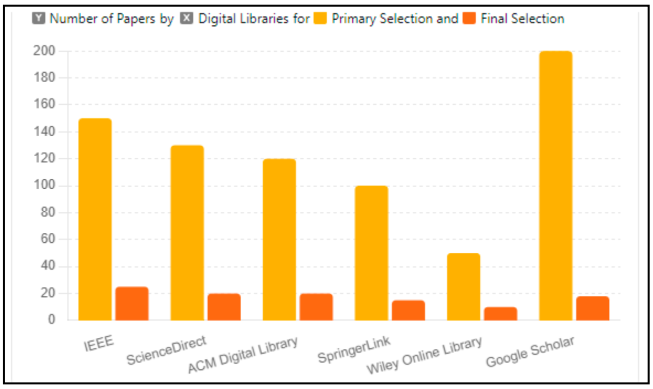


Figure 2. Distribution of the selected papers from different digital libraries.

5). Step 5: Data Extraction

The information relevant to the review question is collected from the chosen articles using a non-structured questionnaire. The extracted data typically includes:

- Study Title
- Authors
- Publication Year
- Research Objectives
- Methodology
- Key Findings
- Limitations
- Recommendations for Future Research

6). STEP 6: Quality Assessment

Quality appraisal of each selected study is done using specific quality criteria. Common quality assessment criteria include:

- Clarity of research objectives
- The rigor of the methodology
- Internal and external validity of the research
- Contribution to the field
- Honesty regarding certain constraints and perspectives

7). STEP 7: Data Synthesis

In addition, patterns, themes, and relations between the studies are established from the extracted data. This refers to the number of themes identified and the style of making the synthesis where one provides narrative synthesis. Concerning the research questions, the findings are posed systematically.

8). STEP 8: The Findings

The results of the SLR are compiled into a detailed report structured in Section IV of this study.

Thus, by following these simple and structured steps, the literature review will help gain a thorough appreciation of people's maturity and significance in secure software development, help identify the missing links, and help develop a perfect and comprehensive Fuzzy-AHP decision-making system.

B. Phase 2: Empirical Study

We performed an empirical study using an online survey to gather insights from SSD experts and practitioners. The focus was on their experiences handling HSFs and HSVs and their practice's impact on SSD projects.

Employing an online survey in this study offered several advantages [41]:

- This method eliminated the necessity for scheduling meetings or global travel, allowing for efficient data collection from experts across continents.
- Utilizing online resources that are both cost-effective and accessible, such as Google Online Form, made the implementation process more practical.

The survey had design and sampling phases. Several systematic and unsystematic sampling methods were considered during the design phase when formulating sample questions [42]. We chose an online survey, a non-scientific data collection method often used for information gathering by other researchers [43–45], because it was more practical than directly collecting data from experts in different countries.

1). Development of Questionnaire Survey

HSFs and HSVs were evaluated in the closed-ended section using a five-point Likert scale, ranging from 'strongly agree' to 'strongly disagree.

2). The Pilot of Questionnaire Survey

To evaluate the questionnaire survey, a pilot study involved experts from software development organizations, namely, the "Cyber-Physical Systems Research Group, University of Southampton, UK", "College of Computer and Information Sciences, King Saud University, Saudi Arabia", "Software Engineering Research Group (SERG_UOM) Pakistan”. Subsequently, expert feedback was incorporated, leading to revisions in the questionnaire survey.

The geographical regions and industries that were included in this study were the USA, Germany, Finland, the UK, Canada, Australia, etc. The industries or sectors being focused on secure software development organizations, Cybersecurity, and IT management.

3). Data Collection Sources

Utilizing snowball sampling, expert data was gathered [46]. Email and numerous social media sites, such as ResearchGate, Facebook, LinkedIn, and Gmail, were used to make contact. Out of 90 responses received from SSD survey participants, 20 were excluded as their expertise didn’t align with secure software development. Subsequently, the final dataset of 70 survey responses underwent manual evaluation.

Respondents predominantly held Master's degrees, while those with PhDs were less represented. Most participants hailed from prominent SSD enterprises, with many holding decision-making roles.

Figure 3 presents the demographic details of the 70 survey participants for identifying human success factors (HSFs) and human security vulnerabilities (HSVs) that impact secure software development (SSD) projects.

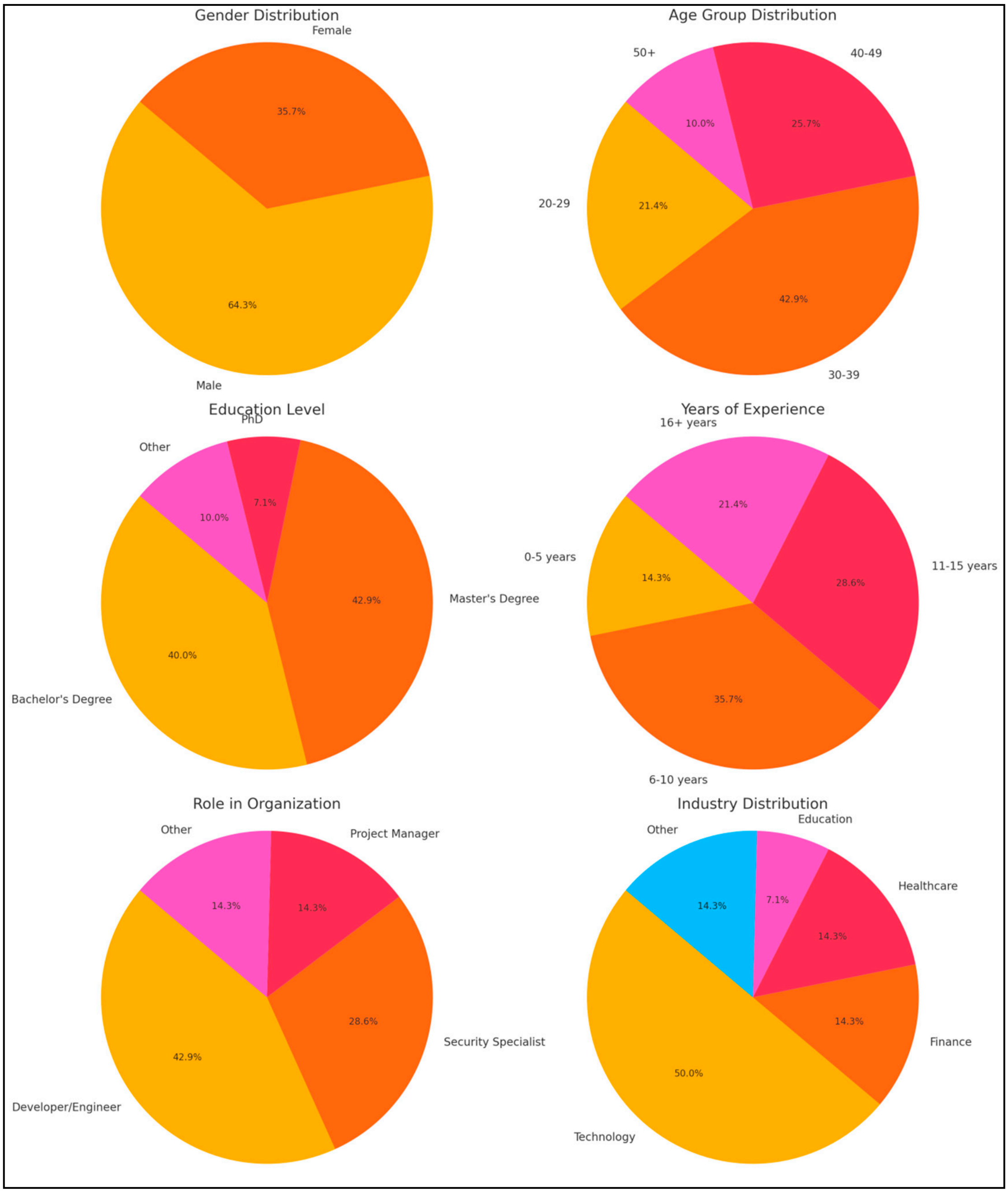


Figure 3. Demographic Details of Survey Participants.

C. Phase 2: Fuzzy Analytical Hierarchy Process (Fuzzy-AHP)

Fuzzy-AHP functions as a decision-making approach, which utilizes Analytic Hierarchy Process principles but incorporates fuzzy logic to process uncertain and imprecise decision situations [47,48]. AHP decision-making processes begin with arranging problems as hierarchical structures with the main decision goal positioned on top followed by evaluation criteria that ultimately lead to candidate alternatives. Users perform pairwise assessments to determine criterion rankings that get converted into numerical data for investigation purposes.

Decision-makers benefit from Fuzzy-AHP because it adds fuzzy logic to enable the use of "somewhat important" and "very important" linguistic expressions instead of numerical values in preference assessments [49]. The approach delivers outstanding support for situations that involve ambiguous data or unquantifiable numerical assessment from decision-makers. Fuzzy-AHP converts verbal decision parameters into triangular or trapezoidal fuzzy numbers that fill the gap of uncertainty during decision-making [50].

We employed Fuzzy-AHP within in this study to establish rankings for various people's maturity and software security components. The framework implements Fuzzy-AHP to process human experts' subjective expertise and decision approaches flexibly and thus helps find optimal security plans for software development systems.

The list of HSF and HSV practices was ranked based on the findings of the fuzzy analytical hierarchy process (Fuzzy-AHP).

Employing Structured Elicitation Methods: We use techniques or methods used to guide the Fuzzy expert elicitation process, such as:

- Anonymity: This strategy ensures that experts are not aware of what others say to avoid peer pressure or to conform to majority views.
- Iterative Process: This approach implements several rounds of elicitation in which experts can revise their assessments based on the group feedback, which can help in refining weights and minimizing bias. Such an iterative method makes the results more robust. Experts, who revised their initial weight assignments following group talks to minimize possible biases and reach agreement, carried out a two-round elicitation process.
- A sensitivity analysis was performed to determine the robustness of the decision-made framework to consider how changes in the weight assignment could potentially impact the ranking of success factors and variables in the overall sense. This enabled one to understand the contributing role of expert judgment on the outcomes better.

1). Fuzzy Set

This section provides an in-depth discussion of fuzzy set theory and fundamental AHP concepts. Initially introduced by Zadeh [51], fuzzy set theory was devised to address uncertainties and vagueness inherent in real-world problems. Its main benefit is representing imprecise data [52]. Within a fuzzy set, membership functions map objects onto a scale between '0' and '1'.

Definition: In Figure 4, a triangular fuzzy number (TFN) denoted as F comprises a set (fl, fm, fu), and its membership function μF (x) is defined by equation (1).

$$\mu_F(x) = \begin{cases} \frac{x-f^l}{f^m-f^l}, & f^l \leq x \leq f^m \\ \frac{f^u-x}{f^u-f^m}, & f^m \leq x \leq f^u \\ 0, & otherwise \end{cases} \qquad (1)$$

when the exact values for the lowest, most significant, and highest ranges are denoted by (fl, fm, fu). Triangular fuzzy numbers (TFNs) for T1 and T2 are mathematically represented in Table 1.

Table 1. Triangular Fuzzy Numbers.

Operation Laws	Expression
Addition ($F_1 \otimes F_2$)	$(f^l{}_1,\ f^m{}_1,\ f^u{}_1) \otimes (f^l{}_2,\ f^m{}_2,\ f^u{}_2) = (f^l{}_1 + f^l{}_2,\ f^m{}_1 + f^m{}_2,\ f^u{}_1 + f^u{}_2)$
Subtraction ($F_1 \otimes F_2$)	$(f^l{}_1,\ f^m{}_1,\ f^u{}_1) \otimes (f^l{}_2,\ f^m{}_2,\ f^u{}_2) = (f^l{}_1 - f^l{}_2,\ f^m{}_1 - f^m{}_2,\ f^u{}_1 - f^u{}_2)$
Multiplication ($F_1 \otimes F_2$)	$(f^l{}_1,\ f^m{}_1,\ f^u{}_1) \otimes (f^l{}_2,\ f^m{}_2,\ f^u{}_2) = (f^l{}_1 * f^l{}_2,\ f^m{}_1 * f^m{}_2,\ f^u{}_1 * f^u{}_2)$
Division ($F_1 \otimes F_2$)	$(f^l{}_1,\ f^m{}_1,\ f^u{}_1) \otimes (f^l{}_2,\ f^m{}_2,\ f^u{}_2) = (f^l{}_1 / f^l{}_2,\ f^m{}_1 / f^m{}_2,\ f^u{}_1 / f^u{}_2)$
Inverse ($F_1 \otimes F_2$)	$(f^l{}_1,\ f^m{}_1,\ f^u{}_1)^{-1} = (1/f^l{}_1,\ 1/f^m{}_1,\ 1/f^u{}_1)$
For any real number k (Kf_1)	$k\,(f^l{}_1,\ f^m{}_1,\ f^u{}_1) = kf^l{}_1,\ kf^m{}_1,\ kf^u{}_1$

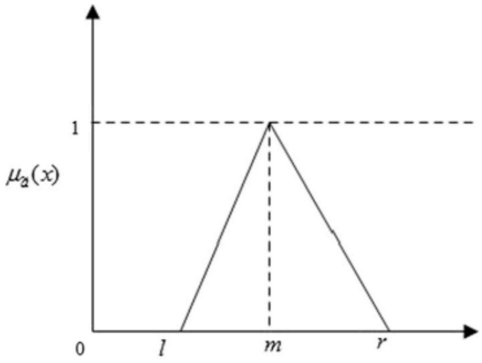


Figure 4. Triangular Fuzzy Number.

2). Fuzzy-AHP

The Analytical Hierarchy Process (AHP) is used to solve "multi-criteria decision-making" (MCDM) problems. The limitations of conventional AHP methods, such as a "crisp environment," an absence of uncertainty consideration, unbalanced judgmental scales, and subjective judgment selection, have been greatly helped by AHP and fuzzy set theory. Thus, in MCDM scenarios involving uncertainties and fuzziness, the Fuzzy Analytical Hierarchy Process (FAHP) has become increasingly popular [53].

Using the Triangular Fuzzy Numbers (TFNs) scale, Fuzzy-AHP allows linguistic terms to be incorporated to collect input from several decision-makers. This provides the opportunity to quantify the linguistic variables. Similar techniques have been utilized to gauge vagueness in fuzzy environments across various engineering disciplines [54]. This study adopts Chang's FAHP [55], which is recognized for its consistency and appropriateness in such analyses.

Imagine a situation where you must prioritize HSF and HSV practices toward SSD projects. There are two sets, X = {x1', x2', … xn} and U = {u1', u2', … un}, where X is the set of objects and U is the set of goals. Chang [55] asserts that each object undergoes measurement while every goal (gi) is pursued. By employing equations (11) and (12), the extent of analysis values (m) for each object can be determined.

$$\widetilde{F^1}_{gi} \; + \; \widetilde{F^2}_{gi} \;, \ldots \; \widetilde{F^m}_{gi} \qquad (2)$$

$$i \; = 1, 2, \ldots, \mathrm{n} \qquad (3)$$

where the TRNs indicate $\widetilde{F^J}_{gi}$ (where j=1, 2, …, m). The following procedures are carried out to apply Chang's extent analysis methodology:

Step 1: The first step is creating a comparison matrix considering fuzzy values.

Step 2: Figure out the fuzzy synthetic extent as it relates to the options

$$Si = \sum_{j=1}^{m} \widetilde{F^J}_{gi} \otimes \left[\sum_{i=1}^{n} \sum_{j=1}^{m} \widetilde{F^J}_{gi}\right]^{-1} \qquad (4)$$

To achieve the expression $\sum_{j=1}^{m} \widetilde{F^J}_{gi}$, "execute the fuzzy addition operation of m extent analysis such as:"

$$\sum_{j=1}^{m} \widetilde{F^J}_{gi} = \left(\sum_{j=1}^{m} \widetilde{f^l}_{gi}, \sum_{j=1}^{m} \widetilde{f^m}_{gi}, \sum_{j=1}^{m} \widetilde{f^u}_{gi}\right) \qquad (5)$$

and to achieve the expression $\left[\sum_{i=1}^{n} \sum_{j=1}^{m} \widetilde{F^J}_{gi}\right]^{-1}$, "the fuzzy addition operation is executed on" $\widetilde{F^J}_{gi}$(j = 1, 2, ...m) value, as follow:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \widetilde{F^J}_{gi} = \left(\sum_{i=1}^{n} \widetilde{f^l}_{i}, \sum_{i=1}^{n} \widetilde{f^m}_{i}, \sum_{i=1}^{n} \widetilde{f^u}_{i}\right) \qquad (6)$$

Finally, calculate the vector's inverse using Eq. (7):

$$\left[\sum_{i=1}^{n} \sum_{j=1}^{m} \widetilde{F^J}_{gi}\right]^{-1} = \left(\frac{1}{\sum_{i=1}^{n} \widetilde{f^u}_{i}}, \frac{1}{\sum_{i=1}^{n} \widetilde{f^m}_{i}}, \frac{1}{\sum_{i=1}^{n} \widetilde{f^l}_{i}}\right) \qquad (7)$$

Step 2: "As F_a and F_b are two triangular fuzzy numbers, then the degree of possibility of" F_a= (f_a^l, f_a^m, f_a^u) ≥ F_b= (f_b^l, f_b^m, f_b^u) is defined as follows. The Equation (8) is also given below:

$$\mathrm{V}(F_a \geq F_b) = \sup\ [\min(\mu_{Fa}(x), \mu_{Fb}(x))] \qquad (8)$$

$$\mathrm{V}(F_a \geq F_b) \quad (F_a \cap F_b) = \mu_{Fa}(d) = \begin{cases} 1, & if\ f_a^m \geq f_b^m \\ \frac{f_a^u - f_b^l}{(f_a^u - f_b^m)+(f_b^m - f_b^l)}, & f_b^l \leq f_a^u \\ 0, & otherwise \end{cases} \quad (9)$$

Here, the ordinate of the highest point where d, μ_Fa, and μ_Fb intersect is represented by the value of d (Figure 5).To find out what P1 and P2 are worth, you need to know the values of V1(F_a≥ F_b) and V2(F_a≥ F_b).

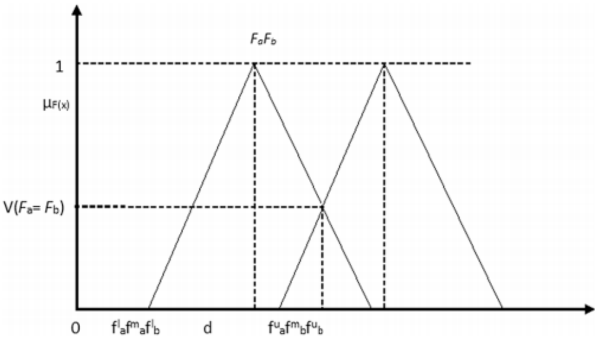


Figure 5. Intersection between Triangular Fuzzy Numbers.

Step 3: The following is one way to express the process of finding the full "degree of possibility of a convex fuzzy number and other convex fuzzy numbers Fi (i=1, 2,…, k)":

$$V(F \geq F1, F2, F3, \ldots Fk) = \min V(F \geq Fi) \qquad (10)$$

Assuming:

$$d/Fi = \min V(Fi \geq Fk) \qquad (11)$$

for k=1,2,…,n; k≠i.

It is through the utilization of Equation 21 that the weight vector is established.

$$\mathrm{W}^{/} = (\mathrm{d}^{/}(\mathrm{F}_1), \mathrm{d}^{/}(\mathrm{F}_2), \mathrm{d}^{/}(\mathrm{F}_3) \ldots, \mathrm{d}^{/}(\mathrm{F}_n)) \qquad (12)$$

in which Fi are definite variables where i=1,2,...,n.

Step 4: The weight vector obtained from Equation (14) and Equation (13) standardizes it. This normalized non-fuzzy value indicates priority weight for HSF and HSV practices.

$$\mathrm{W} = (\mathrm{d}(\mathrm{F}_1), \mathrm{d}(\mathrm{F}_2), \mathrm{d}(\mathrm{F}_3) \ldots, \mathrm{d}(\mathrm{F}_n)) \qquad (13)$$

with W standing for the weight of a security risk's priority.

Step 5: Fifthly, ensure that the matrices used to compare options in fuzzy AHP are consistent [56]. The Consistency Ratio (CR) calculation for each matrix is essential. Fuzzy matrices can be created using the graded-mean method. The triangular fuzzy matrix P = (l, m, u) is transformed using Equation 14:

$$\mathrm{P}_{\mathrm{crisp}} = \frac{(4m+l+u)}{6} \qquad (14)$$

The final consistency ratio is calculated using equations (15) and (16):

$$\mathrm{CI} = \frac{(\lambda_{max} - n)}{n-1} \qquad (15)$$

$$\mathrm{CR} = \frac{CI}{RI} \qquad (16)$$

Where λ_{max} represents "the largest eigenvalue of the comparison matrix", n is "the number of items being compared in the matrix," and RI is "the random index. Its value can be referenced from Table 2." CI is "the consistency index" computed using equation (16). The matrix maintains consistency if CR is below 0.1; otherwise, decision-makers should resort to pair-wise judgments.

Table 2. Random Consistency Index (RI) concerning Matrix Size.

Matrix size	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

IV. Results and Data Analysis

In this section, we describe the detailed analysis of the data acquired through the empirical survey and FAHP used to answer the research questions (RQs) posed in Section I:

A. RQ1: What Are the Human Factors (Success Factors and Security Vulnerabilities), as Reported in the Literature and Real-World Industry, That Influence Secure Software Development Projects?

Several human factors significantly influence success in secure software development (SSD) projects. These factors encompass various aspects of team dynamics, knowledge, communication, and organizational culture. Similarly, various human factors in SSD projects can introduce security vulnerabilities that compromise the software's integrity, confidentiality, and availability. Understanding these factors is crucial to mitigating risks and ensures the SSD process.

Table 3 presents the key human factors (success factors and security vulnerabilities) that impact SSD projects.

Table 3. Human Factors (Success Factors and Security Vulnerabilities), as identified in the Literature Review and Real-World Industry that Influence SSD Projects.

Human Success Factors (HSFs)	**Description**		**Human Security Vulnerabilities (HSVs)**	**Description**	
HSF1: Skill and Expertise [57]	**Technical Proficiency:** The team must possess strong technical skills in software development, cybersecurity principles, and secure coding practices. Expertise in using security tools and technologies is also crucial.	**Continuous Learning:** Encourage constant learning and professional development to keep team members updated with the latest threats, vulnerabilities, and mitigation techniques.	**HSV1: Lack of Security Awareness and Training [17]**	**Insufficient Knowledge:** Developers and other team members who lack proper training in secure coding practice and Cybersecurity principles are more likely to introduce vulnerabilities into the software.	**Outdated Knowledge:** Not staying updated with the latest security threats, trends, and best practices can result in obsolete and insecure methods.
HSF2: Awareness and Mindset [17]	**Security Awareness:** Every team member, from developers to project managers, should know the importance of security in the SDLC. Regular training and awareness programs can	**Security-First Mindset:** Cultivating a security-first mindset means prioritizing security considerations in every phase of the SDLC. This involves integrating security practices into the	**HSV2: Poor Communication and Collaboration [58]**	**Miscommunication:** Inadequate communication between team members, especially between developers and security experts, can lead to misunderstandings and	**Siloed Teams:** Lack of collaboration between teams (development security, QA, operations) can result in incomplete security reviews and

	help reinforce this mindset.	development process rather than the team treating them as an afterthought.		overlooked security requirements.	unaddressed vulnerabilities.
HSF3: Communication and Collaboration [58]	**Effective Communication:** Clear and open communication channels within the team and with stakeholders are essential. This ensures that security requirements, risks, and mitigation strategies are well understood and properly implemented.	**Collaboration:** Cross-functional collaboration between developers, security experts, quality assurance teams, and operations staff is crucial. Collaborative efforts help identify potential security issues early and address them efficiently.	**HSV3: Inadequate Testing and Review [17]**	**Insufficient Security Testing:** Failing to perform comprehensive security testing, such as static code analysis, dynamic testing, and penetration testing, can leave vulnerabilities undetected.	**Lack of Peer Reviews:** Skipping code reviews or conducting superficial reviews can allow security flaws to persist in the codebase.
HSF4: Leadership and Management [48]	**Supportive Leadership:** Leaders and managers should support and advocate for secure development practices. This includes allocating necessary resources, providing training opportunities, and emphasizing the importance of security in project goals.	**Risk Management:** Effective leadership involves identifying, assessing, and managing security risks throughout the project. Proactive risk management helps mitigate potential threats before they become critical issues.	**HSV4: Pressure and Workload [59]**	**Time Constraints:** Tight deadlines and high workload pressures can lead to cutting corners, skipping security checks, and hastily implementing code that may contain vulnerabilities.	**Burnout:** Overworked and fatigued developers are more prone to making mistakes and overlooking critical security.
HSF5: Process and Methodology [17]	**Defined Processes:** Establishing well-defined processes and methodologies for SSD, such as Secure Development Lifecycle (SDL) or DevSecOps, helps ensure that security practices are	**Compliance and Standards:** Adhering to industry standards and regulatory requirements, such as ISO/IEC 27001, NIST, or GDPR, ensures that security practices are aligned with best	**HSV5: Unclear Roles and Responsibilities [60]**	**Ambiguous Accountability:** When security responsibilities are unclear, important security tasks may be neglected or assumed to be someone else’s responsibility.	**Lack of Ownership:** Without clear ownership of security tasks, there may be a lack of accountability and commitment to secure

	systematically followed.	practices and legal obligations.			coding practices.
HSF6: Culture and Environment [60]	**Security Culture:** Building a security-centric culture within the organization promotes vigilance and accountability. A strong security culture encourages team members to identify and address security concerns proactively.	**Psychological Safety:** Creating an environment where team members feel safe to report security issues without fear of blame or retribution fosters openness and prompt resolution of potential vulnerabilities.	**HSV6: Resistance to Change [61]**	**Inertia:** Resistance to adopting new security tools, practices, or frameworks due to comfort with existing methods can hinder the implementation of more secure practices.	**Fear of Disruption:** Concerns about disrupting existing workflows or delaying projects can lead to resistance against integrating security measures into the development process.
HSF7: Accountability and Responsibility [62]	**Clear Roles and Responsibilities:** Clearly define responsibilities related to security within the team to ensure that everyone knows their part in maintaining and enhancing software security.	**Ownership:** Encouraging team members to take ownership of their work and its security implications leads to higher quality and more SSD outcomes.	**HSV7: Insufficient Resources [63]**	**Limited Budget:** Inadequate funding for security training, tools, and resources can prevent the adoption of necessary security measures.	**Lack of Access to Expertise:** Not having access to experienced security professionals for guidance and support can leave security gaps unaddressed.
HSF8: Resource Allocation [64]	**Adequate Resources:** Providing sufficient resources, including time, budget, and tools, is essential for implementing and maintaining SSD practices.	**Access to Expertise:** Ensuring access to security experts, whether in-house or through external consultants, helps address complex security challenges effectively.	**HSV8: Human Error [65]**	**Coding Mistakes:** Simple coding errors, such as improper input validation, hardcoded credentials, or incorrect configurations, can introduce significant vulnerabilities.	**Configuration Errors:** Misconfigurations in development environments, servers, and applications can create exploitable security gaps.
HSF9: Testing and Validation [66]	**Comprehensive Testing:** Regular and thorough security testing, including code reviews, penetrating testing, and vulnerability assessments, helps identify	**Feedback Loops:** Establishing feedback loops for continuous improvement that ensures lessons learned from past projects are applied to future endeavors, enhancing	**HSV9: Neglecting Secure Development Lifecycle [67]**	**Inconsistent Processes:** Failing to adhere to a standardized, secure development lifecycle (SDL) can result in inconsistent application of security	**Ignoring Best Practices:** Not following established best practices for secure development can introduce common

	and rectify security weaknesses.	overall security practices.		practices and unaddressed vulnerabilities.	vulnerabilities.
HSF10: Incident Response and Recovery [68]	**Preparedness:** A well-defined incident response plan ensures the team can handle security breaches effectively. This includes identifying, responding to, and recovering from security incidents.	**Learning from Incidents:** Analyzing and learning from security incidents helps improve security measures and prevent future occurrences.	**HSV10: Cultural Issues [69]**	**Lack of Security Culture:** An organizational culture that does not prioritize security or view it as everyone's responsibility can lead to neglect of security practices.	**Blame Culture:** A culture that penalizes individuals for reporting security issues can discourage team members from identifying and addressing vulnerabilities.
HSF11: Scalability and Flexibility [70]	**Adaptable Integration:** Ensuring that secure development practices can be integrated into various healthcare settings, from small clinics to large hospitals, and tailored to meet specific needs.	**Scalable Solution:** Develop solutions that can scale up to handle increased volumes of interactions and data while maintaining security standards.	**HSV11: Insufficient Incident Response Preparedness [68]**	**Lack of Preparedness:** Inadequate incident response planning and training can result in poor handling of security incidents, leading to prolonged exposure to vulnerabilities.	**Failure to learn from Incident:** Not analyzing and learning from past security incidents can result in recurring vulnerabilities and security breaches.
HSF12: Ethical and Cultural Sensitivity [69]	**Ethical Considerations:** Addressing ethical considerations comprehensively ensures patient confidentiality, informed consent, and equitable access to new interventions.	**Cultural Sensitivity:** Recognizing and respecting cultural differences in healthcare practices and patient expectations enhances acceptance and effectiveness of secure software.	**HSV12: Over-Reliance and Tools [71]**	**Tool Dependency:** Relying too heavily on automated security tools without human oversight can lead to missed vulnerabilities that require contextual understanding and manual intervention.	**False Sense of Security:** Assuming that using security tools alone is sufficient can lead to complacency and underestimation of the need for thorough security practices.

By focusing on these human success factors (HSFs) and human security vulnerabilities (HSVs), organizations can significantly enhance the success of their SSD projects, resulting in more resilient and trustworthy software solutions.

B. RQ2: What Are the Similarities and Differences Between HSFS and HSVS, as Identified Through Literature and Real-World Industries, That Influence SSD Projects?

To analyze the similarities and differences between human success factors (HSFs) and human security vulnerabilities factors (HSVFs) as identified through literature and real-world industries that influence secure software development (SSD), we considered how these factors and vulnerabilities are perceived and addressed in both contexts (as presented in Tables 4 and 5).

Table 4. Similarities and Differences of HSFs identified through Literature Review and Real-World Industries.

HSFs	Literature	Real-World Industries	Similarities	Differences
HSF1: Skill and Expertise	Emphasizes the need for technical proficiency and continuous learning	Highlights the importance of training and identifying skill gaps in practical scenarios	Both stress the importance of technical skills and ongoing education	The literature review focuses more on the ideal skill sets and continuous learning, while surveys highlight real-world skill deficiencies
HSF2: Awareness and Mindset	Stress the importance of security awareness and a security-first mindset throughout the SDLC.	Indicates a lack of security awareness as a common issue among practitioners	Both emphasize the critical role of awareness and mindset in ensuring security.	Literature review emphasizes theoretical importance, while surveys highlight practical shortcomings in awareness and mindset.
HSF3: Communication and Collaboration	Highlights the importance of clear communication and cross-functional collaboration	Identifies poor communication and collaboration as significant issues impacting security	Both recognize that effective communication and collaboration are essential for security.	The literature review discusses ideal communication strategies, while surveys focus on practical communication breakdowns.
HSF4: Leadership and Management	Discusses the role of supportive leadership and proactive risk management	Often highlights a lack of clear leadership and accountability in practice	Both agree on the importance of solid leadership and management in promoting security	The literature review provides theoretical frameworks for leadership, while surveys offer specific examples of leadership deficiencies
HSF5: Process and Methodology	Advocates for defined processes and	Points out the neglect of secure development	Both stress the necessity of structured	The literature review focuses on ideal

	adherence to secure development lifecycles	lifecycles and methodologies in practice	processes and methodologies for security.	processes, while the survey highlights the practical neglect of these processes.
HSF6: Culture and Environment	Emphasizes the need for a security-centric culture and supportive environment	Identifies cultural issues and resistance to change as significant barriers to security	Both recognize the influence of organizational culture on security practices.	The literature review discusses theoretical and cultural frameworks, while surveys address specific organizational cultural challenges.
HSF7: Accountability and Responsibility	Stresses the importance of clear roles and responsibilities in maintaining security	Highlights the impact of unclear roles and responsibilities on security practices in real-world scenarios	Both emphasize the need for clear accountability to ensure security	A literature review may discuss theoretical role definition, while surveys highlight real-world ambiguities and their impact on security
HSF8: Resource Allocation	Discusses the need for adequate resources, including time, budget, and tools, for secure development	Identifies insufficient resources as a common issue affecting security efforts	Both stress the importance of resource allocation for maintaining security	The literature review focuses on ideal resource allocation, while surveys highlight practical
HSF9: Testing and Validation	Advocates for thorough security testing and validation processes throughout the development lifecycle	Points out inadequate testing and review as significant vulnerabilities in practice	Both agree on the importance of comprehensive testing and validation	Literature review details ideal testing methodologies, while surveys reveal practical deficiencies in testing and validation efforts
HSF10: Incident Response and Recovery	Emphasizes the need for preparedness and learning from security incidents to improve future responses	Highlights insufficient incident response preparedness and real-world challenges in handling incidents	Both stress the importance of having robust incident response and recovery plans.	The literature review discusses theoretical incident response plans, while surveys highlight practical gaps

				and challenges in preparedness.
HSF11: Scalability and Flexibility	Discuss the need for adaptable and scalable security solutions that can grow with the organization.	Often implied in resource allocation and process flexibility but not always explicitly addressed.	Both recognize the need for scalability and flexibility in security practices.	The literature review explicitly addresses scalability and flexibility, while the survey focuses more on immediate practical concerns.
HSF12: Ethical and Cultural Sensitivity	Highlights the importance of ethical considerations and cultural sensitivity in SSD	Often implied in broader cultural issues and security awareness but not always explicitly addressed.	Both touch upon the importance of considering ethical and cultural factors in security practices	Literature review explicitly addresses ethical and cultural frameworks, while surveys focus on immediate practical concerns.

Table 5 summarizes the similarities and differences between human security vulnerabilities (HSVs) identified through literature and those observed in real-world industries, clearly comparing common themes and unique challenges faced in both contexts.

Table 5. Similarities and Differences of HSVs identified through Literature Review and Real-World Industries.

HSVs	Literature	Real-World Industries	Similarities	Differences
HSV1: Lack of Security Awareness and Training	Emphasizes the importance of continuous education and awareness programs	Highlights frequent security breaches due to sufficient knowledge	Both stress the need for awareness and training in security practices	Literature focuses on theoretical frameworks, while the real world highlights practical training gaps
HSV2: Poor Communication and Collaboration	Academic studies underline the importance of clear communication and collaboration.	Industries experience issues where a lack of collaboration leads to security gaps	Both recognize the importance of effective communication and collaboration	Literature provides ideal communication strategies, while the real world highlights practical communication issues
HSV3: Inadequate Testing and Review	Security vulnerabilities often arise from inadequate testing and review processes.	Time constraints and resource limitations result in rushed or incomplete testing.	Both stress the importance of comprehensive testing and validation	Literature details ideal testing methodologies, while the real world highlights practical deficiencies of testing efforts

HSV4: Unclear Roles and Responsibilities	Research identifies the need for clearly defined roles and responsibilities to avoid security issues.	Unclear responsibilities can lead to essential security tasks being overlooked or improperly executed.	Both agree on the importance of clearly defined roles and responsibilities in promoting security.	Literature provides theoretical frameworks for roles, while the real world highlights practical issues related to unclear roles.
HSV5: Insufficient Resources	Emphasizes the need for adequate resources, including budget, tools, and personnel	Budget constraints often lead to prioritizing other operational needs over security investments.	Both stress the importance of resource allocation for security	Literature focuses on ideal resource allocation, while the real world highlights practical resource constraints
HSV6: Human Error	Academic work acknowledges human error as a critical factor in security breaches.	Human error is a common cause of security incidents in industry reports	Both agree on the importance of addressing human error in security practices	Literature provides theoretical approaches, while the real world highlights practical human error issues
HSV7: Pressure and Workload	Less commonly highlighted compared to other vulnerabilities	High pressure and workload are frequently cited as major contributors to security lapses	Both recognize the impact of workload on security practices	Literature rarely addresses this as a standalone factor, while the world highlights it as a significant practical issue
HSV8: Resistance to Change	They are often discussed in a more theoretical context.	They are frequently a barrier to implementing new security measures and protocols.	Both acknowledge that resistance to change can impede security improvements.	The literature discusses it theoretically, while the world focuses on practical resistance issues.
HSV9: Neglecting Secure Software Development Lifecycle	Stresses the importance of incorporating security throughout the development lifecycle	They are often neglected due to cost, time constraints, or lack of immediate perceived benefits.	Both emphasize the necessity of including security in every development lifecycle phase.	Literature focuses on ideal processes, while the world highlights the practical neglect of secure development practices.
HSV10: Cultural Issues	Emphasizes the importance of a security-centric culture within organizations	Struggles to integrate security into core values and daily practices	Both recognize the influence of organizational culture on security practices	Literature provides theoretical and cultural frameworks, while the real world addresses specific cultural challenge

HSV11: Over-Reliance on Tools	Advocates for a balanced approach combining tools with human oversight	Often places excessive reliance on automated tools, underestimating the need for human judgment	Both highlight the risks of relying solely on automated tools for security	Literature focuses on balanced approaches, while the world highlights practical issues due to over-reliance on tools

Table 6 provides a comparative analysis of the impact percentage of human success factors (HSFs) and human security vulnerabilities (HSVs) identified through literature review and empirical study.

Table 6. Comparative Analysis of the Impact Percentage of HSFs and HSVs identified through Literature Review and Empirical Survey on SSD Projects.

Human Success Factors (HSFs)	Literature Review Impact %	Empirical Survey Impact %	Human Security Vulnerabilities (HSVs)	Literature Review Impact %	Impact Percentage
HSF1	94	90	HSV1	90	91
HSF2	91	85	HSV2	89	85
HSF3	86	81	HSV3	78	84
HSF4	82	79	HSV4	75	82
HSF5	79	78	HSV5	69	78
HSF6	78	76	HSV6	68	66
HSF7	75	73	HSV7	64	60
HSF8	73	72	HSV8	56	59
HSF9	71	70	HSV9	50	52
HSF10	63	60	HSV10	47	49
HSF11	60	52	HSV11	45	43
HSF12	55	50	HSV12	44	40

Key Observation about Table 6:

- **Top Success Factors:** HSF1 (Skill and Expertise) and HSF2 (Awareness and Mindset) are rated highest in the literature review and empirical survey, indicating their critical importance in SSD projects.
- **Consistency:** Most factors show high consistency between literature and survey impacts, reflecting a general effect on their importance.
- **Lower Impact Factors:** HSF11 (Scalability and Flexibility) and HSF12 (Ethical and Cultural Sensitivity) are rated lower, suggesting these are less emphasized in practical settings than technical and managerial skills.
- **Top Security Vulnerabilities:** HSV1 (Lack of Security Awareness and Training) and HSV2 (Poor Communication and Collaboration) are consistently rated highest, highlighting their significant impact on security.
- **Variability:** There is more variability in the impact percentages of vulnerabilities than success factors, suggesting differences in perceptions of vulnerability impacts.
- **Lower Impact Vulnerabilities:** HSV10 (Cultural Issues), HSV11 (Insufficient Incident Response Preparedness), and HSV12 (Over-Reliance on Tools) are rated lower, indicating these are perceived as less critical compared to others.

Figure 6 depicts that both the literature review and empirical survey data agree that skills and expertise (HSF1), awareness and mindset (HSF2), and communication and collaboration (HSF3) are critical success factors with very high impact percentages. This indicates a strong consensus on the importance of continuous education and awareness programs to secure the software development processes. Similarly, effective communication and collaboration are crucial for identifying and addressing security issues. Ethical and cultural sensitivity (HSF11), scalability, and flexibility (HSF12) are rated less impactful than other factors.

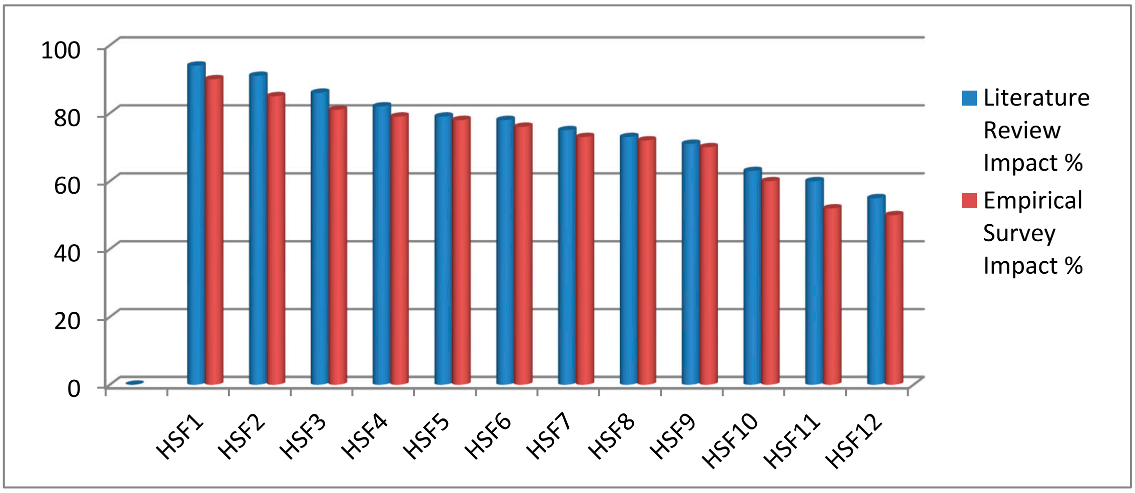


Figure 6. Comparison of Human Success Factors identified through Literature review and Empirical Survey Impact on SSD Projects.

Figure 7 shows a high level of agreement between literature reviews and empirical surveys on the impact of various human security vulnerabilities. The percentages are consistently close across most categories. The most critical vulnerabilities identified are lack of awareness and training (HSV1), poor communication and collaboration (HSV2), and inadequate testing and reviews (HSV3). Cultural issues (HSV10) and over-reliance on tools (HSV12) are related as less impactful compared to other factors.

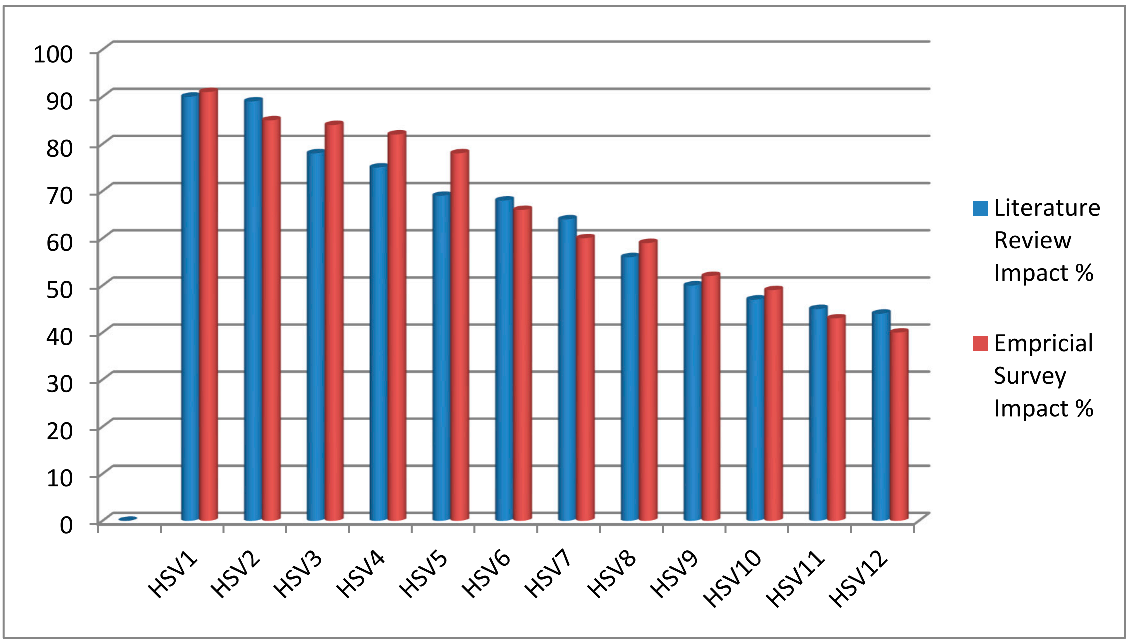


Figure 7. Comparison of Human Security Vulnerabilities identified through Literature review and Empirical Survey Impact on SSD Projects.

C. RQ3: What Are the Practices for Addressing HSFS and HSVS, as Identified Through Literature and Real-World Industries, That Influence SSD Projects?

Addressing human success factors (HSFs) in secure software development (SSD) projects requires a comprehensive approach that balances technical excellence with well-being, collaboration, and continuous improvement of the development team. Table 7 presents some best practices identified through literature review and empirical study.

Table 7. Practices for Addressing HSFs that Impact SSD Projects.

Practices for Addressing Human Success Factors	**Sub-Practices**		
Security Training and Awareness	**Regular Training:** Provide ongoing security training to update the development team on the latest threats, secure coding practices, and regulatory requirements.	**Security Champions:** Identify and empower security champions within the team to advocate for and enforce security best practices.	**Awareness Campaigns:** Run regular security awareness campaigns to reinforce the importance of security in the development lifecycle.
Collaborative Culture	**Team Collaboration:** Foster a culture of collaboration where security is a shared responsibility across all team members, including developers, testers, and operations.	**Cross-Functional Teams:** Encourage forming cross-functional teams that include security experts to integrate security considerations into all stages of development.	**Knowledge Sharing:** Promote knowledge sharing through internal workshops, seminars, and collaborative tools.
Leadership and Management Support	**Executive Support:** Ensure strong support from leadership to prioritize security and allocate necessary resources.	**Clear Vision:** Communicate a clear vision and strategy for integrating security into the software development process.	**Empowerment:** Empower team members to take ownership of security practices and make decisions that enhance security.
Effective Communication	**Open Channels:** Establish open and effective communication channels to discuss security concerns and share updates.	**Regular Meetings:** Hold regular security-focused meetings to review potential vulnerabilities, share best practices, and update the team on new security threats.	**Feedback Mechanisms:** Implement feedback mechanisms to gather input from team members on security practices and improvement areas.
Skill Development	**Continuous Learning:** Encourage continuous learning and professional	**Mentorship Programs:** Implement mentorship programs where	**Hands-On Practice:** Provide opportunities for hands-on practice with

	development in security through certifications, courses, and conferences.	experienced security professionals guide less experienced team members.	secure coding, threat modeling, and penetration testing.
User-Centric Security	**User Involvement:** To understand their security concerns and expectations, involve end-users early in the development process.	**User Feedback:** Collect and act on user feedback regarding security features and usability.	**Usability Testing:** Conduct usability testing focusing on security features to ensure they are user-friendly and effective.
Agile and DevSecOps Practices	**Agile Security Integration:** Integrate security practices into Agile workflows to ensure continuous attention to security.	**DevSecOps:** Adopt DevSecOps practices to automate security testing and integrate security into the CI/CD pipeline.	**Security Retrospectives:** Conduct regular security retrospectives to review security incidents, identify lessons learned, and plan improvements.
Clear Roles and Responsibilities	**Defined Roles:** Clearly define roles and responsibilities related to security within the development team.	**Role Flexibility:** Allow role flexibility to adapt to changing security needs and leverage team members' strengths.	**Accountability:** Ensure accountability for security practices and outcomes across all team members.
Stakeholder Engagement	**Stakeholder Communication:** Maintain regular communication with stakeholders to inform them about security measures and risks.	**Expectation Management:** Manage stakeholder expectations by being transparent about security risks and the measures in place to address them.	**Stakeholder Involvement:** Involve stakeholders in security planning and decision-making processes.
Risk Management	**Risk Identification:** Proactively identify potential security risks that could impact the project.	**Mitigation Strategies:** Develop and implement risk mitigation strategies to address identified security risks.	**Incident Response Plans:** Have incident response plans to address security breaches and quickly minimize impact.
Continuous Improvement	**Performance Metrics:** Use security performance metrics to assess security practices' effectiveness and identify areas for improvement.	**Process Optimization:** Continuously optimize security processes based on feedback and performance data.	**Innovation Encouragement:** Encourage innovation in security practices and use new tools and techniques to enhance security.

By implementing Table 7 practices, SSD projects can effectively address HSFs, leading to improved security outcomes, enhanced team performance, and higher stakeholder satisfaction.

Addressing human security vulnerabilities (HSVs) in SSD projects involves training, process improvements, technology, and a strong security culture. Table 8 presents some practices identified through literature review and real-world industries.

Table 8. Practices for Addressing HSVs that Impact SSD Projects.

Practices for Addressing Human Security Vulnerabilities	**Sub-Practices**		
Comprehensive Security Training and Awareness	**Regular Training:** Conduct regular training sessions to inform the development team about the latest security threats, secure coding practices, and compliance requirements.	**Phishing Simulations:** Implement phishing simulations to educate employees on recognizing and responding to phishing attacks.	**Security Certifications:** Encourage team members to obtain security certifications such as CISSP, CEH, or CSSLP.
Security Culture and Leadership	**Security Champions:** Designate security champions within development teams to promote security best practices and act as liaisons with the security team.	**Leadership Commitment:** Ensure that leadership demonstrates a commitment to security by prioritizing it in all aspects of the project.	**Reward and Recognition:** Recognize and reward team members who contribute to improving security within the project.
Effective Communication and Collaboration	**Clear Communication Channels:** Establish clear communication channels for reporting security issues and sharing security-related information	**Cross-Functional Teams:** Create cross-functional teams with security experts to integrate security considerations throughout the development lifecycle.	**Regular Security Briefings:** Hold regular security briefings to update the team on new threats and security incidents.
Access Control and Least Privilege	**Role-Based Access Control (RBAC):** Implement RBAC to ensure that team members have access only to the resources they need to perform their jobs.	**Least Privilege Principle:** Apply the principle of least privilege to minimize the potential impact of security breaches,	**Multi-Factor Authentication (MFA):** Use MFA to add extra security to critical systems and applications.
Secure Development Practices	**Secure Coding Standards:** Adopt and enforce secure coding standards (e.g., OWASP Secure Coding Practices) to prevent common vulnerabilities.	**Code Review:** Regular code reviews should focus on security to identify and address vulnerabilities early.	**Static and Dynamic Analysis:** Use static and dynamic analysis tools to detect security issues in the code.
Incident Response and Management	**Incident Response Plan:** Develop and maintain an incident response plan to quickly and effectively address security incidents.	**Simulation Drills:** Conduct regular incident response drills to ensure the team is prepared to handle security breaches.	**Post-Incident Analysis:** Perform post-incident analysis to learn from security incidents and improve future responses.
Continuous Monitoring and Testing	**Vulnerability Scanning:** Implement regular vulnerability scanning to identify and address security weaknesses.	**Penetration Testing:** Conduct penetration testing to simulate attacks and evaluate the effectiveness of security measures.	**Security Audits:** Perform regular security audits to ensure compliance with security policies and standards.
Data Protection and Privacy	**Encryption:**	**Data Minimization:** Collect and retain only the data necessary for the	**Privacy by Design:** Incorporate privacy considerations into the

	Use encryption to protect sensitive data both at rest and in transit.	project to reduce the risk of data breaches.	design and development of software to protect user data.
Supply Chain Security	**Third-Party Risk Management:** Assess and manage the security risks associated with third-party vendors and partners.	**Secure Supply Chain:** Implement security measures to protect the software supply chain, including code signing and verification.	**Contractual Security Requirement:** Include security requirements in contracts with third-party vendors to ensure they adhere to security best practices.
User Education and Support	**Security Awareness for Users:** Educate users about security best practices and how to recognize potential threats.	**User-Friendly Security Features:** Design security features that are user-friendly and encourage proper security behaviors.	**Support Channels:** Provide clear support channels for users to report security concerns and get help with security-related issues.
Regulatory Compliance and Standards	**Compliance Programs:** Establish compliance programs to adhere to relevant security regulations and standards (e.g., GDPR, HIPAA).	**Regular Audits:** Conduct regular audits to ensure ongoing compliance with security regulations and standards.	**Documentation:** Maintain thorough documentation of security policies, procedures, and compliance efforts.

By implementing Table 8 practices, SSD projects can effectively address HSVs, leading to more resilient and improved security outcomes.

D. RQ4: How Can a Decision-Making Framework Be Developed to Identify the Weights of HSFS and HSVS Practices That Impact SSD Projects?

In this paper, we employed the coding system of Strauss' [72] ground theory (GT) technique to identify, classify, and organize the identified practices for HSFs and HSVs that impact SSD projects. Although we have already collected the data through a literature review, we used the four main phases of the GT coding scheme to map the practices into four major categories (i.e., "code," "categories", "sub-categories," and theory/theoretical model"). All the authors of this study are on the mapping team. First, we allocated a unique code/label to each practice we studied. The second phase involved categorizing the examined practices into nine broad phases/groups, “C1: Security Training and Awareness”, “C2: Collaborative Security Culture and Leadership”, “C3: Effective Communication and Collaboration”, “C4: Skill Development and Stakeholder Engagement”, “C5: Agile and DevSecOps Practices”, “C6: User-Centric Security, Clear Roles and Responsibilities”, “C7: Supply Chain Security and Risk Management”, “C8: Continuous Improvement, Monitoring and Testing”, and “C9: Data Protection and Privacy, Access Control and Least Privilege”. In the third step, the sub-practices were mapped into these categories. In the fourth step, we engineered a theoretical model depicted in Figure 8.

The primary goal of this categorization is to construct a hierarchical framework for executing the FAHP. Furthermore, this categorization will help academic researchers and practitioners to identify the most essential practice for HSFs and HSVs in the context of SSD. We identified 38 practices mapped in each category, as stated in Table 9.

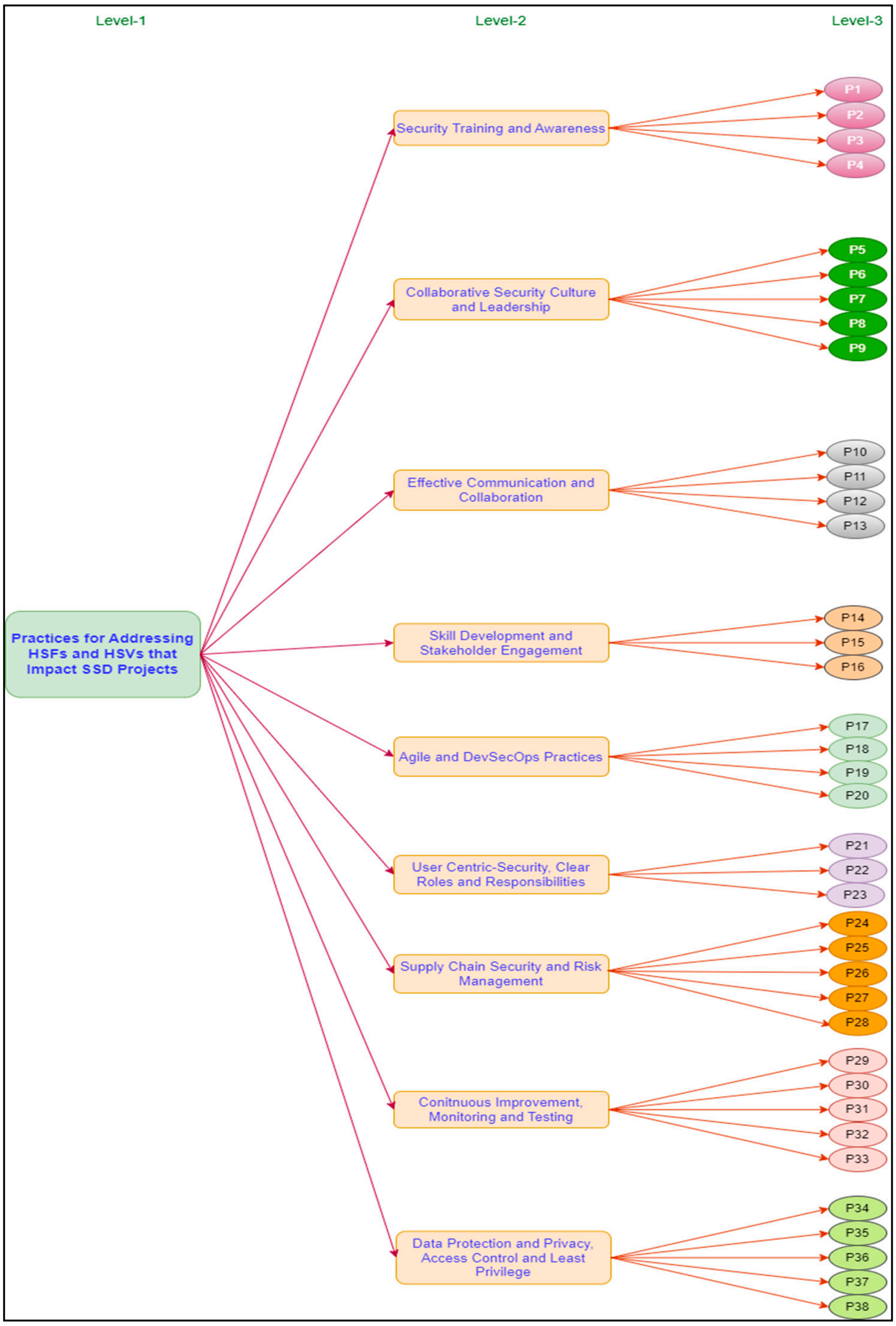

Level-1
Level-2
Level-3
Practices for Addressing HSFs and HSVs that Impact SSD Projects
Security Training and Awareness
Collaborative Security Culture and Leadership
Effective Communication and Collaboration
Skill Development and Stakeholder Engagement
Agile and DevSecOps Practices
User Centric-Security, Clear Roles and Responsibilities
Supply Chain Security and Risk Management
Conitnuous Improvement, Monitoring and Testing
Data Protection and Privacy, Access Control and Least Privilege
P1
P2
P3
P4
P5
P6
P7
P8
P9
P10
P11
P12
P13
P14
P15
P16
P17
P18
P19
P20
P21
P22
P23
P24
P25
P26
P27
P28
P29
P30
P31
P32
P33
P34
P35
P36
P37
P38

Figure 8. Theoretical Model of the Practices for HSFs and HSVs that Impact on SSD Projects.

Table 9. HSFs and HSVs Practices Categories and Subcategories.

HSFs and HSVs: Practices Categories	HSFs and HSVs: Practices Subcategories
C1: Security Training and Awareness	P1: Regular Training and Phishing Simulations P2: Security and Awareness Campaigns P3: Security Certifications and Support Channels P4: User-Friendly Security Features
C2: Collaborative Security Culture and Leadership	P5: Team Collaboration and Cross-Functional Teams P6: Knowledge Sharing and Executive Support P7: Leadership Commitment

	P8: Reward and Recognition P9: Clear Vision and Empowerment
C3: Effective Communication and Collaboration	P10: Open Channels and Regular Meetings P11: Feedback Mechanisms P12: Clear Communication Channels P13: Regular Security Briefings
C4: Skill Development and Stakeholder Engagement	P14: Continuous Learning and Mentorship Programs P15: Hands-On Practice and Stakeholder Communication P16: Expectation Management and Stakeholder Involvement
C5: Agile and DevSecOps Practices	P17: Agile Security Integration and DevSecOps P18: Security Retrospectives P19: Secure Coding Standards P20: Code Review, Static and Dynamic Analysis
C6: User-Centric Security, Clear Roles and Responsibilities	P21: Defined Roles and Role Flexibility P22: Accountability and Usability Testing P23: User Involvement and User Feedback
C7: Supply Chain Security and Risk Management	P24: Risk Identification and Mitigation Strategies P25: Incident Response Plans and Post-Incident Analysis P26: Simulation Drills P27: Third-Party Risk Management P28: Secure Supply Chain and Contractual Security
C8: Continuous Improvement, Monitoring and Testing	P29: Performance Metrics and Process Optimization P30: Innovation Encouragement and Compliance Programs P31: Vulnerability Scanning and Penetration Testing P32: Security and Regular Audits P33: Documentation
C9: Data Protection and Privacy, Access Control and Least Privilege	P34: Role-Based Access Control (RBAC) P35: Least Privilege Principle P36: Multi-Factor Authentication (MFA) P37: Encryption and Data Minimization P38: Privacy by Design

This section presents an analysis of the results obtained from the fuzzy AHP. The FAHP steps were meticulously executed to ascertain HSF and HSV practice weights for SSD projects within and across categories. Subsequent subsections detail the complete sequence of FAHP steps undertaken, along with their corresponding implications:

1). STEP 1: Sorting out a Problem's Components (HSFs and HSVs Practices) Using a Hierarchical Framework

Motivated by existing literature [53,67,73,74], we constructed a hierarchical structure to address progressively complex decision-making problems. This hierarchy consisted of three levels, as depicted in Figure 8.

At Level 1, the primary issue (prioritization of HSFs and HSVs for SSD projects) was outlined. Level 2 encompassed the categories of practices, while Level 3 detailed their respective practices, as illustrated in Figure 8.

2). STEP 2: Comparing Matrix Pairs

In Step 2, within the FAHP analysis, our objective was to rank the identified HSF and HSV practices based on their significance within SSD projects. This required a pair-wise matrix comparison and expert input to prioritize practices and categories.

We expanded our first survey with a Fuzzy-AHP version, but only 25 of 70 people were invited to take part willing to do so. Rigorous checks were applied to ensure consistency and completeness of the 20 complete responses obtained from healthcare security experts for the pair-wise matrix comparison. While the sample size of 20 might seem limited, it aligns with comparable studies [53,67,73–75] and allows for reasonable generalization.

Figure 9 presents the demographic details of the 20 experts for pair-wise matrix comparison. A detailed description of how experts were selected, could include:

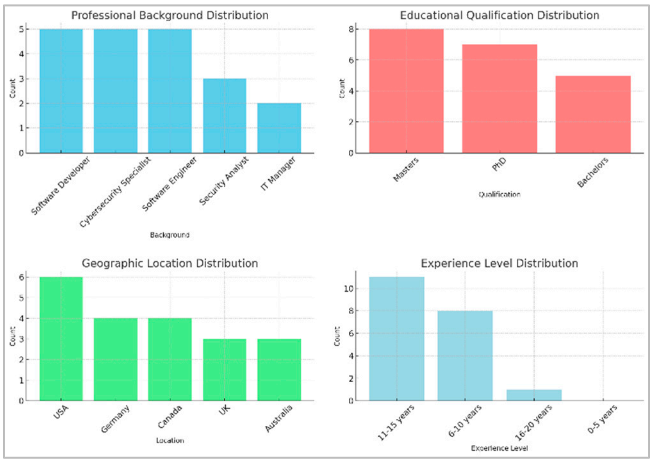


Figure 9. Demographic details of 20 Experts for Pair-wise Matrix Comparison.

- Experts were chosen because of their extensive background in secure software development, cybersecurity, and decision-making frameworks. Their knowledge was directly applicable to the study’s goal of increasing security through people's maturity.
- Experts needed to have a minimum of [X] years of professional experience in secure software development, cybersecurity, or similar fields to guarantee that they had an in-depth knowledge of the issues at hand.
- The sample comprised professionals who were software developers, cybersecurity analysts, and IT managers, all of whom had contributed valuable knowledge from their domains.
- The professionals had high degrees in computer science, cybersecurity, software engineering, and other domains, some of them had security management certificates and project management certificates.
- Experts were picked from a varied array of geographic areas including North America, Europe, and Asia to ensure the framework include it involves a wide variety of security practices and people maturity models.
- The sample of 20 experts was chosen to create a balanced representation of professionals from different sectors and geographical areas to be representative of the relevant expertise for the fuzzy-AHP analysis.
- Although for the expert sample, we strictly selected people to have diversity and relevance of expertise, one needs to bear in mind that, nevertheless, there can exist some biases, especially

associated with procedures typical for a certain industry or peculiarities of secure software development practices in different regions.

The geometric mean formula below-converted expert survey responses to Triangular Fuzzy Numbers (TFN):

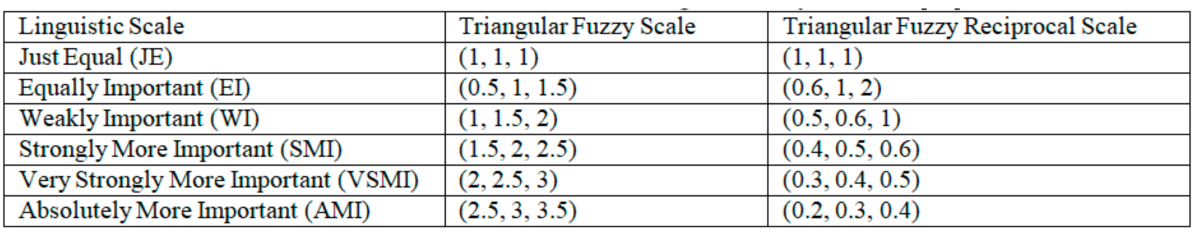
$$\text{Geometric mean} = \sqrt[n]{a1 \times a2 \times a3 \ldots \ldots an}$$

a = weight of each response

n = number of responses

Table 10 shows the fuzzy triangle values used to apply the linguistic variable. As described in [76], the triangular matrix method was used to formulate pair-wise comparison matrices. Table 11 displays the paired matrix comparison of HSF and HSV practice categories for SSD projects.

Table 10. Conversion Scale of Triangular Fuzzy Numbers [76].



Linguistic Scale	Triangular Fuzzy Scale	Triangular Fuzzy Reciprocal Scale
Just Equal (JE)	(1, 1, 1)	(1, 1, 1)
Equally Important (EI)	(0.5, 1, 1.5)	(0.6, 1, 2)
Weakly Important (WI)	(1, 1.5, 2)	(0.5, 0.6, 1)
Strongly More Important (SMI)	(1.5, 2, 2.5)	(0.4, 0.5, 0.6)
Very Strongly More Important (VSMI)	(2, 2.5, 3)	(0.3, 0.4, 0.5)
Absolutely More Important (AMI)	(2.5, 3, 3.5)	(0.2, 0.3, 0.4)

Table 11. Fuzzified Pair-wise Matrix Comparison of the Practices Categories for HSFs and HSVs.

Categories	C-1	C-2	C-3	C-4	C-5	C-6	C-7	C-8	C-9	Sum	Sum * Inverse of the total sum of the columns
C-1	(1, 1, 1)	(0.4, 0.5, 0.6)	(0.5, 1, 1.5)	(0.5, 1, 1.5)	(0.5, 0.6, 1)	(0.5, 1, 1.5)	(0.6, 1, 2)	(0.6, 1, 2)	(0.5, 0.6, 1)	(5.1, 7.7, 12.1)	(0.0392, 0.0893, 0.2008)
C-2	(1.5, 2, 2.5)	(1, 1, 1)	(0.5, 1, 1.5)	(0.4, 0.5, 0.6)	(0.5, 1, 1.5)	(1, 1.5, 2)	(0.5, 0.6, 1)	(0.5, 0.6, 1)	(1, 1.5, 2)	(7.4, 9.7, 13.1)	(0.0569, 0.1125, 0.2174)
C-3	(0.6, 1, 2)	(0.6, 1, 2)	(1, 1, 1)	(1, 1.5, 2)	(1.5, 2, 2.5)	(0.5, 1, 1.5)	(0.5, 1, 1.5)	(1.5, 2, 2.5)	(0.6, 1, 2)	(7.8, 11.5, 17)	(0.0600, 0.1334, 0.2822)
C-4	(0.6, 1, 2)	(1.5, 2, 2.5)	(0.5, 0.6, 1)	(1, 1, 1)	(0.5, 1, 1.5)	(1.5, 2, 2.5)	(0.5, 1, 1.5)	(0.5, 1, 1.5)	(0.5, 1, 1.5)	(7.1, 10.6, 15)	(0.0546, 0.1229, 0.2490)
C-5	(1, 1.5, 2)	(0.6, 1, 2)	(0.4, 0.5, 0.6)	(0.6, 1, 2)	(1, 1, 1)	(0.6, 1, 2)	(0.5, 0.6, 1)	(0.6, 1, 2)	(0.4, 0.5, 0.6)	(5.7, 8.1, 16.2)	(0.0438, 0.0939, 0.2689)
C-6	(0.6, 1, 2)	(0.5, 0.6, 1)	(0.6, 1, 2)	(0.4, 0.5, 0.6)	(0.5, 1, 1.5)	(1, 1, 1)	(1, 1.5, 2)	(1, 1.5, 2)	(1.5, 2, 2.5)	(7.1, 10.1, 14.6)	(0.0546, 0.1171, 0.2423)

C-7	(0.5, 1, 1.5)	(1, 1.5, 2)	(0.6, 1, 2)	(0.6, 1, 2)	(1, 1.5, 2)	(0.5, 0.6, 1)	(1, 1, 1)	(0.5, 0.6, 1)	(0.5, 0.6, 1)	(6.2, 8.8, 13.5)	(0.0477, 0.1020, 0.2241)
C-8	(0.5, 1, 1.5)	(1, 1.5, 2)	(0.4, 0.5, 0.6)	(0.6, 1, 2)	(0.5, 1, 1.5)	(0.5, 0.6, 1)	(1, 1.5, 2)	(1, 1, 1)	(1, 1.5, 2)	(6.5, 9.6, 13.6)	(0.0500, 0.1113, 0.2257)
C-9	(1, 1.5, 2)	(0.5, 0.6, 1)	(0.5, 1, 1.5)	(0.6, 1, 2)	(1.5, 2, 2.5)	(0.4, 0.5, 0.6)	(1, 1.5, 2)	(0.5, 0.6, 1)	(1, 1, 1)	(7, 9.7, 13.6)	(0.0539, 0.1125, 0.2257)
The total sum of the columns										(59.9, 85.8, 128.7)	
The inverse of the total sum of the columns										(0.0077, 0.0116, 0.0166)	

3). STEP 3: Consistency Evaluation

The practices Categories matrix (refer to Table 12) was employed to assess consistency. This table illustrates the Fuzzy Crisp Matrix (FCM) derived from de-fuzzifying practices. It divides categories into crisp numbers through pair-wise matrix comparisons using equation (14).

Table 12. Fuzzy Crisp Matrix of Practices Categories for HSFs and HSVs that Impact SSD Projects.

Categories	C-1	C-2	C-3	C-4	C-5	C-6	C-7	C-8	C-9
C-1	1	0.5	1	1	0.61	1	1	1	0.61
C-2	2	1	1	0.5	1	1.5	0.61	0.61	1.5
C-3	1	1	1	1.5	2	1	1	2	0.61
C-4	1	2	0.61	1	1	2	1	1	1
C-5	1.5	1	0.5	1	1	1	0.61	1	0.5
C-6	1	0.61	1	0.5	1	1	1.5	1.5	2
C-7	1	1.5	1	1	1.5	0.61	1	0.61	0.61
C-8	1	1.5	0.5	1	1	0.61	1.5	1	1.5
C-9	1.5	0.61	1	1	2	0.5	1.5	0.61	1
Sum	11	9.72	7.61	8.5	11.11	9.2	9.72	9.33	9.32

4). STEP 4: Identification of Local Priority Weights of HSFS and HSVS of Practice Categories

a. A Numerical Example

To find the weight vector's normalized values, we used Equation (15). Tables 13 and 14 show the priority weights assigned to each HSF and HSV practice category in the development of the SSD project, and these values are normalized and non-fuzzy.

Table 13. Degree of Possibilities of Practice Categories for HSFs and HSVs.

Categories	Sum * Inverse of the total sum of the columns	C-1	C-2	C-3	C-4	C-5	C-6	C-7	C-8	C-9	Priority Weight

V (C1≥…)	(0.0392, 0.0893, 0.2008) (l1, m1, u1)	-	0.6625	0.5991	0.6359	0.7416	0.6523	0.7091	0.6765	0.6671	0.5991
V (C2≥…)	(0.0569, 0.1125, 0.2174) (l2, m2, u2)	1	-	0.8827	0.9399	1	0.9725	1	1	1	0.8827
V (C3≥…)	(0.0600, 0.1334, 0.2822) (l3, m3, u3)	1	1	-	1	1	1	1	1	1	1
V (C4≥…)	(0.0546, 0.1229, 0.2490) (l4, m4, u4)	1	1	0.9473	-	1	1	1	1	1	0.9473
V (C5≥…)	(0.0438, 0.0939, 0.2689) (l5, m5, u5)	1	0.9193	0.8409	0.8808	-	0.9023	0.9646	0.9263	0.9203	0.8409
V (C6≥…)	(0.0546, 0.1171, 0.2423) (l6, m6, u6)	1	1	0.9179	0.9700	1	-	1	1	1	0.9179
V (C7≥…)	(0.0477, 0.1020, 0.2241) (l7, m7, u7)	1	0.9409	0.8393	0.8902	1	0.9182	-	0.9492	0.9418	0.8393
V (C8≥…)	(0.0500, 0.1113, 0.2257) (l8, m8, u8)	1	0.9929	0.8823	0.9365	1	0.9672	1	-	0.9930	0.8823
V (C9≥…)	(0.0539, 0.1125, 0.2257) (l9, m9, u9)	1	1	0.8879	0.9427	1	0.9738	1	1	-	0.8879

Table 14. Normalized Matrix of Practices Categories for HSFs and HSVs.

Categories	C-1	C-2	C-3	C-4	C-5	C-6	C-7	C-8	C-9	W
C-1	0.09090	0.05144	0.13140	0.11764	0.05490	0.10869	0.10288	0.10718	0.06545	0.09227
C-2	0.18181	0.10288	0.13140	0.05882	0.09000	0.16304	0.06275	0.06538	0.16094	0.11300
C-3	0.09090	0.10288	0.13140	0.05882	0.18001	0.10869	0.10288	0.21436	0.06545	0.11726
C-4	0.09090	0.20576	0.08015	0.11764	0.09000	0.21739	0.10288	0.10718	0.10729	0.12435
C-5	0.13636	0.10288	0.06570	0.11764	0.09000	0.10869	0.06275	0.10718	0.05364	0.09387
C-6	0.09090	0.06275	0.13140	0.05882	0.09000	0.10869	0.06275	0.16077	0.21459	0.10896
C-7	0.09090	0.15432	0.13140	0.11764	0.13501	0.06630	0.10288	0.06538	0.06545	0.10325
C-8	0.09090	0.15432	0.06570	0.11764	0.09000	0.06630	0.15432	0.10718	0.16094	0.11922
C-9	0.13636	0.06275	0.13140	0.11764	0.18001	0.05434	0.15432	0.06538	0.10729	0.11216

$$\lambda_{max} = (11 \times 0.09227) + (9.72 \times 0.11300) + (7.61 \times 0.11726) + (8.5 \times 0.12435) + (11.11 \times 0.09387) + (9.2 \times 0.10896) + (9.72 \times 0.10325) + (9.33 \times 0.11922) + (9.32 \times 0.11216)$$

$$= 1.01431 + 1.09836 + 0.89234 + 1.05697 + 1.04289 + 1.00243 + 1.00359 + 1.11232 + 1.04533$$

$$= 9.26854$$

In Table 2, the random index (RI) is determined as 1.45, considering the presence of 9 elements. Equations (15) and (16) were used to calculate the consistency index (CI) and, by extension, the consistency ratio (CR):

$$\text{CI} = \frac{(9.26854 - 9)}{9-1}$$

$$\text{CI} = \frac{0.26854}{8}$$

$$\text{CI} = 0.03356$$

$$\text{CR} = \frac{CI}{RI}$$

$$\text{CR} = \frac{0.03356}{1.45}$$

$$\text{CR} = 0.02314$$

CR value of 0.02314, which is <0.10

This CR value of 0.02314, being <0.10, indicates the consistency of the pair-wise matrix representing HSF and HSV practice categories.

Tables 15–24 show the results of applying these same procedures to assign weights to each HSF and HSV practice category, its practices, and the fuzzified pair-wise matrix comparison.

Table 15. Comparison of Fuzzified Pair-wise Matrix for C-1: Security Training and Awareness.

C-1	P-1	P-2	P-3	P-4	Weight
P-1	(1, 1, 1)	(0.5, 1, 1.5)	(0.5, 0.6, 1)	(1, 1.5, 2)	0.2478
P-2	(0.6, 1, 2)	(1, 1, 1)	(0.5, 1, 1.5)	(0.5, 1, 1.5)	0.2447
P-3	(1, 1.5, 2)	(0.6, 1, 2)	(1, 1, 1)	(0.5, 0.6, 1)	0.2531
P-4	(0.5, 0.6, 1)	(0.6, 1, 2)	(1, 1.5, 2)	(1, 1, 1)	0.2543

Table 16. Comparison of Fuzzified Pair-wise Matrix for C-2: Collaborative Security Culture and Leadership.

C-2	P-5	P-6	P-7	P-8	P-9	Weight
P-5	(1, 1, 1)	(0.5, 0.6, 1)	(2, 2.5, 3)	(0.5, 0.6, 1)	(2.5, 3, 3.5)	0.2727
P-6	(1, 1.5, 2)	(1, 1, 1)	(1, 1.5, 2)	(0.5, 1, 1.5)	(0.4, 0.5, 0.6)	0.1764
P-7	(0.3, 0.4, 0.5)	(0.5, 0.6, 1)	(1, 1, 1)	(2.5, 3, 3.5)	(0.5, 1, 1.5)	0.1985
P-8	(1, 1.5, 2)	(0.6, 1, 2)	(0.2, 0.3, 0.4)	(1, 1, 1)	(0.3, 0.4, 0.5)	0.1134
P-9	(0.2, 0.3, 0.4)	(1.5, 2, 2.5)	(0.6, 1, 2)	(2, 2.5, 3)	(1, 1, 1)	0.2388

Table 17. Comparison of Fuzzified Pair-wise Matrix for C-3: Effective Communication and Collaboration.

C-3	P-10	P-11	P-12	P-13	Weight
P-10	(1, 1, 1)	(0.2, 0.3, 0.4)	(1, 1.5, 2)	(1.5, 2, 2.5)	0.2588
P-11	(2.5, 3, 3.5)	(1, 1, 1)	(0.4, 0.5, 0.6)	(0.5, 1, 1.5)	0.3038
P-12	(0.5, 0.6, 1)	(1.5, 2, 2.5)	(1, 1, 1)	(0.5, 0.6, 1)	0.2219
P-13	(0.4, 0.5, 0.6)	(0.6, 1, 2)	(1, 1.5, 2)	(1, 1, 1)	0.2153

Table 18. Comparison of Fuzzified Pair-wise Matrix for C-4: Skill Development and Stakeholder Engagement.

C-4	P-14	P-15	P-16	Weight
P-14	(1, 1, 1)	(0.4, 0.5, 0.6)	(0.5, 1, 1.5)	0.2554
P-15	(1.5, 2, 2.5)	(1, 1, 1)	(0.6, 1, 2)	0.4142
P-16	(0.6, 1, 2)	(0.5, 1, 1.5)	(1, 1, 1)	0.3302

Table 19. Comparison of Fuzzified Pair-wise Matrix for C-5: Agile and DevSecOps Practices.

C-5	P-17	P-18	P-19	P-20	Weight
P-17	(1, 1, 1)	(1.5, 2, 2.5)	(0.5, 1, 1.5)	(0.5, 1, 1.5)	0.2902
P-18	(0.4, 0.5, 0.6)	(1, 1, 1)	(0.5, 1, 1.5)	(1.5, 2, 2.5)	0.2475
P-19	(0.6, 1, 2)	(0.6, 1, 2)	(1, 1, 1)	(1, 1.5, 2)	0.2707
P-20	(0.6, 1, 2)	(0.4, 0.5, 0.6)	(0.5, 0.6, 1)	(1, 1, 1)	0.1914

Table 20. Comparison of Fuzzified Pair-wise Matrix for C-6: User-Centric Security, Clear Roles and Responsibilities.

C-6	P-21	P-22	P-23	Weight
P-21	(1, 1, 1)	(0.4, 0.5, 0.6)	(1, 1.5, 2)	0.2868
P-22	(1.5, 2, 2.5)	(1, 1, 1)	(0.6, 1, 2)	0.3978
P-23	(0.5, 0.6, 1)	(0.5, 1, 1.5)	(1, 1, 1)	0.3153

Table 21. Comparison of Fuzzified Pair-wise Matrix for C-7: Supply Chain Security and Risk Management.

C-7	P-24	P-25	P-26	P-27	P-28	Weight
P-24	(1, 1, 1)	(0.5, 1, 1.5)	(0.6, 1, 2)	(1, 1.5, 2)	(0.5, 0.6, 1)	0.1964
P-25	(0.6, 1, 2)	(1, 1, 1)	(1.5, 2, 2.5)	(0.4, 0.5, 0.6)	(2, 2.5, 3)	0.2373
P-26	(0.5, 1, 1.5)	(0.4, 0.5, 0.6)	(1, 1, 1)	(0.3, 0.4, 0.5)	(0.5, 1, 1.5)	0.1346
P-27	(0.5, 0.6, 1)	(1.5, 2, 2.5)	(2, 2.5, 3)	(1, 1, 1)	(0.6, 1, 2)	0.2519
P-28	(1, 1.5, 2)	(0.3, 0.4, 0.5)	(0.6, 1, 2)	(0.5, 1, 1.5)	(1, 1, 1)	0.1794

Table 22. Comparison of Fuzzified Pair-wise Matrix for C-8: Continuous Improvement, Monitoring, and Testing.

C-8	P-29	P-30	P-31	P-32	P-33	Weight
P-29	(1, 1, 1)	(1.5, 2, 2.5)	(0.5, 1, 1.5)	(0.5, 1, 1.5)	(1, 1.5, 2)	0.2379
P-30	(0.4, 0.5, 0.6)	(1, 1, 1)	(0.5, 1, 1.5)	(1.5, 2, 2.5)	(0.5, 1, 1.5)	0.1887
P-31	(0.6, 1, 2)	(0.6, 1, 2)	(1, 1, 1)	(1, 1.5, 2)	(1.5, 2, 2.5)	0.2376
P-32	(0.6, 1, 2)	(0.4, 0.5, 0.6)	(0.5, 0.6, 1)	(1, 1, 1)	(0.5, 1, 1.5)	0.1652
P-33	(0.5, 0.6, 1)	(0.6, 1, 2)	(0.4, 0.5, 0.6)	(0.6, 1, 2)	(1, 1, 1)	0.1701

Table 23. Comparison of Fuzzified Pair-wise Matrix for C-9: Data Protection and Privacy, Access Control and Least Privilege.

C-9	P-34	P-35	P-36	P-37	P-38	Weight
P-34	(1, 1, 1)	(2, 2.5, 3)	(0.4, 0.5, 0.6)	(0.6, 1, 2)	(1, 1.5, 2)	0.2308
P-35	(0.3, 0.4, 0.5)	(1, 1, 1)	(2, 2.5, 3)	(1, 1.5, 2)	(0.4, 0.5, 0.6)	0.1767
P-36	(1.5, 2, 2.5)	(0.3, 0.4, 0.5)	(1, 1, 1)	(2.5, 3, 3.5)	(0.5, 1, 1.5)	0.2633
P-37	(0.5, 1, 1.5)	(0.5, 0.6, 1)	(0.2, 0.3, 0.4)	(1, 1, 1)	(1.5, 2, 2.5)	0.1549
P-38	(0.5, 0.6, 1)	(1.5, 2, 2.5)	(0.6, 1, 2)	(0.4, 0.5, 0.6)	(1, 1, 1)	0.1739

Table 24. Fuzzified Local and Global Weights of Practices and Sub-Practices Categories for Addressing HSFs and HSVs that Impact SSD Projects.

Practices Categories for Addressing HSFs and HSVs that Impact SSD Project	**Categories Weights**	**Sub-Practices**	**Local Weight**	**Local Rank**	**Global Weight**	**Global Rank**
C1: Security Training and Awareness	0.09227	P1: Regular Training and Phishing Simulations	0.2478	3	0.022865	24
		P2: Security and Awareness Campaigns	0.2447	4	0.022578	25
		P3: Security Certifications and Support Channels	0.2531	2	0.023354	22
		P4: User-Friendly Security Features	0.2543	1	0.023464	21
C2: Collaborative Security Culture and Leadership	0.11300	P5: Team Collaboration and Cross-Functional Teams	0.2727	2	0.030815	8
		P6: Knowledge Sharing and Executive Support	0.1764	4	0.019933	30
		P7: Leadership Commitment	0.1985	3	0.022431	27
		P8: Reward and Recognition	0.1134	5	0.012814	38
		P9: Clear Vision and Empowerment	0.2388	1	0.026984	14
C3: Effective Communication and Collaboration	0.11726	P10: Open Channels and Regular Meetings	0.2588	2	0.030347	9
		P11: Feedback Mechanisms	0.3038	1	0.035624	4
		P12: Clear Communication Channels	0.2219	3	0.02602	15
		P13: Regular Security Briefings	0.2153	4	0.025246	19
C4: Skill Development and Stakeholder Engagement	**0.12435**	P14: Continuous Learning and Mentorship Programs	0.2554	3	0.031759	6
		P15: Hands-On Practice and Stakeholder Communication	**0.4142**	**1**	**0.051506**	**1**
		P16: Expectation Management and Stakeholder Involvement	0.3302	2	0.04106	3
C5: Agile and DevSecOps Practices	0.09387	P17: Agile Security Integration and DevSecOps	0.2902	1	0.027241	13
		P18: Security Retrospectives	0.2475	3	0.023233	23
		P19: Secure Coding Standards	0.2707	2	0.025411	18

		P20: Code Review, Static and Dynamic Analysis	0.1914	4	0.017967	35
C6: User-Centric Security, Clear Roles and Responsibilities	0.10896	P21: Defined Roles and Role Flexibility	0.2868	3	0.03125	7
		P22: Accountability and Usability Testing	0.3978	1	0.043344	2
		P23: User Involvement and User Feedback	0.3153	2	0.034355	5
C7: Supply Chain Security and Risk Management	0.10325	P24: Risk Identification and Mitigation Strategies	0.1964	3	0.020278	29
		P25: Incident Response Plans and Post-Incident Analysis	0.2373	2	0.024501	20
		P26: Simulation Drills	0.1346	5	0.013897	38
		P27: Third-Party Risk Management	0.2519	1	0.026009	16
		P28: Secure Supply Chain and Contractual Security	0.1794	4	0.018523	34
C8: Continuous Improvement, Monitoring and Testing	0.11922	P29: Performance Metrics and Process Optimization	0.2379	1	0.028362	11
		P30: Innovation Encouragement and Compliance Programs	0.1887	3	0.022497	26
		P31: Vulnerability Scanning and Penetration Testing	0.2376	2	0.028327	12
		P32: Security and Regular Audits	0.1652	5	0.019695	32
		P33: Documentation	0.1701	4	0.020279	28
C9: Data Protection and Privacy, Access Control and Least Privilege	0.11216	P34: Role-Based Access Control (RBAC)	0.2308	2	0.025887	17
		P35: Least Privilege Principle	0.1767	3	0.019819	31
		P36: Multi-Factor Authentication (MFA)	0.2633	1	0.029532	10
		P37: Encryption and Data Minimization	0.1549	5	0.017374	36
		P38: Privacy by Design	0.1739	4	0.019505	33

5). STEP 5: Calculation of Local and Global Ranks of HSF and HSV Practices

The significance of various categories and HSF and HSV practices in SSD projects was assessed through a weighted analysis. Table 24 displays the rankings of these categories and their practices, calculated using local and global weights.

The local ranking results emerge from dividing defuzzified results obtained from pairwise comparison processes by normalization. A criterion weight within the hierarchy equals its normalized value. The rankings demonstrate how each element stands relative to the evaluation of human security vulnerabilities together with success practices.

The global ranking system produces results by applying the weights from higher-level criteria to local ranking scores of each criterion or sub-criterion. The aggregation method allows sub-criteria weights to adjust based on the decision-making process's primary goal such as human security and success practices improvement. Each element's global rank shows its total importance when viewed against the entire decision-making problem.

Elements such as human security vulnerabilities and success practices receive their rankings based on their determined global importance after performing both local and global calculations. Decision-making elements, which obtain the highest global rank, receive priority status for achieving the decision objective (such as enhanced human security and practice implementation in software development).

The fuzzy-AHP method requires the development of a hierarchical problem structure followed by fuzzy pairwise assessments leading to defuzzification calculations to determine local and global ranks for human security vulnerability and success practice prioritization. These rankings serve as an effective combined method for performing subjective evaluations and quantitative assessments of factors, which proved beneficial during complex decision-making situations.

These global weights were derived by multiplying the local weight of a practice by the respective category weights. For instance, the category "C4: Skill Development and Stakeholder Engagement" is ranked highest at rank-1 and possesses the greatest weight of 0.12435, as detailed in Table 24, among the identified HSFs and HSVs practices for SSD projects. Similarly, Table 24 shows the highest global weight, which is 0.051506, and the global ranked (rank-1) HSF and HSV practice is "P15: Hands-On Practice and Stakeholder Communication".

V. Securing Software Development Through People Maturity: A Fuzzy-AHP Decision-Making Framework

To illustrate the impact of each human success factor (HSF) and human security vulnerability (HSV) practice for secure software development (SSD), we have constructed a prioritization-oriented taxonomy depicted in Figure 8. This taxonomy outlines the relative priority of each practice within its category. It compares them across all categories in securing software development through people maturity. Visualizing both local and global impacts of each practice aids practitioners in discerning the most significant practice relevant to their roles and responsibilities. The insights in Figure 8 are intended to guide the software development team maturing in refining and formulating new strategies to execute SSD projects.

This prioritization-oriented taxonomy serves as a focal point, enabling the team to concentrate on critical areas essential for the success of the SSD projects within the software engineering landscape.

In Figure 10, the category "C4: Skill Development and Stakeholder Engagement" is ranked highest at rank-1 and possesses the most significant weight of 0.12435, as detailed in Table 19, among the identified HSFs and HSVs practices for SSD projects. 'Skill Development and Stakeholder Engagement' remain core prerequisites for SSD because they are the primary determinants of overall security and project success [77]. A Skilled team ensures that the personnel involved have the skills and the necessary and updated knowledge of how to integrate good security standards right from the development process.

This learning never stops the organization from checking and eliminating possible risks, using recommended standards, and maintaining relevance with threats. On the other hand, stakeholder engagement specifically refers to embracing every stakeholder during the project implementation, coordinating with the clients, users, or any other interested party. Since people are encouraged to communicate and create synergy, it guarantees that security concerns are clearly defined, valued, and implemented in the process. Stakeholders are involved in offering feedback, contributing to the initiatives regarding the implementation of security policies, and assisting in the direction in which specific objectives of the project should be aligned with those of the organization. Captured jointly, skill enhancement and stakeholder participation ensure that everyone within the software supply chain comprehends security and works towards hardening the output to create adequately secure, dependable, and efficient software solutions.

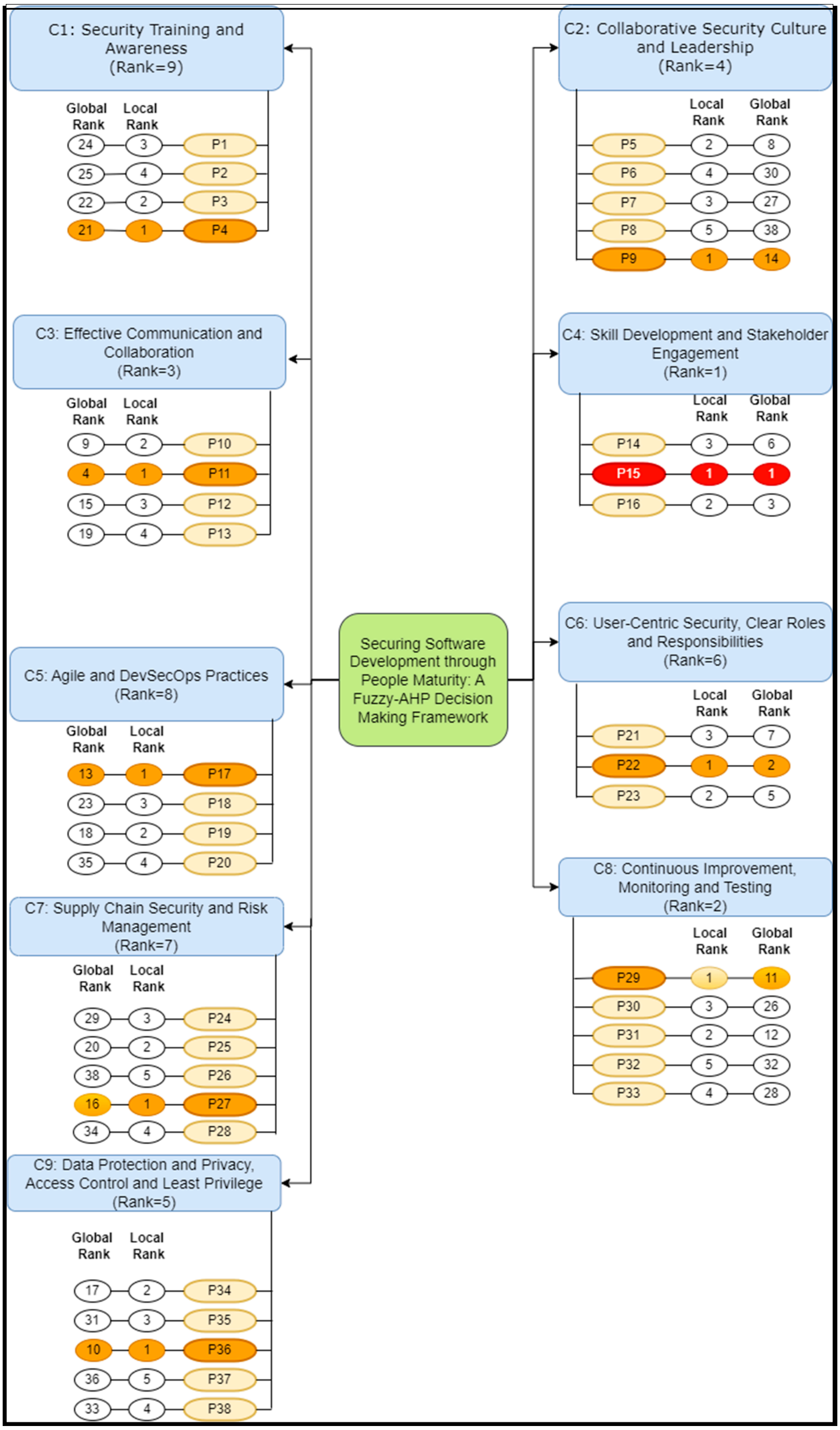

Securing Software Development through People Maturity: A Fuzzy-AHP Decision Making Framework
C1: Security Training and Awareness (Rank=9)
Global Rank
Local Rank
24 3 P1
25 4 P2
22 2 P3
21 1 P4
C2: Collaborative Security Culture and Leadership (Rank=4)
Local Rank
Global Rank
P5 2 8
P6 4 30
P7 3 27
P8 5 38
P9 1 14
C3: Effective Communication and Collaboration (Rank=3)
9 2 P10
4 1 P11
15 3 P12
19 4 P13
C4: Skill Development and Stakeholder Engagement (Rank=1)
P14 3 6
P15 1 1
P16 2 3
C5: Agile and DevSecOps Practices (Rank=8)
13 1 P17
23 3 P18
18 2 P19
35 4 P20
C6: User-Centric Security, Clear Roles and Responsibilities (Rank=6)
P21 3 7
P22 1 2
P23 2 5
C7: Supply Chain Security and Risk Management (Rank=7)
29 3 P24
20 2 P25
38 5 P26
16 1 P27
34 4 P28
C8: Continuous Improvement, Monitoring and Testing (Rank=2)
P29 1 11
P30 3 26
P31 2 12
P32 5 32
P33 4 28
C9: Data Protection and Privacy, Access Control and Least Privilege (Rank=5)
17 2 P34
31 3 P35
10 1 P36
36 5 P37
33 4 P38

Figure 10. Securing Software Development through People Maturity: A Fuzzy-AHP Decision-Making Framework.

Similarly, in Figure 10, the highest global weight is 0.051506, and the global ranked (rank-1) HSF and HSV practice is "P15: Hands-On Practice and Stakeholder Communication" [78]. Action and engagement with the projects' stakeholders can be identified as key components in guaranteeing success and security in software development projects. Gaining experience implies interactivity with the tools, technologies, and methodologies used to develop secure software. A direct interaction of this nature improves their working knowledge and skills, thus making them aware and better placed to avoid security risks. It also helps to develop a culture of minimized stoppages and constant learning, which is especially important in cybersecurity due to its high dynamics.

On the other hand, the stakeholders' communication entails all the involved parties, such as the developers, project managers, clients, and the project's end-users, to be in Compliance with the project's goals, requirements, and limitations. Thus, communication facilitates a clear description of expectations for security, an assessment of the consequences of failed security, and the evaluation of the need for security and the degree to which essential measures are taken. It also encourages timely feedback, decision-making, and teamwork, allowing for the quick identification of new risks and developing ways of handling them. Thus, effectively strengthening practical orientations in developing software solutions and improving communication with key stakeholders can create more secure, intrinsically reliable, and friendly end-user applications.

The taxonomy also presents the variation in the impact of identified HSF and HSV practices concerning their position in their respective categories and overall people's maturity towards SSD projects. For example, the "C1: Security Training and Awareness" category comprises four practices. According to the local ranking in Figure 10, "P4: User-Friendly Security Features" is ranked as the highest priority practice in C-1, but considering the global ranking, it stands at 21st. Similarly, P3 and P1 stand at 2nd and 3rd in C-1, but both stand at 22nd and 24th concerning global ranking. Hence, the SSD team needs to consider both ranking orders (local and global) while developing secure software projects.

VI. Implications of the Study

The findings of the study have several implications for both theoretical and practical work in the sphere of secure software development:

- **Integration between technical controls and human factors:** It meshes with the growth in human maturity, better decision-making, risk awareness, and adherence to security protocols, which could enhance automated security tools, encryption, and other technical measures' adoption and efficiency. For instance, the greater the level of human maturity in security practices, the more likely humans would use technical tools such as automated intrusion detection systems or encryption in their daily activities. The union of a maturing workforce and technical controls ensures a layered defense mechanism that enhances the general security posture. The framework can be stretched to address how technical measures, like real-time tracking or vulnerability scanning, are to be intertwined with human elements, like training and awareness of risk so that security is not only automated but also actively managed by a trained crew.
- **Comprehensive Security Framework:** "Securing Software Development through People Maturity: A Fuzzy-AHP Decision-Making Framework" offers a wider perspective on securing software development by incorporating people maturity and technical security controls. According to the framework, software security should not be dependent on one of the dimensions (human or technical) but rather it has to balance both of them. With the alignment of people maturity and technical security controls, like leveraging the use of automatic tools that need human intervention or training employees on how to make effective use of encryption, organizations will be able to bolster the level of security resilience."
- **Illustrating Synergies between Human and Technical Factors**:

Figure 11 of the Synergies that exist between Human Maturity, and Technical Controls as a way of attaining a Holistic Security. The diagram illustrates the relationship between human factors including Training, Awareness, and Decision-Making with technical components, which include Encryption, Firewalls, and Automated Tools making a combined impact on the enhancement of overall security.

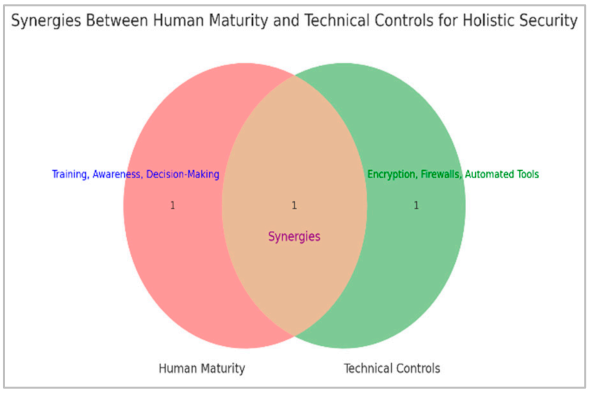

Synergies Between Human Maturity and Technical Controls for Holistic Security
Training, Awareness, Decision-Making
Encryption, Firewalls, Automated Tools
1
1
1
Synergies
Human Maturity
Technical Controls

Figure 11. Synergies between Human and Technical Factors.

- **Enhanced Decision-Making in Security Investments**: The allocation of security resources is therefore facilitated by the Fuzzy-AHP decision-making framework, which gives organizations a quantitative method of evaluating the maturity of their workforce and software about security as well as priority ranking of investment in security throughout the software development life cycle. This method assists firms in getting better value for the resources used by identifying areas that can offer improved security results.
- **Integration of Human Factors in Security Protocols:** The earlier approaches to modeling software security usually come with all the methodologies owning to resources-based technologies and have qualities neglecting the human aspects. This research calls for incorporating people's maturity into security to foster the right approach to secure software development. Thus, it can be described that the general improvement of the security situation in various companies can be achieved with the help of multiple programs, human development, training, and awareness.
- **Improved Risk Management:** By including people's maturity in a list of factors that can affect software security, the framework helps organizations minimize the problems related to the human factor. This takes a different perspective on risk management, where it protects certain technological lapses and people-related issues.
- **Adaptability to Different Organizational Contexts:** Since the Fuzzy-AHP model is created and designed to be scalable and adaptive, it can fit well in different organizational settings and across different organization sizes. This also means that it can be applied across multiple industries depending on the security needs, organizational readiness, and the company's strategic and tactical aspirations and goals. Additionally, this flexibility places the framework in high-essential security areas like finance, healthcare, and government realms.
- **Guidance for Policy and Standards Development**: This study's people maturity and decision-making focus might inform the formation of new standards or policies to advance the secure software industry. This work may be helpful for policymakers and industry groups who wish to adjust security guidance based on the findings here, which could motivate organizations to consider human factors in their software security practices.
- **Contribution to Future Research in Secure Software Development:** The work provides directions for future research by presenting a new method incorporating human maturity, fuzzy logic, and AHP into the process. For the above reasons, future studies can extend this research to other decision-making techniques, develop a more detailed measure of people maturity integration, or test its applicability to other software development paradigms.

- **Practical Applications in Security Training and Development:** This work has shown how one may gain from putting resources into security training and development programs that are customized to suit the maturity of the development teams in an organization. Better training schemes and continuing staff development in organizations are possible to establish higher levels of security awareness and take the necessary actions to produce more secure software.
- **Bridging the Gap Between Security and Development Teams:** The framework will also enhance a closer working relationship between the security and development teams since people's aspects of security risks can now be effectively evaluated and addressed through a clear mechanism. This enhanced communication is essential to prevent the gaps and misinterpretations that are usually associated with the emergence of new security vulnerabilities as new software is developed.

In sum, this research offers practical directives to organizations that seek to protect their software development from adversarial proliferation and tampering and do so considering the workforce maturity level. The proposed Fuzzy-AHP framework intends to be useful for decision-making and improving software security's technical and human aspects.

VII. Summary and Limitations of the Study

This study aims to discern HSFs and HSVs and their practices for SSD projects and formulate a people maturity decision-making framework for secure software development, enhancing software development initiatives. Initially, 24 HSFs and HSVs were identified along with 38 practices to mitigate them through empirical studies involving 70 SSD experts. Additionally, 20 SSD expert panels were convened to analyze the interrelationship between HSF and HSV practices using the FAHP method. Based on collected data, FAHP prioritized these practices, highlighting the most critical category, “C4: Skill Development and Stakeholder Engagement,” which ranked highest at rank-1 and possessed the most significant weight of 0.12435, as detailed in Table 24. Similarly, the highest global weight, 0.051506, and global ranked (rank-1) HSF and HSV practice are “P15: Hands-On Practice and Stakeholder Communication,” as detailed in Table 24.

This study utilized an empirical questionnaire survey to explore people's maturity towards SSD by identifying HSFs and HSVs and their practices, ensuring content validity by aligning the survey with prior SLR findings. Construct validity was confirmed by survey respondents using various scales to assess attribute relevance. Internal validity leveraged SLR results to structure the questionnaire, while external validity incorporated a diverse group of international security experts who participated voluntarily, ensuring varied industry and project backgrounds. The small sample size in the fuzzy-AHP analysis could potentially limit findings due to the interpretive nature of this method.

While this study introduces a Fuzzy-AHP decision-making framework to enhance the security of software development by assessing people's maturity, several limitations should be considered:

- Limited Scope of Participants: While the findings from this study provide valuable insights within the context of secure software development organizations, Cybersecurity, and IT management, their applicability to other sectors or geographic areas may be limited. People's maturity in the study sample was evaluated on a limited and particular number of organizations and professionals. As this may not be a comprehensive list, practices and measures on security and the level of maturity exist in practice across different industries or other geographical locations.
- Subjective Nature of Fuzzy Logic: Despite the effort of the Fuzzy-AHP methods to reduce subjectivity in decision-making, the process of determining fuzzy numbers is still the expert's subjective judgment. Therefore, the measures and the findings depend on the skills and perceptions of the assessors, which may lead to variations in the results.
- Focus on Human Factors: This research mainly deals with the human factors in software development, including skill, knowledge awareness, and organizational culture. It is also an

amalgamation of risk. However, other critical factors include technological tools, process maturity, and infrastructure, which significantly affect software security and are not captured in this framework.

- The Complexity of Implementation: Using the Fuzzy-AHP approach, the decision models are sophisticated, and incorporating software security techniques is quite challenging. As will be seen in the subsequent sections, applying the framework to organizations may be feasible but challenging, particularly for organizations with limited resources or peculiar domain knowledge.
- Dynamic Nature of Software Development: The overall landscape of software development is constantly changing, and there are frequent changes in the approach to security. It may need updating occasionally because the appreciation of some security practices and people's maturity factors may change with time.
- Absence of Longitudinal Analysis: The study gives a cross-section of people's maturity in software security at a particular time. However, it fails to depict the development of people's maturity and security effectiveness. As for the research that should be carried out in the future, it might be useful to evaluate the changes in the future outcomes of long-term software security if maturity is increased. The robustness of this framework can be further strengthened by future studies that involve longitudinal assessments in the determination of effectiveness over time. Such investigations would give us beneficial information about the mannerism in which the framework aligns itself to the new threats of security and shifting attitudes of humans in software development crews.
- Even though the current study offers interesting perspectives on the ability to secure a software development process with people's maturity, the sample used during the empirical validation was constrained by the diversity of industry and coverage geography. This weak spot could make the findings applicable to other sectors or regions with various security requisites and practices. To improve the generalisability of findings, future studies will focus on increasing a broad range of participants, not only industry-wise (healthcare, finance, or manufacturing) but also geographical-wise (different countries, regions) with the different regulatory frameworks and security practices. This extension will give a better understanding of how the framework, Fuzzy-AHP performs in different contexts and make the findings more applicable to a larger population.

These limitations point out the direction for future research and development to improve the comprehensiveness and applicability of the described framework.

VIII. Conclusion and Future Direction

The study advocates integrating security measures for SSD projects, emphasizing the importance of addressing HSFs and HSVs and their practices. This empirical study fills a void by delving into HSFs and HSVs, their specific practices, and their interrelationships, creating a framework via the fuzzy-AHP approach. The taxonomy developed offers valuable insights for academic researchers and practitioners, guiding prioritization and fostering innovation in SSD processes. The proposed framework aids SSD organizations in assessing and enhancing people's maturity for secure software development activities.

In this paper, we introduced a new decision-making approach that contributes to the authorization of development processes of software programs by Fuzzy-AHP that looks at people's maturity. This model combines human and technical aspects as one of the best solutions for measuring and enhancing the security of development personnel. Primarily, the fuzzy-AHP method helped elaborate on people's maturity uncertainties and somewhat ambiguous judgments, increasing the measurement’s reliability. We explained why the people's maturity level strongly impacts software development security using criteria like team competence, organizational culture, leadership, and communication. That is why the proposed framework can become a practical checklist for organizations that want to improve the security of software products by eliminating the risks connected with human factors.

This study also envisions designing a software product resilient to HSFs and HSVs and their practices and establishing testing protocols for such factors. A comparative analysis will be conducted, pitting our framework against other security models/frameworks in a hybrid approach.

- **Future Work**

Generative AI [79,80], a subset of AI that has experienced exponential growth in recent years, on the one hand, opens up the opportunity for secure software development and, on the other, has enormous challenges. Despite its use in developing better codes and improving efficiency, integrating artificial intelligence into safety-driven software development practices is still in its infancy. Several avenues for future exploration arise from this dynamic intersection:

i. **Leveraging AI for Automated Security:** Generative AI has the most valuable application in automating security controls across the SDLC process. Future studies should explore the possibility of using AI to determine the weakness of a model, estimate the risks in the future development of the model, and produce far safer code. However, the issues of ethics and bias in Artificial Intelligence systems also need to be resolved to guarantee that security measures created with AI are ethical.

ii. **Enhancing People Maturity with AI-Driven Insights:** People maturity models successfully target enhancing organizational enablers and a developer’s competency levels. Embedding generative AI could give instructions that advance human factors, such as decision-making, security consciousness, and teamwork. Possibilities of future work can be to investigate how AI powers feedback and training and how AI-based simulations can enhance people's maturity in decision-making, especially regarding security.

iii. **AI-Augmented Threat Intelligence:** In today’s complex cybersecurity environment, AI is most valuable as a fast and proactive defense mechanism. As for future work, further opportunities should be examined regarding integrating AI-generated threat intelligence into secure software development, where the anti-phishing process shall be more adaptive to emerging security threats. This would require building AI systems that evolve with the new threats and patterns being adopted in the market and allow developers to quickly get the most current practices in security.

iv. **Collaboration between AI and Human Developers**: Subsequent studies may consider the interaction of generative AI with human engineers and how to secure this cooperation. Nevertheless, the use of AI as an aid to creating secure code requires human supervision. Research could include environments in which the content is generated based on AI algorithms combined with professional knowledge to satisfy high security.

v. **Addressing AI-Generated Code Vulnerabilities:** An essential direction for future work is to consider AI-generated code as the aspect that introduces specific code weaknesses. Since more developers rely on generative AI to generate code, the explicit security threats with AI-written code must be known. At the same time, future studies should look into how these are assessed, audited, and managed to have a safer result.

vi. **Ethical Considerations in AI for Security:** Using AI in SSD considers many ethical issues while emerging and evolving. The deployment of AI in Secure Software Development, particularly AI software development models, raises numerous ethical questions. Insofar as multiple kinds of ethical frameworks and AI governance policies may be integrated into security-focused software development, there is room for future work to explore what it means to ensure that AI used in the process is not itself unethical – that is, does not reintroduce bias, obscurantism, or other unwanted elements.

These presented areas for future work unveil the shift that generative AI will bring to secure software development. With the development of new AI technologies, understanding how AI interacts with code and with those designing, writing, and implementing it becomes critical in developing secure software.

Acknowledgment: This research is supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R896), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors also acknowledge the contributions of University of Malakand, Pakistan and special thanks are due to King Saud University, Saudi Arabia, Jouf University, Sakka, Saudi Arabia, Almaarefa University, Riyadh, Saudi Arabia, King Abdulaziz University, Jeddah, Saudi Arabia.

References

1. C. Del-Real, E. De Busser, and B. van den Berg, "Shielding software systems: A comparison of security by design and privacy by design based on a systematic literature review," *Computer Law & Security Review,* vol. 52, p. 105933, 2024/04/01/ 2024.
2. M. Jørgensen and E. Escott, "Relative estimates of software development effort: Are they more accurate or less time-consuming to produce than absolute estimates, and to what extent are they person-independent?," *Information and Software Technology,* vol. 143, p. 106782, 2022/03/01/ 2022.
3. R. A. Khan, M. A. Akbar, S. Rafi, A. O. Almagrabi, and M. Alzahrani, "Evaluation of requirement engineering best practices for secure software development in GSD: An ISM analysis," *Journal of Software: Evolution and Process,* vol. 36, p. e2594, 2024.
4. L. Barros, C. Tam, and J. Varajão, "Agile software development projects–Unveiling the human-related critical success factors," *Information and Software Technology,* vol. 170, p. 107432, 2024/06/01/ 2024.
5. A. Jadhav and S. K. Shandilya, "Reliable machine learning models for estimating effective software development efforts: A comparative analysis," *Journal of Engineering Research,* vol. 11, pp. 362-376, 2023/12/01/ 2023.
6. A. Nurwidyantoro, M. Shahin, M. R. V. Chaudron, W. Hussain, R. Shams, H. Perera, *et al.*, "Human values in software development artefacts: A case study on issue discussions in three Android applications," *Information and Software Technology,* vol. 141, p. 106731, 2022/01/01/ 2022.
7. S. Adolph, P. Kruchten, and W. Hall, "Reconciling perspectives: A grounded theory of how people manage the process of software development," *Journal of Systems and Software,* vol. 85, pp. 1269-1286, 2012/06/01/ 2012.
8. V. Anu, K. Z. Sultana, and B. K. Samanthula, "A Human Error Based Approach to Understanding Programmer-Induced Software Vulnerabilities," in *2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 2020, pp. 49-54.
9. J. Pottebaum, J. Rossel, J. Somorovsky, Y. Acar, R. Fahr, P. A. Cabarcos, *et al.*, "Re-Envisioning Industrial Control Systems Security by Considering Human Factors as a Core Element of Defense-in-Depth," in *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2023, pp. 379-385.
10. R. Fujdiak, P. Mlynek, P. Mrnustik, M. Barabas, P. Blazek, F. Borcik, *et al.*, "Managing the Secure Software Development," in *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2019, pp. 1-4.
11. R. M. Fontana, I. M. Fontana, P. A. da Rosa Garbuio, S. Reinehr, and A. Malucelli, "Processes versus people: How should agile software development maturity be defined?," *Journal of Systems and Software,* vol. 97, pp. 140-155, 2014/11/01/ 2014.
12. J. Martins, F. Branco, and H. Mamede, "Combining low-code development with ChatGPT to novel no-code approaches: A focus-group study," *Intelligent Systems with Applications,* vol. 20, p. 200289, 2023/11/01/ 2023.
13. A. Averin and N. Zyulyarkina, "Characteristics of a Random Vector Obtained Using Human-Computer Interaction," in *2020 Global Smart Industry Conference (GloSIC)*, 2020, pp. 271-275.
14. H. Zhanwei, H. Song, H. Bin, and Y. Yi, "Software security testing based on typical SSD:A case study," in *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, 2010, pp. V2-312-V2-316.
15. F. Alnaseef, M. Niazi, S. Mahmood, M. Alshayeb, and I. Ahmad, "Towards a successful secure software acquisition," *Information and Software Technology,* vol. 164, p. 107315, 2023/12/01/ 2023.
16. R. van Solingen, E. Berghout, R. Kusters, and J. Trienekens, "From process improvement to people improvement: enabling learning in software development," *Information and Software Technology,* vol. 42, pp. 965-971, 2000/11/15/ 2000.

17. V. Casola, A. De Benedictis, C. Mazzocca, and V. Orbinato, "Secure software development and testing: A model-based methodology," *Computers & Security,* vol. 137, p. 103639, 2024/02/01/ 2024.
18. R. A. Khan, S. U. Khan, H. U. Khan, and M. Ilyas, "Systematic Literature Review on Security Risks and its Practices in Secure Software Development," *IEEE Access,* vol. 10, pp. 5456-5481, 2022.
19. P. Wang, S. Liu, A. Liu, and W. Jiang, "Detecting security vulnerabilities with vulnerability nets," *Journal of Systems and Software,* vol. 208, p. 111902, 2024/02/01/ 2024.
20. A. Alzahrani and R. A. Khan, "Secure software design evaluation and decision making model for ubiquitous computing: A two-stage ANN-Fuzzy AHP approach," *Computers in Human Behavior,* vol. 153, p. 108109, 2024/04/01/ 2024.
21. J. Harmening, "Chapter 24 - Information Security Essentials for IT Managers: Protecting Mission-Critical Systems," in *Computer and Information Security Handbook (Fourth Edition)*, J. R. Vacca, Ed., ed: Morgan Kaufmann, 2025, pp. 423-432.
22. S. K. Katsikas, "Chapter 35 - Risk Management," in *Computer and Information Security Handbook (Fourth Edition)*, J. R. Vacca, Ed., ed: Morgan Kaufmann, 2025, pp. 583-599.
23. A. S. A. Alghawli and T. Radivilova, "Resilient cloud cluster with DevSecOps security model, automates a data analysis, vulnerability search and risk calculation," *Alexandria Engineering Journal,* vol. 107, pp. 136-149, 2024/11/01/ 2024.
24. R. A. Khan, S. U. Khan, M. A. Akbar, and M. Alzahrani, "Security risks of global software development life cycle: Industry practitioner's perspective," *Journal of Software: Evolution and Process,* vol. 36, p. e2521, 2024.
25. O. O. Olusanya, R. G. Jimoh, S. Misra, and J. B. Awotunde, "A neuro-fuzzy security risk assessment system for software development life cycle," *Heliyon,* vol. 10, p. e33495, 2024/07/15/ 2024.
26. I. Gershfeld and A. Sturm, "Evaluating the effectiveness of a security flaws prevention tool," *Information and Software Technology,* vol. 170, p. 107427, 2024/06/01/ 2024.
27. R. A. Khan, S. U. Khan, M. Ilyas, and M. Y. Idris, "The State of the Art on Secure Software Engineering: A Systematic Mapping Study," presented at the Proceedings of the 24th International Conference on Evaluation and Assessment in Software Engineering, Trondheim, Norway, 2020.
28. R. A. Khan, S. U. Khan, and M. Ilyas, "Exploring Security Procedures in Secure Software Engineering: A Systematic Mapping Study," presented at the Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering, Gothenburg, Sweden, 2022.
29. L. Hofmans, A. van der Stappen, and W. van den Bos, "Developmental structure of digital maturity," *Computers in Human Behavior,* vol. 157, p. 108239, 2024/08/01/ 2024.
30. T. Jukić, I. Pluchinotta, R. Hržica, and S. Vrbek, "Organizational maturity for co-creation: Towards a multi-attribute decision support model for public organizations," *Government Information Quarterly,* vol. 39, p. 101623, 2022/01/01/ 2022.
31. C. Koolen, K. Wuyts, W. Joosen, and P. Valcke, "From insight to compliance: Appropriate technical and organisational security measures through the lens of cybersecurity maturity models," *Computer Law & Security Review,* vol. 52, p. 105914, 2024/04/01/ 2024.
32. M. D. Kadenic, K. Koumaditis, and L. Junker-Jensen, "Mastering scrum with a focus on team maturity and key components of scrum," *Information and Software Technology,* vol. 153, p. 107079, 2023/01/01/ 2023.
33. C. Glasauer, "The Prevent-Model: Human and Organizational Factors Fostering Engineering of Safe and Secure Robotic Systems," *Journal of Systems and Software,* vol. 195, p. 111548, 2023/01/01/ 2023.
34. E. Abad-Segura, A. Infante-Moro, M.-D. González-Zamar, and E. López-Meneses, "Influential factors for a secure perception of accounting management with blockchain technology," *Journal of Open Innovation: Technology, Market, and Complexity,* vol. 10, p. 100264, 2024/06/01/ 2024.
35. B. De Win, R. Scandariato, K. Buyens, J. Grégoire, and W. Joosen, "On the secure software development process: CLASP, SDL and Touchpoints compared," *Information and Software Technology,* vol. 51, pp. 1152-1171, 2009/07/01/ 2009.
36. B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering – A systematic literature review," *Information and Software Technology,* vol. 51, pp. 7-15, 2009/01/01/ 2009.

37. B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," 2007.
38. N. Dissanayake, A. Jayatilaka, M. Zahedi, and M. A. Babar, "Software security patch management - A systematic literature review of challenges, approaches, tools and practices," *Information and Software Technology*, vol. 144, p. 106771, 2022/04/01/ 2022.
39. A. Shukla, B. Katt, L. O. Nweke, P. K. Yeng, and G. K. Weldehawaryat, "System security assurance: A systematic literature review," *Computer Science Review*, vol. 45, p. 100496, 2022/08/01/ 2022.
40. B. Kitchenham, "Procedures for performing systematic reviews," *Keele, UK, Keele University*, vol. 33, pp. 1-26, 2004.
41. T. C. Lethbridge, S. E. Sim, and J. Singer, "Studying Software Engineers: Data Collection Techniques for Software Field Studies," *Empirical Software Engineering*, vol. 10, pp. 311-341, 2005/07/01 2005.
42. J. W. Creswell, *Research design: qualitative, quantitative and mixed methods approaches, 3rd edition*: Sage, London, 2009.
43. S. Wagner, D. M. Fernández, M. Felderer, A. Vetrò, M. Kalinowski, R. Wieringa*, et al.*, "Status Quo in Requirements Engineering: A Theory and a Global Family of Surveys," *ACM Trans. Softw. Eng. Methodol.*, vol. 28, p. Article 9, 2019.
44. M. Humayun, M. Niazi, M. Assiri, and M. Haoues, "Secure Global Software Development: A Practitioners&rsquo; Perspective," *Applied Sciences*, vol. 13, p. 2465, 2023.
45. M. Ilyas, S. U. Khan, H. U. Khan, and N. Rashid, "Software integration model: An assessment tool for global software development vendors," *Journal of Software: Evolution and Process*, vol. n/a, p. e2540, 2023.
46. B. Kitchenham and S. L. Pfleeger, "Principles of survey research part 6: data analysis," *SIGSOFT Softw. Eng. Notes*, vol. 28, pp. 24–27, 2003.
47. M. A. Akbar, A. A. Khan, and S. Rafi, "A systematic decision-making framework for tackling quantum software engineering challenges," *Automated Software Engineering*, vol. 30, p. 22, 2023/07/26 2023.
48. Rafiq A. Khan, I. Keshta, Hussein A. Al Hashimi, Alaa O. Almagrabi, Hathal S. Alwageed, and M. Alzahrani, "A Fuzzy-AHP Decision-Making Framework for Optimizing Software Maintenance and Deployment in Information Security Systems," *Journal of Software: Evolution and Process*, vol. 37, p. e2758, 2025.
49. W. Alhakami, "Enhancing Cybersecurity Competency in the Kingdom of Saudi Arabia: A Fuzzy Decision-Making Approach," *Computers, Materials and Continua*, vol. 79, pp. 3211-3237, 2024/05/15/ 2024.
50. A. Alzahrani and R. A. Khan, "Secure software design evaluation and decision making model for ubiquitous computing: A two-stage ANN-Fuzzy AHP approach," *Computers in Human Behavior*, p. 108109, 2023/12/26/ 2023.
51. L. A. Zadeh, "Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems," *Advances in Fuzzy Systems — Applications and Theory*, vol. 6, pp. 1-840, 1996.
52. F. Marimon, M. Casadesus, and I. Heras-Saizarbitoria, "ISO 9000 and ISO 14000 standards: An international diffusion model," *International Journal of Operations & Production Management*, vol. 26, pp. 141-165, 02/01 2006.
53. M. Ayhan, "A Fuzzy AHP Approach for Supplier Selection Problem: A Case Study in a Gear Motor Company," *International Journal of Managing Value and Supply Chains*, vol. 4, 10/09 2013.
54. I. Chamodrakas, D. Batis, and D. Martakos, "Supplier selection in electronic marketplaces using satisficing and fuzzy AHP," *Expert Systems with Applications*, vol. 37, pp. 490-498, 2010/01/01/ 2010.
55. D.-Y. Chang, "Applications of the extent analysis method on fuzzy AHP," *European Journal of Operational Research*, vol. 95, pp. 649-655, 1996/12/20/ 1996.
56. D. Stelzer and W. Mellis, "Success factors of organizational change in software process improvement," *Software Process: Improvement and Practice*, vol. 4, pp. 227-250, 1998.
57. R. Kaur, T. Klobučar, and D. Gabrijelčič, "Harnessing the power of language models in cybersecurity: A comprehensive review," *International Journal of Information Management Data Insights*, vol. 5, p. 100315, 2025/06/01/ 2025.
58. A. R. Khan, S. Khan, and M. Ilyas, *Exploring Security Procedures in Secure Software Engineering: A Systematic Mapping Study*, 2022.

59. M. Aquino, J. Griffith, T. Vattaparambil, S. Munce, M. Hladunewich, and E. Seto, "Patients' and Providers' Perspectives on and Needs of Telemonitoring to Support Clinical Management and Self-care of People at High Risk for Preeclampsia: Qualitative Study," *JMIR Human Factors*, vol. 9, 2022/01/01/ 2022.
60. M. Peixoto, D. Ferreira, M. Cavalcanti, C. Silva, J. Vilela, J. Araújo, *et al.*, "The perspective of Brazilian software developers on data privacy," *Journal of Systems and Software*, vol. 195, p. 111523, 2023/01/01/ 2023.
61. K. Petrenko, A. Mashatan, and F. Shirazi, "Assessing the quantum-resistant cryptographic agility of routing and switching IT network infrastructure in a large-size financial organization," *Journal of Information Security and Applications*, vol. 46, pp. 151-163, 2019/06/01/ 2019.
62. C. D. Tupper, "9 - Data Organization Practices," in *Data Architecture*, C. D. Tupper, Ed., ed Boston: Morgan Kaufmann, 2011, pp. 175-190.
63. A. Austin, C. Holmgreen, and L. Williams, "A comparison of the efficiency and effectiveness of vulnerability discovery techniques," *Information and Software Technology*, vol. 55, pp. 1279-1288, 2013/07/01/ 2013.
64. M. Tanque and H. J. Foxwell, "Chapter Three - Cyber risks on IoT platforms and zero trust solutions," in *Advances in Computers*. vol. 131, A. R. Hurson, Ed., ed: Elsevier, 2023, pp. 79-148.
65. Z. Almahmoud, P. D. Yoo, E. Damiani, K.-K. R. Choo, and C. Y. Yeun, "Forecasting Cyber Threats and Pertinent Mitigation Technologies," *Technological Forecasting and Social Change*, vol. 210, p. 123836, 2025/01/01/ 2025.
66. R. N. Rajapakse, M. Zahedi, M. A. Babar, and H. Shen, "Challenges and solutions when adopting DevSecOps: A systematic review," *Information and Software Technology*, vol. 141, p. 106700, 2022/01/01/ 2022.
67. R. A. Khan, S. U. Khan, M. A. Akbar, and M. Alzahrani, "Security risks of global software development life cycle: Industry practitioner's perspective," *Journal of Software: Evolution and Process*, vol. 36, pp. 1-34, 2023.
68. C. E. Budde, A. Karinsalo, S. Vidor, J. Salonen, and F. Massacci, "CSEC+ framework assessment dataset: Expert evaluations of cybersecurity skills for job profiles in Europe," *Data in Brief*, vol. 48, p. 109285, 2023/06/01/ 2023.
69. T. Tam, A. Rao, and J. Hall, "The good, the bad and the missing: A Narrative review of cyber-security implications for australian small businesses," *Computers & Security*, vol. 109, p. 102385, 2021/10/01/ 2021.
70. X. Liang, C. Konstantinou, S. Shetty, E. Bandara, and R. Sun, "Decentralizing Cyber Physical Systems for Resilience: An Innovative Case Study from A Cybersecurity Perspective," *Computers & Security*, vol. 124, p. 102953, 2023/01/01/ 2023.
71. N. Medeiros, N. Ivaki, P. Costa, and M. Vieira, "Trustworthiness models to categorize and prioritize code for security improvement," *Journal of Systems and Software*, vol. 198, p. 111621, 2023/04/01/ 2023.
72. J. M. Corbin and A. Strauss, "Grounded theory research: Procedures, canons, and evaluative criteria," *Qualitative sociology*, vol. 13, pp. 3-21, 1990.
73. A. A. Khan, M. Shameem, M. Nadeem, and M. A. Akbar, "Agile trends in Chinese global software development industry: Fuzzy AHP based conceptual mapping," *Applied Soft Computing*, vol. 102, p. 107090, 2021/04/01/ 2021.
74. R. A. Khan, M. Y. Idris, S. U. Khan, M. Ilyas, S. Ali, A. U. Din, *et al.*, "An Evaluation Framework for Communication and Coordination Processes in Offshore Software Development Outsourcing Relationship: Using Fuzzy Methods," *IEEE Access*, vol. 7, pp. 112879-112906, 2019.
75. T. Yaghoobi, "Prioritizing key success factors of software projects using fuzzy AHP," *Journal of Software: Evolution and Process*, vol. 30, p. e1891, 09/11 2017.
76. A. K. Barbara, B. D. Budgen, and O. P. Brereton, "Using mapping studies as the basis for further researchA participant-observer case study," *Information Software Technology*, vol. 53, pp. 638-651, 2011.
77. W. S. Admass, Y. Y. Munaye, and A. A. Diro, "Cyber security: State of the art, challenges and future directions," *Cyber Security and Applications*, vol. 2, p. 100031, 2024/01/01/ 2024.
78. B. Li, Q. Zhou, Y. Cao, and X. Si, "Cognitively reconfigurable mimic-based heterogeneous password recovery system," *Computers & Security*, vol. 116, p. 102667, 2022/05/01/ 2022.
79. A. Ding, G. Li, X. Yi, X. Lin, J. Li, and C. Zhang, "Generative Artificial Intelligence for Software Security Analysis: Fundamentals, Applications, and Challenges," *IEEE Software*, pp. 1-8, 2024.

80. A. Gurtu and D. Lim, "Chapter 101 - Use of Artificial Intelligence (AI) in Cybersecurity," in *Computer and Information Security Handbook (Fourth Edition)*, J. R. Vacca, Ed., ed: Morgan Kaufmann, 2025, pp. 1617-1624.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.