

## Characterizing Software Quality Assurance Practices in Kenya

Anthony W. Wambua<sup>1\*</sup>, Benard M. Maake<sup>2</sup>

<sup>1</sup>Department of Computer Science, Daystar University, PO Box 44400-00100 Nairobi, Kenya.

<sup>2</sup>Department of Computing Sciences, Kisii University, PO Box 408-40200 Kisii, Kenya.

**ABSTRACT** – Given the increased reliance on technology, Software Quality Assurance(SQA) has become a vital area in Software Engineering (SE). SQA practices require training, cost and often take more time than actual code writing. Owing to these requirements, software developers often ignore or partly implement SQA practices, leading to potentially poor quality software development. The goal of the study is to characterise SQA practices of software developers in Kenya. As such, quantitative empirical research was conducted. Seventy-seven (N=77) completed questionnaires were received and analysed to yield the required insights. The analysis of the findings indicates compliance with SQA practices. However, the research unearths concerns such as failure to comply with Software Development Life Cycle (SDLC) models as having the potential to lower the quality of software products. The assessment found that Unit testing was the most common type of software test. Based on the findings and literature, recommendations are made. The need to improve software engineering education and invest in software testing is underscored. The results can be generalised to most developing countries and used by software developers and trainers to identify areas in SQA that need strengthening.

### ARTICLE HISTORY

Received: 3 July 2021

Revised: 1 Feb 2022

Accepted: 17 Feb 2022

### KEYWORDS

*Software Testing*

*Software Quality*

*Software Assurance*

*Software Practices*

*Software Education*

## INTRODUCTION

Digital systems running on software are currently controlling our lives in areas such as financial markets, travel, telecommunication, military and commerce. Because many stakeholders are affected by this software, Software Quality has become a core component in system design and development. Software Quality remains elusive in the face of dynamic complexities and changes from either the ever-evolving frameworks, diverse platforms or clients. Various groups have come up with various strategies of assuring that software has quality, yet it cannot be trusted that these organizations' software quality solutions will match the numerous software quality assurance needs of all organizations.

SQA can be defined as the various strategies and techniques adapted to convey a certain level of confidence in the final (software) product quality, ensuring a degree of repeatability and reliability [1], as well as certifying and confirming the quality of software products [2]. In a broader sense, SQA encompasses Software Quality Control – which assesses the degree to which a software product meets established requirements and Software Testing, on the other hand, which refers to executing programs or applications to discover defects [3]. This paper will focus on Software Testing and how SQA practices are implemented during Software Development Life Cycle (SDLC).

Given the increase in reliance on technology, SQA has become a key area in SE since it is an umbrella activity employed throughout the SDLC. SQA is a technique applied in SE to assure that all software engineering processes, methods, activities and work items comply with well-stated guidelines, hence aiding an organization in achieving high-quality software products at the end of the SDLC [4]. Software failures and defects have cost the United States of America billions of dollars annually, and it should be mentioned that half of these costs are borne by users [5]. Among the consequences to various stakeholders include damaging an organization's reputation, the growth of technical debt, and legal issues that affect all stakeholders' lives [1].

Most software products used in Africa are not built by indigenous companies; instead, most are imported from developed countries outside the continent. For instance, the largest economy in Africa, Nigeria, loses more than \$1 billion to software importation annually, and the primary reason for this scenario is that local companies are not producing quality software products [6].

In Kenya, just like other developing countries in Africa, the following reasons inhibit the implementation of SQA standards; 1) lack of patience from the management; 2) strict deadlines issued by organizations to developing companies; 3) pride and ego of developers; 4) lack of or high cost of tools required to perform quality checks; 5) lack of adequately trained personnel in charge of the software; 6) different opinions of various key personnel; 7) the futility of the process in previous situations, and 8) bureaucracy among many other factors [6].

[7] established that in Kenya, the quality assurance activities that were given prominence included those that ensured the correct functionality of software products such as correctness, usability and reliability; developers give more focus

to the Functional Requires (FRs) to the periphery of Non-Functional Requirements (NFRs) such as security and performance and sustainability [8]. Their study further revealed that software quality assurance was regarded as new functionality that required more attention and that lack of resources was the leading obstacle to its realization. Few studies concerning SQA have been done in Kenya, but the software quality ecosystem in developing nations will be used as a yardstick of benchmarking with what is happening in Kenya.

The rest of this paper is organized as follows: the next section presents all information about the related work of this study, followed by the methodology section that presents the tools and techniques used in this research. The research results are then presented, followed by a section that discusses the results of this research; then, finally we make conclusions in the last section of the paper.

## RELATED WORK

A mixed-method survey conducted in Finland majorly focusing on private companies found that most companies carried out automated testing and, majorly, unit testing [9]. [10] surveyed both Brazil and Uruguay to understand practitioners' perceptions towards the use and importance of Software testing practices. Although this study could not confirm or disconfirm the prevalence of unit testing, it found that unit testing, performance, integration, system, acceptance, regression, and exploratory tests were perceived as more critical and prevalently used.

[11] carried out a study in Pakistan, a developing country, to understand testing practices amongst Software developers. The study revealed that most software testing effort was spent on User acceptance testing. The research further revealed that the prevalent SDLC was Agile.

[6] conducted research in Nigeria to determine whether software developing companies were engaging in software quality planning, whether these companies utilized certified standardized processes and procedures, and finally, the challenges causing these companies not to adopt quality practices. It was established that most organizations did not understand what quality standards were and what they entailed. It noted that the activities associated with quality planning were never undertaken. Most organizations admitted the lack of SQA practices due to various organizational problems.

[12] conducted a study to identify whether there are software testing practises in Cameroon and what obstacles were hindering software testing automation. The survey's findings indicated that many organizations do not ensure that tests are done, save the ones done by the developers, and there is a very small percentage of automated tests in many organizations. Finally, it was discovered that most organizations were against test automation since it was perceived as time-consuming and costly to get the tools, implement, design, and maintain the same.

[13] conducted a study to determine the influence of software quality assurance infrastructure on in-house SQA in large government corporations in Kenya. His findings were that government regulations had a minimal effect on ensuring quality assurance. This was due to a lack of clear local policies to support quality assurance and infrastructure to support the in-house development of quality software.

An investigation to find knowledge gaps between practice and education in Software testing by [14] found negative gaps – indicating deficiencies in Software Testing.

## RESEARCH METHODOLOGY

The research design follows the Goal, Question, Metric approach [15]; the goal is to understand Kenya's software quality assurance practices. Further, the study partly adopted the guide for surveys in software engineering suggested by [16], [17]. Figure 1 illustrates steps followed in the implementation of the present research.

In particular, the study seeks to address the following research questions

RQ1: What are the profiles of software developers in Kenya?

The metrics used in creating profiles of software developers included the following; education levels of the developers and whether they are student developers, freelancers, or part of software development teams employed in formal companies. In this research, student developers are developers currently in school – post-secondary schooling, yet involved in commercial software development. On the other hand, freelancers are professional developers who are not employed by any formal company but work on short contracts/projects, primarily online through [freelancer.com](#)<sup>1</sup> and [upwork.com](#)<sup>2</sup>. The last category of developers work in teams and are employed in formal setups.

RQ2: What Software Development Life Cycles do software developers in Kenya apply?

To establish the developers' knowledge and use of SDLCs, metrics under this question included the specific SDLCs developers used in their projects and the nature of projects developers worked on - whether they were Web Applications, Mobile Applications.

RQ3: What are the most used and the least used testing levels and techniques in Kenya?

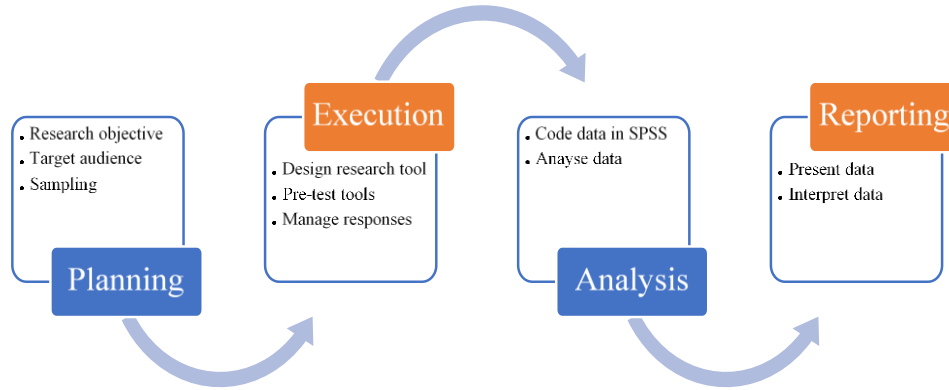
<sup>1</sup> [Hire Freelancers & Find Freelance Jobs Online | Freelancer](#)

<sup>2</sup> [Upwork | The World's Work Marketplace for Freelancing](#)

To address this research question, the metrics included the testing types and techniques developers employed in their projects.

An empirical qualitative study was designed with the metrics converted into a survey questionnaire and administered online through Google forms. Administering the survey online would help the researcher reach the maximum number of respondents and conform to the restriction of movement enforced by the government to control the spread of the novel coronavirus at the time of data collection. The survey tool was pre-tested with a group of conveniently chosen developers, and their suggestions were incorporated into the questionnaire to improve the clarity of the survey questions.

The target population was software developers in Kenya. Convenience and Snowballing sampling techniques were used to identify the survey respondents. The survey link was posted on Facebook groups for Kenyan developers. Researchers further requested developers to share the link with other developers they knew and who might not be in those Facebook groups.



**Figure 1.** Adopted research methodology

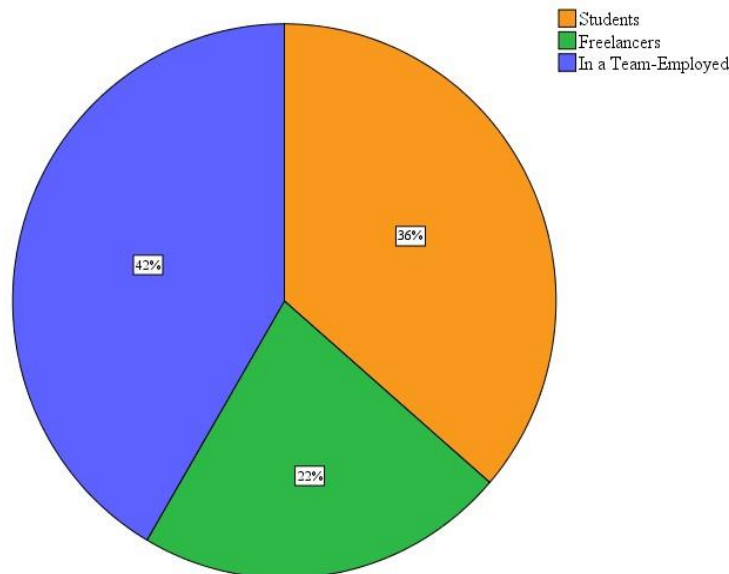
- ✦ The planning stage identifies the research goal, target audience, and sampling method.
- ✦ Execution involves designing and pre-testing the research tools and managing responses.
- ✦ Analysis of data by coding it on the appropriate software.
- ✦ Reporting the study’s finding

## RESULTS

This section presents the results of the survey. Seventy-seven (N=77) completed responses received from the survey were analyzed qualitatively using the SPSS software. Descriptive statistics such as frequencies and tables were used to draw insights that address the research questions.

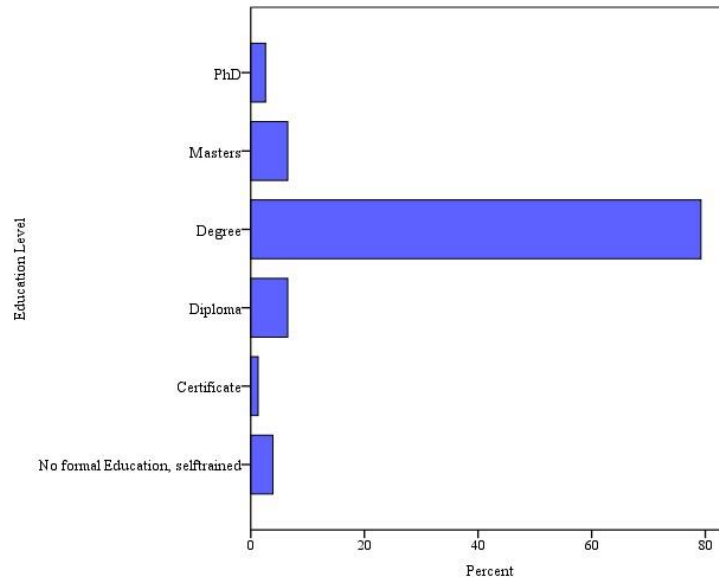
### The research participants profiles (RQ1)

Of the 77 participants, 36% were students. In addition, 22% were freelancers, while 42% were employed in software development companies and majorly worked in professional teams, as shown in Figure 2.



**Figure 2.** Work status of participants

Further, to understand the participant education and training in software development, the survey sought to establish the highest level of training that participants had attained. As shown in Figure 3, this study revealed that the majority (79.2%) of the participants had earned an undergraduate degree. 6.5% were master’s degree holders; a further 6.5% held a diploma, while Ph.D. holders were only 2.6%. The certificate level had 1.3% of the participants. It was fascinating to note that 3.9% had no formal training in software development but had taught themselves.



**Figure 3.** Education level of participants

**Software projects worked on (RQ2)**

The study found that most (28%) of the projects were web applications, followed by backend development at 21.8%. Mobile applications accounted for 15.5%, API development 14.1%, and data science 6.8% of all the handled projects. Distributed computing-related projects were the least handled projects. Table 1 illustrates the projects handled by the survey participants.

**Table 1.** Nature of projects handled by respondents.

Nature of Projects	N	Responses	
		Percent	Percent of Cases
Web Development	63	28.6%	82.9%
Mobile Development	34	15.5%	44.7%
Back end Development	48	21.8%	63.2%
API Development	31	14.1%	40.8%
Data Science Projects	15	6.8%	19.7%
Cloud Computing	7	3.2%	9.2%
Desktop Applications	13	5.9%	17.1%
Embedded Systems Dvt	7	3.2%	9.2%
Distributed Computing	2	.9%	2.6%
<b>Total</b>	<b>220</b>	<b>100.0%</b>	<b>289.5%*</b>

\*Percentages add up to 289.5%, more than 100% since participants could choose more than one option in the multi-response survey question.

A probe into the Software Development Life Cycles (SDLCs) applied by the software developers revealed that most (45.5%) were using Agile Methodologies. 15.6% still use the Waterfall Model. Of threat to Software Quality Assurance (SQA) is the 16.9% of participants who do not use any SDLC. 11.7% are not aware that such methodologies exist, while 5.2% are aware of the existence of the SDLC but do not use them. Figure 4 shows the SDLCs utilization by the survey participants.

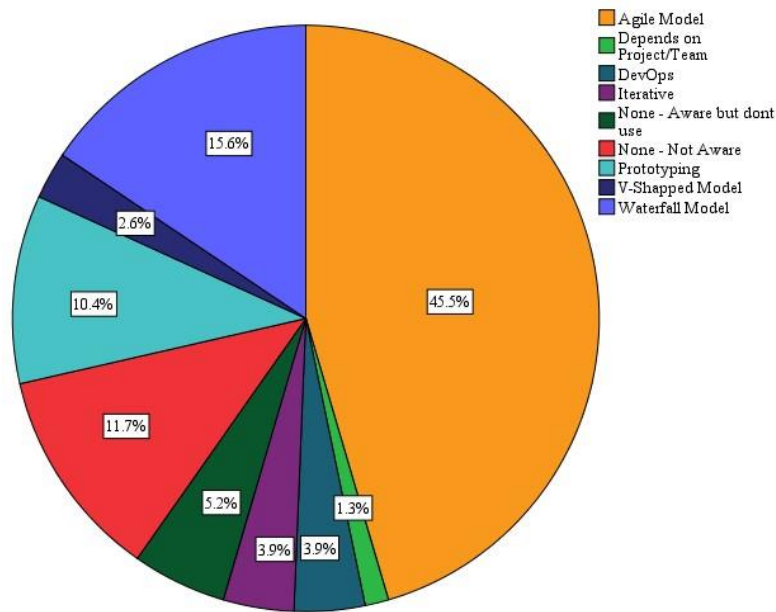


Figure 4. Software Development Life Cycle (SDLC) used by participants

### Software Testing Levels and Techniques (RQ3)

The research sought to establish software testing levels and techniques employed by software developers in Kenya. As depicted in Table 2, the study observed that the most common (20.1%) testing level was Unit testing, followed by System testing 15.1%. The least performed type of testing was Security testing at 8.1%.

Table 2. Levels and Types of testing

	Responses		
	N	Percent	Percent of Cases
Unit Testing	55	20.1%	71.4%
Integrated Testing	33	12.1%	42.9%
System Testing	41	15.0%	53.2%
User Acceptance Testing	38	13.9%	49.4%
Usability Testing	25	9.2%	32.5%
Security Testing	22	8.1%	28.6%
Compatibility Testing	27	9.9%	35.1%
Performance Testing	32	11.7%	41.6%
<b>Total</b>	<b>273</b>	<b>100.0%</b>	<b>354.5%</b>

\*Percentages add up to 354.5%, more than 100% since participants were allowed to choose more than one option in the multi-response survey question.

## DISCUSSION

This research aims to characterize the Software Quality Assurance(SQA) practices amongst Kenyan software developers.

RQ1: Most developers have formal training related to Software Development. 85.7% hold at least an undergraduate degree, those holding a diploma, a certificate or formerly untrained are few (14.3%). In this study [5], weak software developers who do not have a background understanding of SDLCs are identified as a significant problem in improving software quality. The study observes that some developers have a non-computer science background and take short programming courses. The paper notes that 36% of the developers are students. While this is exciting, Higher Education Institutions (HEI) have a crucial role in ensuring student developers produce quality software. Software Engineering Education (SEE) continues to receive the attention of researchers. The focus is to make sure SEE aligns with current

software engineering practices. Even with such attention, research continues to record major gaps in SEE. There is a dire need to introduce pragmatic software testing within CS curriculum, not as an elective but as part of SE course [18]. Authors in [19] surveyed Computer Science senior projects and established they mostly followed the Waterfall model and not the Agile Methodologies widely adopted as the industry standard. Educators and practitioners have to come together to design software engineering curriculums that address industry needs [20]. The project-based learning approach also helps expose learners to a real project development experience.

22% of the developers are freelancers. In Finland, a study [21] showed that software developers turned freelance due to perceived autonomy and fair payment factors. Such a trend is expected to continue and is reflected in our findings. There are implications on Software Quality as formal companies provide expertise and resources for thorough software testing. Companies that employ freelancers ought to have a more robust SQA mechanism since freelancers are more likely not to have software testing teams in place; they focus on the functional aspect of the software.

RQ2: Our study further established that most developers work on Web applications and that the most prevalent SDLCs are the Agile Methodologies. Authors in [22] carried out a survey in North America that agrees with our finding on the prevalence of Agile Methodologies. Web applications can be accessed from different devices, locations, and many users. These benefits explain their prevalence but also the reasons for which such applications need more security.

RQ3: Lastly, the present research shows that Unit Testing is the most common one while Security Testing was the least conducted. We note that emphasis is made on functional tests over non-functional tests. Our findings contradict the findings by [11] conducted in Pakistan and established that the User acceptance test was the most performed. Most projects handled are Web applications, yet the least type of testing conducted is Security Testing raises significant concerns. The ubiquitous nature of Web Applications makes them a prime target for hackers.

### Adoption of Agile Methodologies

Even though the research revealed that most developers had adopted Agile Methodologies, traditional SDLCs such as Waterfall, V-Shaped, and Iterative models were still prevalent. Traditional SDLCs focus on processes and documentation, and the significant drawback is that testing cannot begin until coding is done [23], potentially compromising the quality of the software products. Agile Methodologies models like Extreme Programming (XP), Scrum, Test-Driven Development (TDD), Dynamic System Development Model (DSDM), Feature Driven Development (FDD), and Crystal methods are recommended. Agile Software Development (ASD) methodologies are adaptive and focus on collaboration, allowing high-quality software development in iterations and increments within a specified time limit [24] and prioritizing users [25]. Testing can be done at every iteration, and thus ASD methodologies produce high-quality software SEE should begin to focus on this. Capstone projects for students should apply Agile Methodologies.

### Non-functional software testing

Non-functional testing focuses on Non-Functional Requirements (NFRs) of a software product, while functional testing only focuses on Functional Requirements (FRs). Non-Functional Requirement tests include Usability tests, Performance tests, Compatibility tests, and Security tests. Authors in [26] observed that such tests had been de-emphasized as they were perceived as low risk. Our results confirm that the non-functional tests are the least performed than the functional tests. According to [27], a cultural mindset change on the software developer's side was necessary to prioritize non-functional tests.

## CONCLUSION AND FUTURE WORK

The present research makes the first attempt to characterize Software Quality Assurance (SQA) practices for software developers in Kenya. The study explored the profiles of software developers, their management of software projects concerned with quality assurance, and software test types, levels, and techniques used by Kenyan software developers. The research revealed an invigorating situation where most software developers use SDLC methodology to manage their projects. A trend towards the use of Agile Methodology was also noted. Further, the survey establishes that most developers hold an undergraduate degree relevant to Software Engineering and are employed and part of a software development team. This paper witnesses that developers carry out software tests and that Unit testing is the most common form of testing. Given the critical role of SQA in Software Engineering, the paper makes recommendations on improvement in areas such as full compliance to SDLCs, increased security, and usability testing.

In the future, at-scale research is proposed in Kenya and other developing countries; the study should also look into (1) the test tools used in software testing, (2) the extent of software testing automation, (3) budgets allocated to software quality assurance, (4) time and effort assigned to software testing, (5) whether or not software developers have additional training in software quality assurance (6) the stages in SDLC where software testing is carried out.

## REFERENCES

- [1] N. Walkinshaw, "Software quality assurance," *Springer Int Publ*, vol. 10, pp. 978-983, 2017.
- [2] B. Maqbool, F. U. Rehman, M. Abbas, and S. Rehman, "Implementation of Software Testing Practices in Pakistan's Software Industry," presented at the Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences, Wuhan, China, 2018. [Online]. Available: <https://doi.org/10.1145/3180374.3181340>.
- [3] C. Y. Laporte and A. April, *Software quality assurance*. John Wiley & Sons, 2018.
- [4] M.-C. Lee, "Software quality factors and software quality metrics to enhance software quality assurance," *British Journal of Applied Science & Technology*, vol. 4, no. 21, pp. 3069-3095, 2014.
- [5] M. Hossain, "Challenges Of Software Quality Assurance And Testing," *International Journal of Software Engineering and Computer Systems*, vol. 4, no. 1, pp. 133-144, 2018.
- [6] O. Y. Sowunmi, S. Misra, L. Fernandez-Sanz, B. Crawford, and R. Soto, "An empirical evaluation of software quality assurance practices and challenges in a developing country: a comparison of Nigeria and Turkey," *SpringerPlus*, vol. 5, no. 1, pp. 1-13, 2016.
- [7] E. K. Wachira, "The Practice of Software Quality Assurance by Computer Consulting Firms in Kenya," University of Nairobi, 2003.
- [8] S. Aljarallah and R. Lock, "A Comparison of Software Quality Characteristics and Software Sustainability Characteristics," in *Proceedings of the 2019 3rd International Symposium on Computer Science and Intelligent Control: Association for Computing Machinery*, 2019, p. Article 26.
- [9] T. Hynninen, J. Kasurinen, A. Knutas, and O. Taipale, "Software Testing: Survey of the industry practices," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2018: IEEE, pp. 1449-1454.
- [10] A. C. Dias-Neto, S. Matalonga, M. Solari, G. Robiolo, and G. H. Travassos, "Toward the characterization of software testing practices in South America: looking at Brazil and Uruguay," *Software Quality Journal*, vol. 25, no. 4, pp. 1145-1183, 2017.
- [11] M. S. Jahan, M. T. Riaz, and M. Abbas, "Software testing practices in IT industry of Pakistan," in *Proceedings of the 6th Conference on the Engineering of Computer-Based Systems*, 2019, pp. 1-10.
- [12] T. Maxime Carlos and M. N. Ibrahim, "Practices in software testing in Cameroon challenges and perspectives," *The Electronic Journal of Information Systems in Developing Countries*, vol. 87, no. 3, p. e12165, 2021.
- [13] J. J. Kimuyu, J. O. Ogalo, and S. M. Kiura, "Influence of Organizational Factors on In-House Software Quality Assurance in Strategic State Corporations in Kenya," *International Journal of Scientific Research and Management (IJSRM)*, vol. 5, no. 10, pp. 7238-7246, 2017.
- [14] L. P. Scatalon, M. L. Fioravanti, J. M. Prates, R. E. Garcia, and E. F. Barbosa, "A survey on graduates' curriculum-based knowledge gaps in software testing," in *2018 IEEE Frontiers in Education Conference (FIE)*, 2018: Ieee, pp. 1-8.
- [15] V. R. Basili, "Software modelling and measurement: the Goal/Question/Metric paradigm," 1992.
- [16] J. S. Molléri, K. Petersen, and E. Mendes, "Survey guidelines in software engineering: An annotated review," in *Proceedings of the 10th ACM/IEEE international symposium on empirical software engineering and measurement*, 2016, pp. 1-6.
- [17] K. Ahmadnaumanghazi, S. Reddy, and H. Nekkanti, "Survey Research in Software Engineering: Problems and Mitigation Strategies."
- [18] M. Aniche, F. Hermans, and A. v. Deursen, "Pragmatic Software Testing Education," presented at the Proceedings of the 50th ACM Technical Symposium on Computer Science Education, Minneapolis, MN, USA, 2019. [Online]. Available: <https://doi.org/10.1145/3287324.3287461>.
- [19] K. Saeedi and A. Visvizi, "Software Development Methodologies, HEIs, and the Digital Economy," *Education Sciences*, vol. 11, no. 2, p. 73, 2021.
- [20] O. Cico, L. Jaccheri, A. Nguyen-Duc, and H. Zhang, "Exploring the intersection between the software industry and Software Engineering education - A systematic mapping of Software Engineering Trends," *Journal of Systems and Software*, vol. 172, p. 110736, 2021/02/01/ 2021, doi: <https://doi.org/10.1016/j.jss.2020.110736>.
- [21] S. Frisk, "Motivation behind transition to self-employment: an empirical research of software developers in Finland," 2020.
- [22] H. Assal and S. Chiasson, "'Think secure from the beginning' A Survey with Software Developers," in *Proceedings of the 2019 CHI conference on human factors in computing systems*, 2019, pp. 1-13.
- [23] G. Kumar and P. K. Bhatia, "Comparative analysis of software engineering models from traditional to modern methodologies," in *2014 Fourth International Conference on Advanced Computing & Communication Technologies*, 2014: IEEE, pp. 189-196.
- [24] F. Anwer, S. Aftab, U. Waheed, and S. S. Muhammad, "Agile software development models tdd, fdd, dsdm, and crystal methods: A survey," *International journal of multidisciplinary sciences and engineering*, vol. 8, no. 2, pp. 1-10, 2017.
- [25] M. Amorndettawin and T. Senivongse, "Non-functional Requirement Patterns for Agile Software Development," presented at the Proceedings of the 2019 3rd International Conference on Software and e-Business, Tokyo, Japan, 2019. [Online]. Available: <https://doi.org/10.1145/3374549.3374561>.
- [26] P. Singh and A. K. Tripathi, "Treating nfr as first grade for its testability," 2012.
- [27] C. R. Camacho, S. Marczak, and D. S. Cruzes, "Agile team members perceptions on non-functional testing: influencing factors from an empirical study," in *2016 11th international conference on availability, reliability and security (ARES)*, 2016: IEEE, pp. 582-589.