

Bridging the Gap: Addressing Software Testing Challenges in Namibian Startups through a Tailored Training Approach

Hilma Aludhilu[†]
 School of Computing
 University of Eastern Finland
 Joensuu, Finland
 hilmaalu@uef.fi

Erkki Sutinen
 Department of Computing
 University of Turku
 Turku, Finland
 erkki.sutinen@utu.fi

ABSTRACT

Software testing is an important aspect of the software development process. It assists in ensuring that the software meets the user's requirements and is of high quality. However, testing can be a challenging task, especially for startups. This paper identifies some of the key software testing challenges experienced by 11 Namibian startups. The findings reveal a range of challenges, including a lack of familiarity with testing methods and techniques, unable to use various testing tools proficiently, especially appropriate test automation tools, lack of extensive test coverage of software products, and unsure of when to start testing the software. The challenges lead to poor product quality, which poses a threat to the Namibian software startup ecosystem. As a first step towards addressing the challenges, the study proposes a software testing training approach tailored to the specific requirements of startups. The training program will be devised through a combination of theoretical and empirical approaches. The approach to training will center around an integration of the cognitive and affective domains of Bloom's Taxonomy framework, emphasizing Awareness, Skills, and Attitude, which will guide the needs assessment, design, delivery, and evaluation stages of the training program. Concurrently, empirical data will be gathered through needs assessments, participant feedback, performance metrics, and observations, important in grounding the theoretical concepts in the specific context of the startups and their requirements. The training approach has the potential to assist researchers, educators, and industry professionals in delivering essential software testing training to software developers.

CCS CONCEPTS

- Social and professional topics
- Professional topics
- Computing education
- Informal education



This work licensed under Creative Commons Attribution International 4.0 License.

IWSiB '24, April 16, 2024, Lisbon, Portugal
 © 2024 Copyright is held by the owner/author(s). ACM ISBN 979-8-4007-0571-7/24/04. <https://doi.org/10.1145/3643690.3648244>

KEYWORDS

Software testing, Software testing training, Software startups

ACM Reference format:

Hilma Aludhilu and Erkki Sutinen. 2024. Bridging the Gap: Addressing Software Testing Challenges in Namibian Startups through a Tailored Training Approach. In *Proceedings of 2024 ACM/IEEE International Workshop on Software-intensive Business (IWSiB'24)*. Lisbon, Portugal, <https://doi.org/10.1145/3643690.3648244>

1 Introduction

In the fast-paced and ever-evolving landscape of software development, software startups play a vital role in driving innovation. Software startups are newly created companies designed to create new products and services under extreme conditions of uncertainty [1]. The software startups are mostly product-driven, with a small development team usually led by developers [2]. They often seek to provide clients with products quickly, while operating with limited resources, causing the product quality to suffer. Aludhilu & Sutinen [3] identified that poor-quality software poses a significant threat to the Namibian software startup ecosystem. Poor product quality is an obstacle in the quest for stability and growth [4] and among the problems that harm the quality is the lack of testing [5].

Software testing involves evaluating the software's functionalities to ascertain whether the software complies with the specified requirements [6] and identifies the completeness, correctness, and overall quality of the software [7]. Testing allows quicker product releases, assisting startups in accelerating time-to-market and feature development and enhancing customer satisfaction [8]. Klotins [5] highlights that software testing has a direct impact on software quality, and the product must have an acceptable level of quality in all relevant aspects.

Software developers in startups encounter challenges when implementing effective software testing. These challenges can stem from various factors, including insufficient professional

training upon entering the workforce [9-11]. This study investigates the software testing challenges experienced by 11 Namibian startups. To address the challenges, the study proposes a software testing training approach tailored to the specific demands of Namibian startups. The findings serve as a valuable resource for software startup practitioners, quality assurance professionals, and researchers interested in understanding and improving software testing in startups through training to ensure the delivery of high-quality software.

The remainder of this paper is organized as follows: Section 2 provides the literature review of software testing in startups. Section 3 outlines the study design, describing the data collection process and analysis. Section 4 presents the findings of the research and Section 5 discusses the findings. Finally, Section 6 concludes the paper.

2 Literature Review

This section presents a literature review focusing on software testing in startups, exploring the testing challenges faced by startups and examining how to address the challenges through training.

2.1 Software Testing in Startups

Software testing is an important aspect of software development for startups as it helps to identify and fix defects in the software product, leading to increased market acceptance of products [12]. Testing supports the sustainable evolution of the product and enables faster product releases, thus allowing the teams to reduce time-to-market and to iterate new features faster [8]. Therefore, startups need to pay attention to software testing to ensure that their products are of high quality and meet the needs of their customers.

Startups integrate testing into their software development processes at various stages: some adopt testing techniques right from the start, while others introduce them at later stages. A study conducted in eight software start-ups in southern Brazil [12] found that during the development phase of Minimum Viable Products (MVPs), no testing techniques were used. However, in the second phase, the technical teams employed software testing techniques such as functional tests, pilot clients, specialist testers, and unit tests. A study by Klotins [5] found that some start-ups tend to perform testing when obvious issues emerge, and they rely on user feedback to determine discrepancies instead of testing. Feedback from users is used to spot discrepancies in the product instead of performing rigorous internal testing. The study suggests that software testing practices could be overlooked by startups and that further research is needed to understand the state of practice in software testing in the start-up context [5].

A study [8] with reports from 86 startups identified that manual testing is the leading practice to ensure quality, although there are initiatives that employ automation. Responses in the study suggest that regression testing is an

increasing concern over the startup life cycle, and startups at inception and maturity stages report that they spend substantial effort on manually testing the entire product. Some startups are looking to improve and automate their product testing as good testing practices and test automation can support faster product releases, improve customer satisfaction, and facilitate the onboarding of new developers. It is stated that more test automation, skills regarding software testing, and systematic testing would have been helpful for start-ups. The study also mentions that a dedicated tester role is needed in some start-ups.

2.2 Testing Challenges in Startups

Software startups experience various challenges in conducting effective software testing. Startups often have limited financial resources and a small workforce, making it challenging to allocate sufficient time and personnel for comprehensive testing. These startups do not devote much money and time to testing as resources are limited for them and startups are in a hurry to meet deadlines [13-14]. Also, the skillful use of testing methods and various testing tools is a challenge for software startups because they use simple online tools [13].

A study by Gruner and VanZyl [15] analyzed the software testing process in a small South African IT company, to determine the problems that cause it to deliver software products with too many defects. According to the study, startups face challenges in ensuring software quality due to a rushed or disorganized development process, with the testing step frequently being ignored [15]. Startups operate in a fast-paced environment and aim to deliver products quickly to gain a competitive edge. This urgency leads to testing being rushed or overlooked. Additionally, there is mostly no dedicated test team or testers available, and testing is geared towards quick user-acceptance tests rather than deeper small-scale tests. The study [15] discusses several solutions to the testing problems identified in small IT companies: testing training and education, test planning, and test cases, performing unit testing and test automation, having in-house guidelines; policy and strategy, test timing; early versus late testing, test documentation, test tracking, and recording. Another solution is to establish a dedicated test team or hire testers to perform testing activities and implement more structured testing processes, such as having a dedicated test role in projects. The study also suggests that small IT companies can improve their testing processes by adopting test standards and by using test process improvement models.

A systematic mapping study on 74 primary papers [16] found that software testing is critical to startups' success. However, startups face challenges such as scarce resources and time pressure, as well as a lack of knowledge about customers and users. Testing is costly and time-consuming, hindering startups from testing products required for the development of high-quality software products [16]. Startups need to find an optimal balance between cost or time spent on testing activities

and how this evolves. Also, a study by Unterkalmsteiner et al., [17] found that testing software is costly and often compromised in startups, as it is challenging for startups to fulfill customer needs on time while simultaneously delivering a high-quality product.

2.3 Software Testing Training

There is a need for a highly skilled software testing workforce in the industry to match the constantly rising demand for affordable software testing [9]. Consequently, universities are making significant efforts when it comes to teaching software testing and incorporating practical into their curricula [18], emphasizing testing education and practical [9-10,19-21] to enhance technical capabilities and bridge the gap between academia and industry [22]. Also, various pedagogical approaches, course materials, and testing tools have been proposed for enhancing testing education [9].

Although universities provide the testing foundation for the developers who may join startups, the developers still struggle to perform effective software testing. This is mostly due to inadequate professional training when joining the workforce [9,11]. This shows that in addition to formal software testing education at universities, there is a need for further practical software testing training within the dynamic industry. A study by Hayashiguchi et al., [23] emphasizes the importance of training programs for new graduates entering the IT industry. They highlight the need for training programs that address the skills required for digital transformation and the evolving technology landscape. This underscores the significance of structured training initiatives to bridge the gap between university education and industry skill requirements, ensuring that graduates are equipped with the necessary competencies to thrive in the rapidly changing IT environment.

Training is commonly given through certification programs, self-learning using books or online resources, or industrial training [9]. In terms of industrial training, a study [11] found that internal training programs, external training courses provided by businesses, and self-study are all common. On the other hand, shorter courses provided by universities and higher diploma courses offered by vocational education institutions were less frequent. In addition to university education and industrial training, we advocate for on-the-job learning for developers in startup environments. By providing developers with the opportunity to learn and grow on the job, startups can create a more skilled and engaged workforce. This can lead to increased productivity and innovation. To achieve proficiency in software testing, the developers can derive substantial benefits from an approach that combines university education, industry training, and ongoing on-the-job learning, as illustrated in Figure 1.

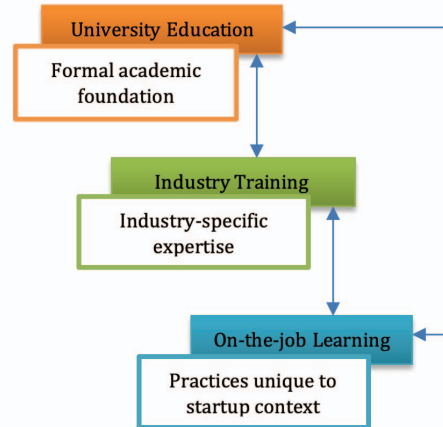


Figure 1: Software education, training, and learning for developers in startups

While software testing education and training have been actively researched and practiced, there is still an ongoing need for improved software testing education and training [9]. This need is emphasized in a study by Garousi [9] where many startup developers admitted that they have not achieved efficient testing due to a lack of proper education and training. Existing research primarily concentrates on software testing education in academia [9-10,19-21], with limited attention to industry training. Hence, this study focuses on software testing training in the industry, focusing on startups. We aim to develop a testing training approach for startups, encompassing a range of testing activities aimed at enhancing testing skills and effectively addressing the immediate software testing requirements and gaps, especially within Namibian startups.

3 Study Design

The objective of this study is to explore software testing challenges experienced by software startups and propose a training approach to address the challenges. The study design, as shown in Figure 2, involves data collection using questionnaires and interviews to identify testing challenges in startups and how they can be addressed. Data collection is followed by data analysis, and based on the analysis, the testing challenges faced by software startups are identified and how to address them. After analyzing the data, we developed a software testing training approach tailored to the findings when it comes to software testing challenges identified in startups.

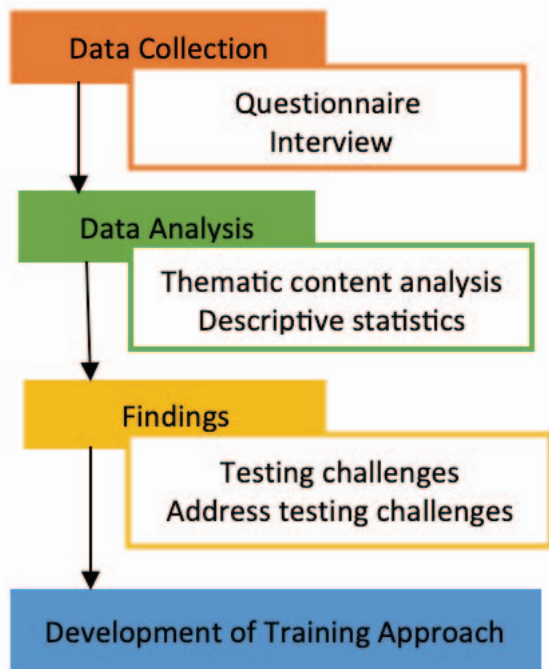


Figure 2: Study Design

3.1 Data Collection

We collected data using an online questionnaire and carried out in-depth interviews with developers in software startups. A questionnaire consisting of single and multiple-response questions was used. The questionnaire consisted of questions regarding software testing in startups and the testing challenges they experience. A summary of the questions asked in the questionnaire is shown in Table 1. Before the questionnaire was made available to the responders, it underwent validation. Two professionals reviewed the questionnaire to ensure it reflected the intended topic of the research and to identify any mistakes, such as repetitive and unclear questions [24]. A preliminary pilot test and further adjustments led to the questionnaire's finalization. The pilot testing consisted of administering the last version of the questionnaire to two (2) intended respondents. Using the snowball sampling method, the questionnaire was sent to two initial participants, and they recruited additional participants. A total of 11 software startups in Namibia completed the questionnaire.

Table 1: Questions asked in the questionnaire

Questions
At what level of development is the testing done in the company? (Select all that apply)
What type of testing is carried out? (Select all that apply and specify others)

Does the company have a dedicated software tester? If not, does it need a dedicated tester?
What software testing techniques are used in the company? (Select all that apply)
Is automated testing done in the company? If not, are you looking to improve and automate your testing?
What are the testing challenges experienced in the company? (Select all that apply and specify others)
How can the startup overcome the challenges it is facing with software testing? (Select all that apply and specify others)

Semi-structured interviews were conducted with 5 software developers from 3 startups. Interviews script was created with open questions to better understand the software testing challenges that software startups experience, and how the challenges can be solved. A summary of the questions asked during the interview is shown in Table 2. The interviews were conducted online using Zoom, each lasting between 30 and 45 minutes.

Table 2: Questions asked during interviews

Question
What are the testing challenges experienced in the company?
<ul style="list-style-type: none"> What contributes to the challenges?
How can the startup overcome the challenges it is facing with software testing?
<ul style="list-style-type: none"> Why do you think that way is suitable?

Table 3 summarizes the characteristics of the startups that participated.

Table 3: Characteristics of companies

	Years operating	Emplo yees	Provided data through
S1	Less than a year	2	Questionnaire
S2	1 year	1	Questionnaire
S3	1 year	4	Questionnaire, Interviewed 2 developers
S4	2 years	3	Questionnaire
S5	3 years	3	Questionnaire, Interviewed 2 developers
S6	4 years	5	Questionnaire
S7	4 years	6	Questionnaire
S8	5 years	5	Questionnaire
S9	5 years	3	Questionnaire
S10	6 years	4	Questionnaire, Interviewed 1 developer
S11	9 years	5	Questionnaire

3.2 Data Analysis

The collected data underwent both quantitative and qualitative analysis. The closed-ended data from the questionnaires were analyzed using descriptive statistics, including percentages. This analysis highlighted any prevailing trends. Interview transcripts were analyzed using thematic content analysis. The analysis commenced with data familiarization which involved transcribing audio and then proceeded with data exploration guided by the following statements: software testing challenges experienced by startups and how the testing challenges can be addressed. Coding was then done where codes are assigned to specific sections of the text to describe the content. As the analysis progressed, the codes were then organized into categories, leading to themes. Concerning challenges, the themes of lack of awareness, resistance, and inadequate resources are identified. These challenges are mapped onto the dimensions of awareness (lack of awareness), attitude (resistance), and skill (inadequate resources). The themes related to strategies comprised comprehensive training programs, leadership commitment, and inclusive policies. These strategies were also mapped onto multiple dimensions of Awareness, Skill, and Attitude.

3.3 Development of Training Approach

In our efforts to develop a software testing training approach, we recognize that effective training involves key stages with distinct focuses. The key stages are training needs assessment, training design, training delivery, and training evaluation. Our approach is based on the fundamental understanding that training should cultivate a comprehensive competence in software testing in addition to imparting knowledge. Therefore, the approach will be structured around three essential pillars: awareness, skills, and attitude. Awareness focuses on the ability to identify and comprehend the various testing methodologies, techniques, and tools available to testers. Skills enable testers to effectively design, execute, and analyze tests, and utilize testing tools and automation frameworks. Attitude cultivates a proactive approach to testing, driving continuous improvement.

4 Findings

In this section, we present the results from the study regarding software testing in 11 Namibian software startups, focusing on the testing challenges they experience and how the challenges can be addressed. Startups perform both functional and non-functional testing. They perform various tests including integration testing, user testing, exploratory testing, scenario-based testing, unit testing, acceptance testing, and usability testing. Overall, 86% of responders indicated that they spend much effort on manually testing products and do not

perform automated testing. However, they are looking to improve and automate their testing. Some startups (43%) only test at the end of the development, while others (29%) test at the start, in the middle, and at the end of the development. Additionally, other startups (29%) conduct testing in the middle and at the end of the development. All studied startups do not have dedicated testers and a majority (57%) indicated that they need dedicated testers, as:

“Quality Assurance is quite integral to the success of any software product” (S8).

Other respondents (43%) indicated that they do not need a dedicated software tester, as

“Various projects require developers with different skills” (S7) and “Depends on the size of the company” (S7).

4.1 Software Testing Challenges

The findings revealed several testing challenges experienced by startups. A majority (86%) of the startups responded that they experience the challenge of a lack of familiarity with testing methods and techniques. Also, they are not able to use various testing tools proficiently, especially appropriate test automation tools. A developer during an interview said,

“It is hard to know which tool to use to automatically test certain products or functions because we are not familiar with them” (S5).

Another developer said:

“We are familiar with automated testing and only a few automated testing tools like Selenium, but we do not know how to use them proficiently” (S10).

Other challenges also include the lack of extensive test coverage of software products (57%) and unsure of when to start testing the software (43%). When it comes to the cost of testing, some startups do not use automated testing due to the cost they believe is associated with automated testing tools. One developer during an interview said:

“Using automated testing could be expensive for startups as sometimes one needs to pay for tools” (S3).
“If we could afford it, we would like to use the tools as it is fast to identify the errors in the software” (S5).

When asked about the causes of testing challenges, respondents (57%) indicated that the lack of professional testers in the start-up, lack of testing expertise, absence of standardized test processes, and lack of investment in the testing team contribute to testing challenges. In addition, frequently changed requirements and misunderstanding of requirements, lack of resources, and lack of time as software testing is time-consuming, all contribute to the challenges experienced by startups. Testing challenges are also caused by

managers paying less attention to testing, and lack of testing training.

Key findings regarding the testing challenges mapped to awareness, skills, and attitude:

Awareness: *Developers lack awareness of various testing methods and techniques, and they are not familiar with a range of testing tools, especially those for test automation.*

Skills: *Developers lack proficiency in using testing tools, especially appropriate test automation tools. There is a need to enhance skills related to testing and test automation.*

Attitudes: *The attitude towards using automated testing tools is influenced by a perception of unaffordability. Some developers resist adopting automated testing tools due to the belief that they are costly.*

4.2 Addressing the Challenges

The study explored potential solutions to the testing challenges experienced by startups. A majority (86%) of the startups responded that startups should engage most of their team members in the testing process. An interview participant said that

“By involving most of their developers in testing, startups could improve their testing as more developers will be involved” (S5).

Also, a majority (86%) responded that startups should invest in good technology and better automation tools to improve their testing process. When it comes to automation tools, the developers need better exposure to automated testing as interviewees highlighted:

“Better exposure to automated testing is needed for developers since the type of testing the developers are exposed to influences the way they test” (S3).
 “Startups should also try to make use of automated testing to improve the testing, even if they use the free versions of some automated testing tools” (S5).
 “Some of the testing tools could be used by startups but this will require getting the developers in the startup to the same speed when it comes to the skills of using the testing tools” (S10).

Table 2: Training stages and objectives related to testing awareness, skills, and attitude.

Training Stage	Testing Awareness	Testing Skills	Attitude towards testing
Training Needs Assessment	Assess existing knowledge gaps.	Identify skill gaps.	Evaluate attitudes towards testing.
Training Design	Design content to address gaps.	Create a skill-focused curriculum.	Develop content to influence a proactive attitude.
Training Delivery	Deliver presentations and exercises to build knowledge.	Provide hands-on training sessions.	Promote a positive attitude towards testing during sessions.
Training Evaluation	Measure improvements in awareness.	Assess skills development.	Evaluate changes in attitude towards testing.

Participants in interviews identified testing training as a key solution to address the testing challenges faced by startups. Developer during the interview said that:

“Developers in startups should start attending available automated software testing training, especially free training for them to familiarize themselves with automated testing” (S3) and “Developers in startups need to be trained and informed more about the importance of automated testing” (S5).

It is also recommended that startups invest in testing resources and developers highlighted that:

“Startups should start having testing policies that guide the testing processes, including what needs to be tested using automated testing and the tools that can be used” (S10).

Key findings from the results on addressing testing challenges mapped to awareness, skills, and attitude:

Awareness: *Developers need to attend training programs, including free opportunities, to increase their awareness about the importance of testing in software development. Training can also provide developers with awareness of different testing methodologies, especially automated testing.*

Skills: *Developers need to attend training programs to get better exposure to testing, especially automated testing to enhance their skills and understanding. This will assist developers in gaining hands-on experience through training, enhancing their practical testing skills.*

Attitudes: *Startups should actively involve team members in testing to instill a proactive attitude toward ensuring product quality. A positive attitude toward embracing automation can be encouraged amongst developers to improve the efficiency and effectiveness of the testing process.*

4.3 Training Approach for Testing

From the results obtained from the study, we propose a software testing training approach to address some of the testing challenges that startups are experiencing. The software testing training approach comprises four key stages: training needs assessment, design, delivery, and evaluation as shown in Table 2. Training should cultivate a comprehensive competence in testing, in addition to imparting knowledge.

Hence, the approach centers around Awareness, Skills, and Attitude, incorporating the cognitive and affective domains of Bloom's Taxonomy (Krathwohl, 2002) into the pedagogical approach, to ensure well-rounded training that equips developers with technical skills and nurtures a positive attitude toward testing. Bloom's Taxonomy is a widely recognized framework in educational psychology, that categorizes learning objectives into cognitive domains (knowledge, comprehension, application, analysis, synthesis, evaluation) and affective domains (receiving, responding, valuing, organizing, characterizing). In this training approach, the cognitive domain will be addressed through the development of Awareness and Skills, encompassing the acquisition of knowledge and the application of practical abilities, respectively. The affective domain will be targeted through the cultivation of a positive attitude, focusing on participants' reception, response, and valuation of the training content. By incorporating both cognitive and affective elements, this approach aims to create a well-rounded learning experience that not only imparts knowledge and skills but also fosters a positive mindset conducive to effective application in real-world scenarios.

5 Discussion

Startups experience various testing challenges, including developers' unfamiliarity with testing techniques and testers lacking proficiency in various testing tools, including automation tools. The challenges align with the existing literature on software testing challenges in startups [15-17]. The results show that the challenge stems from a lack of training, limited resource access, or a shortage of experienced testers familiar with the tools. This highlights a critical gap in the knowledge and skills of startup teams regarding software testing techniques and tools. Therefore, there is a need for training, skill development, or knowledge-sharing within the startups to enhance their testing capabilities. Investing in training and skill development can empower startups to equip their testing teams with a deeper understanding of testing techniques, tools, and best practices. This will allow developers to be able to be actively involved in testing as recommended by developers in the study. There is a perception among some developers in startups that automated testing tools are expensive and that startups cannot afford them. This perception may hinder the adoption of automated testing in startups, which is crucial for improving testing efficiency and ensuring faster product releases. We suggest addressing this perception through training and exposure to affordable or free automated testing tools. This will provide developers with better exposure to testing, especially automated testing, to enhance their skills and understanding.

Possible solutions to the testing challenges are recommended in the study. The recommendation to actively

involve team members in the testing process and invest in better automation tools reflects the broader literature on best practices in software testing [8,12,15]. Engaging the entire team in testing promotes a culture of quality and shared responsibility for product outcomes. Investing in better automation tools aligns with the industry trend toward test automation to improve efficiency and quality [8]. In this study, we further addressed the unawareness, or unfamiliarity, of software developers in startups when it comes to software testing techniques and tools. In acknowledging that effective training transcends mere knowledge transfer, our approach will prioritize effective training by first identifying knowledge gaps and challenges. For instance, challenges faced by startups, such as unfamiliarity with testing techniques and tool proficiency, will be addressed through this approach. Among various training methods like in-house training, online courses, workshops, and seminars [9,11], we recommend hands-on complimentary workshops. The training will introduce developers to current industry testing practices and aims to be focused and shorter, saving startup time. It will also emphasize hands-on skill development to encourage proactive testing and a culture of quality assurance amongst developers in startups.

Regarding the training approach stages that we proposed in Table 2, we have started with the first training stage of needs assessment, using the results of this study as preliminary results. In this study, we assessed some of the existing knowledge gaps, identified skill gaps, and started evaluating the developer's attitudes toward testing. Our next step is to develop concrete training activities to address testing challenges experienced by developers. This will shape the content, ensuring relevance and incorporating it with industry best practices. Also, it will be done considering the nature of the environment the startups in Namibia operate in, the startup's budget, time constraints, and testing requirements. Next, we will refine the approach, select trainers, conduct pilot tests, deliver training, and assess its effectiveness. This shows that the proposed training approach will be tested and refined through practical workshops.

The findings of this study contribute to the understanding of the current landscape of software testing challenges, and how some can be addressed through training, laying the groundwork for further research. To extend this study, a case study will be done to show how the proposed training approach can be applied in practice. Additionally, collaborations between startups and academia can facilitate knowledge exchange. While this study provides valuable insights into software testing in software startups, there are certain limitations to consider. The study is limited by a small sample size of participating software startups in Namibia. These limitations impact the generalizability as the findings do not fully capture the range of testing challenges experienced by the entire software startup community. Also, the findings of this paper may reflect the state of software testing at a specific

point in time, and changes or advancements in the software startup ecosystem after the study might render different findings.

6 Conclusion

In this paper, we presented the results from the study on the software testing challenges experienced by Namibian software startups. The startups experience various challenges, including a lack of familiarity with testing methods and techniques, not being able to use various testing tools proficiently, especially appropriate test automation tools, lack of extensive test coverage of software products, and unsure of when to start testing the software. To address these challenges, the study suggests involving team members in the testing process, investing in better technology and automation tools, and implementing training programs. From the challenges identified, we proposed a software testing training approach, focusing on comprehensive competence, incorporating both cognitive and affective domains to enhance developers' skills and attitudes toward testing. Future work will improve the training approach by identifying and developing concrete activities to address testing challenges experienced by developers. This will shape the content, ensuring relevance and incorporating it with industry best practices.

ACKNOWLEDGMENTS

We would like to thank the developers from the startups who participated in this study.

REFERENCES

- [1] Ries, Eric. 2011. The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses. Crown Business.
- [2] Coleman, Gerry, and Rory V. O'Connor, 2008. An Investigation into Software Development Process Formation in Software Startups. *Journal of Enterprise Information Management* 21, 6 (2008), 633-648.
- [3] Aludhilu, Hilma, and Erkki Sutinen. 2022. Software Startup Ecosystem in Namibia. In *International Conference on Agile Software Development*, Cham: Springer Nature Switzerland, 116-124.
- [4] Adriano Pizzini, Ricardo Bortolo Vieira, Rhodrigo Deda Gomes, Gleison Santos, Andreia Malucelli, and Sheila Reinehr. 2021. Software Quality Practices in Growing Startups: A Qualitative Study. In *XX Brazilian Symposium on Software Quality (SBQS '21)*, November 08–11, 2021, Virtual Event, Brazil. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3493244.3493254>
- [5] Klotins, E., Unterkalmsteiner, M., and Gorschek, T. 2019. Software engineering in startup companies: An analysis of 88 experience reports. *Empirical Software Engineering* 24, (2019) 68–102.
- [6] Gamido, H. V., and Gamido, M. V. 2019. Comparative Review of the Features of Automated Software Testing Tools. *International Journal of Electrical and Computer Engineering* 9, 5 (2019), 4473.
- [7] Taley, D. S., and Pathak, B. 2020. Comprehensive Study of Software Testing Techniques and Strategies: A Review. *International Journal of Engineering Research & Technology (IJERT)* 9, 8 (2020), 817-822.
- [8] Klotins, Eriks; Unterkalmsteiner, Michael; Chatzipetrou, Panagiota; Gorschek, Tony; Prikladnicki, Rafael; Tripathi, Nirnaya; Bento Pompermaier, Leandro. 2019. A Progression Model of Software Engineering Goals, Challenges, and Practice in Startups. *IEEE Transactions on Software Engineering*, (Feb 2019).
- [9] Garousi, Vahid, Austen Rainer, Per Lauvås Jr, and Andrea Arcuri, 2022. Software-testing education: A systematic literature mapping. *Journal of Systems and Software* 165, (2022): 110570.
- [10] Lauvås Jr, Per, and Andrea Arcuri. 2018. Recent Trends in Software Testing Education: A Systematic Literature Review. In *Norsk IKT-konferanse for forskning og utdanning* (Norwegian ICT Conference for Research and Education), 2018.
- [11] Chan, F.T.; Tse, T.H.; Tang, W.H.; Chen, T.Y. 2005. Software Testing Education and Training in Hong Kong. In *Fifth International Conference on Quality Software (QSIC'05)*, IEEE. Melbourne, VIC, Australia, 313-316, <https://doi.org/10.1109/QSIC.2005.57>
- [12] Pompermaier, L., Chanin, R., deSales, A.H.C., Fraga, K., & Prikladnicki, R. 2017. An Empirical Study on Software Engineering and Software Startups: Findings from Cases in an Innovation Ecosystem.
- [13] Ausiku, M.S. Entrepreneurship in the age of technology in Namibia | Business News in Namibia. BiztechAfrica (2018)
- [14] Hossain, M. 2018. Challenges of Software Quality Assurance and Testing. *International Journal of Software Engineering and Computer Systems* 4, 1 (2018), 133-144.
- [15] Gruner Stefan and Van Zyl Johan. 2011. Software Testing in Small IT Companies: A (Not Only) South African Problem. *South African Computer Journal* 47, 1 (2011).
- [16] Berg Vebjørn, Birkeland Jørgen, Nguyen-Duc Anh, Pappas O. Ilias and Jaccheri, Letizia, 2018. Software Startup Engineering: A Systematic Mapping Study. *Journal of Systems and Software* 144 (2018), 255-274.
- [17] Unterkalmsteiner Michael, Abrahamsson Pekka, Wang Xiaofeng, Nguyen-Duc Anh, Shah Syed, Bajwa S. Shahid, Yague, A. 2016. Software Startups-A Research Agenda. *e-Informatica Software Engineering Journal* 10, 1 (2016), 89-124.
- [18] Shi Yaqing and Huang Song. 2019. Research on Software Testing Technical Ability Training based on E-learning. In *Proceedings of the 2019 4th International Conference on Distance Education and Learning*, 30-34.
- [19] Aniche Mauricio, Hermans Felienne, and van Deursen, Arie. 2019. Pragmatic Software Testing Education. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*, Association for Computing Machinery, New York, NY, USA, 414–420. <https://doi.org/10.1145/3287324.3287461>
- [20] Gomes, João, Kennedy Nunes, Luis Rivero, Davi Viana, Geraldo Braz Junior, João Almeida, Vandecia Fernandes, Simara Rocha, and Aristófanes Silva. 2022. Enabling Pragmatic Software Testing Education in Brazil through an Education Repository: An Initial Proposal. In *Proceedings of the 2nd Life Improvement in Quality by Ubiquitous Experiences Workshop*, Aveiro, Brasil. SBC, Porto Alegre, Brasil, 22-29. <https://doi.org/10.5753/lique.2022.19998>
- [21] Andrade Stevão Alves, Neves Vânia de Oliveira, Delamaro Márcio Eduardo. 2019. Software Testing Education: Dreams and Challenges When Bringing Academia and Industry Closer Together. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering (SBES '19)*, Association for Computing Machinery, New York, NY, USA, 47–56. <https://doi.org/10.1145/3350768.3353903>
- [22] Goswami Anurag and Walia Gursimran. 2016. Teaching software requirements inspections to software engineering students through practical training and reflection. In *2016 ASEE Annual Conference & Exposition*. New Orleans, Louisiana
- [23] Hayashiguchi Eiji, Washizaki Hironori, Shintani Katsutoshi, Yoshioka Daisuke. 2022. The Competency-based Computing Curricula 2020 and SFIA V7 Comparison Focusing on Digital Transformation Age. In *2022 IEEE World Engineering Education Conference (EDUNINE)*, IEEE, 1-6.
- [24] Aithal Architha and Aithal P.S. 2020. Development and Validation of Survey Questionnaire & Experimental Data-A Systematical Review-based Statistical Approach. *International Journal of Management, Technology, and Social Sciences (IJMTS)* 5, 2 (2020): 233-251.