Exploring Agile Testing Methodologies: A Perspective from the Software Industry

Mahdia Shoaib[1], Eman Fatima[1], Nosheen Sabahat[2], Nosheen Qamar[1]
[1]Department of Software Engineering, University of Management and Technology, Lahore, 54000, Pakistan
[2]Department of Computer Science, Forman Christian College University, Lahore, 54000, Pakistan
*Correspondence:amjad.telecom2040@gmail.com,bushra.abbas.seemi@gmail.com , saeed.qaisrani@cuivehari.edu.pk,

Citation | Shoaib. M, Fatima. E, Sabahat. N, Qamar. N, "Exploring Agile Testing Methodologies: A Perspective from the Software Industry", IJIST, Vol. 6 Issue. 2 pp 980-995, July 2024
Received | April 19, 2024 **Revised |** June 07, 2024 **Accepted |** June 23, 2024 **Published |** July 01, 2024.

Agile testing is a fast-paced testing method that adheres to the principles outlined in the Agile Manifesto. This research paper explores the adoption of Agile testing methodologies in the context of software houses in Pakistan. The study focuses on identifying the prevalent Agile testing techniques preferred by Software Quality Assurance (SQA) teams and the factors influencing their selection. A survey was conducted to gather insights from professionals in the industry, including SQA experts, developers, and project managers. The findings provided valuable information on the most widely used Agile testing methodologies and the reasons behind their popularity. The core objective of this research is to provide the knowledge related to implemented methodologies, reasons behind the selection of these methodologies, factors that influence the selection of testing tools and techniques, satisfaction level of their selected tools, and how effective their selected tools or techniques are in terms of reducing the number of bugs. The study's contribution lies in offering guidance to software houses in Pakistan by facilitating the adoption of effective Agile testing techniques. The research concludes with recommendations for improving testing practices and enhancing the overall quality of software products in the industry.

Keywords: Software Testing Methodologies; Agile Testing; Software Testing; Software Quality Assurance.

Introduction:

Software testing is an essential part of software development that ensures software systems' quality, dependability, and functionality. This process checks software components, modules, or entire systems for defects, errors, and functionality gaps. The field of software testing encompasses different procedures, strategies, and devices to guarantee that product items satisfy the ideal quality guidelines. Software testing plays a significant role in guaranteeing the quality and efficiency of spry ventures. A quality confirmation action begins with prerequisites assembling and continues throughout the product item life cycle. The software's overall quality may be negatively impacted by a lack of testing resources. As a result, the testing procedure should receive sufficient attention and resources to guarantee high-quality deliverables [1][2].

Agile testing is a fast-paced testing method that adheres to the principles outlined in the Agile Manifesto [3]. It emphasizes iterative testing (making gradual small updates or changes to a product based on test results and user feedback and testing them against predefined baseline metrics.) that is different from traditional testing. Scholars have defined agile development as a philosophy encompassing project management and software development [3]. Agile testing is influenced by various agile development models and aims to satisfy user requirements while maintaining the speed, adaptability, and timeliness of software releases. Many organizations adopt agile software development practices to meet quality standards [2]. Agile methodologies employ smart techniques that aim to address changes in requirements by improving processes related to requirements, planning, execution, and testing, with a focus on acknowledging and accommodating changes [4][5]. Additionally, agile strategies emphasize continuous and direct communication with clients and the importance of thorough documentation.

Testing activities are now integrated into the development process rather than treated as a separate phase in agile testing. Skillful groups work intently together, including designers, analyzers, and partners, to guarantee that product is entirely tried and meets the changing conditions and assumptions for the clients. Agile testing strategies, including Test-Driven Improvement (TDD), Behavior Driven Development (BDD), Acceptance Test-Driven Advancement (ATDD), Exploratory testing, and Session-based testing [6] provide frameworks and guidelines to carry out viable testing practices in agile development. The adoption of agile testing methodologies allows for closer collaboration between testers, developers, and stakeholders, fostering effective communication and shared responsibility for software quality. It also promotes a culture of continuous improvement, as teams regularly reflect on their processes and make necessary adjustments. By leveraging test automation frameworks and embracing the iterative nature of sprints, organizations can achieve faster release cycles and rapid feedback from customers. Despite challenges and differences from traditional testing approaches (as agile testing is integrated into the development process and testing is performed concurrently with development in short iterations whereas traditional testing follows a sequential approach where testing is usually done after the development phase is completed), agile testing proves to be a valuable approach in the pursuit of delivering high-quality software products in a dynamic and customer-focused environment.

Novelty Statement:

There are a lot of studies discussing the agile testing approaches but to the best of our knowledge, we are unable to find any research which discusses the most widely used agile testing methodologies, tools, and approaches used in the Pakistan software industry. To fill this gap, we have conducted a survey of Pakistani software houses and explored the specific testing techniques adopted by software houses and gain a deeper understanding of their reasoning and decision-making processes.

Objectives:

Both industry and academia face challenges regarding the selection of testing processes, tools, and techniques in agile development as it's an integrated activity along with the

development process. The main objective of this study is to provide the knowledge/facts related to implemented methodologies, reasons behind the selection of these methodologies, factors that influence the selection of testing tools and techniques, satisfaction level of their selected tools, and how effective their selected tools or techniques are in terms of reducing the number of bugs. Based on these insights, the decision-making would be easy for professionals regarding the selection of software testing tools in the context of agile development.

The rest of the paper is structured as follows: the very next section (Section 2) describes the related work in this research area whereas Section 3 provides an overview of our research methodologies, utilizing selected papers to offer a comprehensive understanding of the topic. In Section 4, the results of the research are discussed. Finally, Section 5 concludes the paper along with some future research directions.

Related Work:

The Agile methodology has gained popularity in recent years due to its flexibility and collaborative approach. Agile techniques have proven to be highly beneficial, and are transforming the software industry. The term "agile" typically refers to increased adaptability, teamwork, and simplicity in software development. "Agile testing" refers specifically to the agile development workflow's bug and error detection and resolution [1]. This approach adheres to the principles of agile software development and is an iterative, collaborative method. It involves all cross-functional agile team members, including the testing team, to guarantee the consistent delivery of customer-requested business requirements at a sustainable pace. Unlike traditional testing methods that occur in a separate phase, agile testing is integrated with development from the project's inception. The achievement of high-quality products and customer satisfaction is the ultimate goal of Agile development and testing. The most common methodologies in testing practice are Test-Driven development (TDD), Behavior-Driven Development (BDD), Acceptance-Test-Driven Development (ATDD), Exploratory Testing (ET), and Session-Based Testing (ST) [6].

The authors in [6][7] discuss the details of Agile methodology. In Test-Driven Development (TDD), the testing phase precedes the coding phase, starting with writing test cases before coding. This approach ensures that the testing process begins early in the development cycle, promoting a focus on functionality and requirements [1]. BDD works on higher-level tests based on user behaviors and requirements. It involves collaboration between product owners, developers, and testers, using automated tests to ensure completeness and code quality, while also allowing business owners to write tests. Unlike traditional Waterfall testing, BDD integrates testing throughout development, reducing the miscommunication between stakeholders. Behavior-Driven Development (BDD) has the potential to enhance software development in large-scale contexts [8].

ATDD (Acceptance Test-Driven Development) emphasizes team collaboration and involves creating tests by the client, developer, and tester. It offers innovative solutions to long-standing software engineering challenges and excels at identifying basic bugs. Each successful test builds on the software's functionality, facilitated by ATDD's tools that maintain comprehensive functional software documentation throughout development [9]. By continuously testing and refactoring, ATDD ensures that the evolving system consistently meets its requirements and remains robust [10]. It focuses on user perception and uses customer input to develop acceptance criteria and tests, ensuring the code meets those criteria. ATDD directly connects with users to understand product usage, aiming to minimize feature redesign in future releases [6][7]. Exploratory testing (ET) introduced by Kaner [11], is a flexible approach widely used in agile models, focusing on discovery, investigation, and learning. Testers have ownership and freedom to test the software in a well-organized yet chaotic manner, mimicking real-life user interactions [6][7]. ET is an approach to test software without generating pre-design test cases [11][12]. Session-based testing combines exploratory testing with structure and accountability to

quickly discover defects and enable creative on-the-fly test design. Testers conduct tests during uninterrupted sessions, reporting the tests conducted, and uncovering hidden bugs and defects in the software [6][7]. Each session comprises of few functionalities [11], with variable test duration, ranging from a few hours to a maximum of twelve hours as suggested by Itkonen [13].

Qamar and Malik [14] studied the impact of pair testing on software quality and team productivity. Another study [15] compares three methods TDD, BDD, and ATDD, while TTD has a very high coverage which means a large percentage of code is tested automatically [8]. In the study [15], authors developed a tool that automates the generation of Behavior-Driven Model test cases for web-based applications, transforming BDD scenarios into deception test cases (Codeception is a comprehensive and feature-rich testing framework specifically designed for PHP web-based applications), and evaluating the tool using questionnaires and user feedback. Some Behavior-driven development (BDD) and Acceptance test-driven development (ATDD) tools include Cucumber, Concordoin, Jbehave, FitNesse, and SpecFlow [8]. These tools are designed to facilitate the implementation of BDD practices in software development projects as illustrated in Table 1.

Table 1: Behavior-driven development (BDD) and Acceptance test-driven development (ATDD) tools.

Tools	Description
Cucumber	Cucumber is a BDD and -supporting framework and testing tool. It portrays the activities of an application utilizing basic. English language text utilizing a straightforward syntax characterized by a language called Gherkin. Cucumber is written in Ruby and it can test codes that are written in Ruby as well as in Python, Java, C#, and so forth.
Jbehave	Jbehave consists of two parts: • Jbehave Web • Jbehave Fundamental Jbehave can be utilized with JAVA IDEs like Netbeans, Shroud, BlueJ, IntelliJ Thought, and so forth. Using this tool, scenarios are based on text. Jbehave has the element of announcing and reports can be created in various arrangements like XML, HTML, or straightforward text.
Fitness	FitNesse is an Open-Source device written in JAVA. FitNesse automates acceptance testing by integrating with the business. There are two kinds of test systems in FitNesse: • FIT: FIT no longer applies. • SLIM: The lighter version of FIT is called SLIM.
Rspec	A well-liked Ruby testing instrument for behavior-driven development (BDD) is RSpec. It has a rich DSL and is used a lot in production applications because it makes writing tests easy and accessible.
Spec Flow	Cucumber was the inspiration for the open-source tool SpecFlow. Like Cucumber, SpecFlow additionally involves Gherkin language for composing situations. SpecFlow is a tool that runs on.net; it moreover coordinates with Visual Studio. Be that as it may, SpecFlow can run from the Order line also. SpecFlow is also referred to as Cucumber for.Net due to its similarity to Cucumber.
Concordia	Concordia is an exceptionally useful asset written in Java for composing and dealing with the acknowledgment test computerization Contents for Java-based projects. Concordia is compatible with Java IDEs like NetBeans, Eclipse, BlueJ, IntelliJ IDEA, and others due to its integration with the JUnit

	framework. Specifications can be written with Concordoin's assistance, but it does so in HTML.

The most popular tools and frameworks used in Test Driven Development (TDD) are Junit, TestNG, Rspec, NUnit, and CS Unit [16]. Descriptions of the above-mentioned tools are listed in Table 2.

Table 2: Test-Driven Development Tools.

Tools	Description
Junit	Java programmers use the open-source framework JUnit for unit testing. It supports the re-execution of tests when new code is added and allows automated testing. JUnit is suitable for test-driven development because it provides graphical feedback, with green representing successful tests and red representing failures.
N Unit	For both front- and back-end testing, N Unit is a.NET unit testing framework that is open-source. It lets you run tests in a variety of environments, supports data-driven testing well, and lets you group tests into suites. It likewise offers similarity with various stages and permits the production of counterfeit articles.
Test NG	JUnit and N Unit serve as models for the testing framework known as TestNG. To make it stronger and more appealing to software testers, it offers additional features and functions. All types of testing, including functional and unit testing, are supported by TestNG.
CS Unit	CS Unit is an open-source unit testing instrument for the .NET System that upholds different .NET viable dialects like C#, C++, J#, and Visual Fundamental. It allows testing of both front-end and back-end components and incorporates x Unit concepts. CS Unit gives fixes to bunch highlights, empowers manual making of counterfeit items, and offers the capacity to bunch tests into test suites utilizing ns Unit.

Exploratory testing is all about flexibility and adaptability, so the best tools are often those that empower testers to explore systems intuitively. Session-based testing is the method of exploratory testing. Some of the popular tools for exploratory testing and session-based testing include Mind Meister, OBS Studio, Jira, Bugzilla, etc.

Table 3: Exploratory Testing and Session-Based Testing Tools.

Tools	Description
Mind Meister	Mind Meister is a knowledge-based mind-mapping tool that can be used for exploratory testing. This helps in organizing and visualizing the test ideas. Testers can brainstorm, map out test coverage, and dynamically adjust their testing strategy as they explore the software.
OBS Studio	This enables testers or quality assurance person to record their testing sessions. Recording sessions can help capture unexpected behaviors, reproduce bugs, and communicate findings effectively to developers and stakeholders.
Jira	These tools provide the central hubs for logging and tracking bugs discovered during exploratory testing. These tools facilitate collaboration among testers, developers, and stakeholders, ensuring that identified issues are effectively addressed and resolved.
Bugzilla	

Research Methodology:

In this section, the authors provided an overview of the step-by-step process followed to conduct this research, as depicted in Figure 1. Firstly, we discussed the identification of research questions that guided our study. Secondly, we defined our target respondents, specifying the specific individuals or groups we aimed to survey. Next, we provided the details of the design of the questionnaire, ensuring that it effectively captures the necessary information.

Additionally, we highlighted the top respondents, focusing on the key demographics and roles represented in our survey. Following this, we outlined the process of collecting responses from participants. Lastly, we emphasized the importance of analyzing the collected data to derive meaningful insights and draw conclusions from our research.

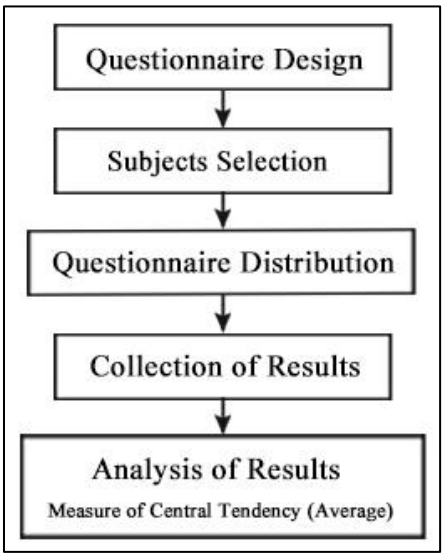

Figure 1: Flow of Research Methodology.

Questionnaire Design:

Based on our findings from the existing literature, we proceeded to develop a questionnaire. The primary objective of this questionnaire was to gain insights into the specific testing technique employed by each software house and the rationale behind their choice, considering the available alternatives. By focusing on the "why" aspect, we aimed to understand the factors that influenced their decision-making process. The questionnaire encompassed questions related to the testing technique being used, its advantages over other techniques, and any challenges or limitations encountered during its implementation. By gathering responses from a diverse range of software houses, we were able to obtain a comprehensive overview of the industry's practices and perspectives regarding agile testing methodologies.

This approach allowed us to bridge the gap between the theoretical knowledge gained from the literature review and the practical insights shared by the software houses. By aligning the research question with the existing literature and formulating a targeted questionnaire, we explored the specific testing techniques adopted by software houses and gained a deeper understanding of their reasoning and decision-making processes. The designed questionnaire may be perceived as lengthy due to the comprehensive nature of the study. To ensure a thorough understanding of the agile testing methodologies and gather valuable insights, we intentionally included a range of questions. The questionnaire is shared in the Appendix at the end of the paper.

Subjects/Target Audience Selection:

We specifically targeted practitioners from software houses active in the Pakistan Software industry who are members of P@SHA (Pakistan Software Houses Association for IT and ITeS (1992–Present)). This organization is a functional trade body and registered association for the IT industry in Pakistan (www.pasha.org.pk) to ensure the reliability of our results. Our target respondents are the Software Quality Testers, Software Quality Assurance, and Software Quality Engineers who have adopted the agile methodology techniques for testing their software.

Among the diverse range of respondents, we gathered from various companies across Pakistan, the top respondent group consisted of individuals holding different positions within their respective organizations. Notably, 39.8% of the responses were from members of the

Quality Assurance and testing teams, while 31% were from programmers or developers. Additionally, 10% of the responses came from Project Managers, and the remaining responses were contributed by individuals of various designations such as CEO, software engineer, and team leads as shown in Figure 2.

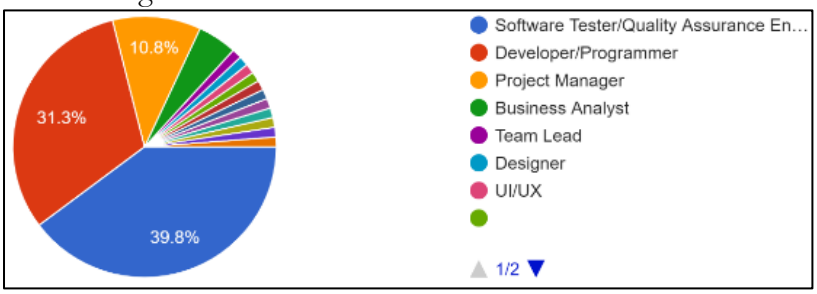

Figure 2: Respondents' Designation.

In terms of familiarity with agile testing, as demonstrated in Figure 3, the surveyed respondents displayed varying levels of expertise. A significant portion, accounting for 43.4%, indicated an intermediate level of familiarity with agile testing practices. Around 31.3% expressed advanced familiarity, while 6% possessed a high level of expertise. The remaining respondents considered themselves beginners in agile testing.

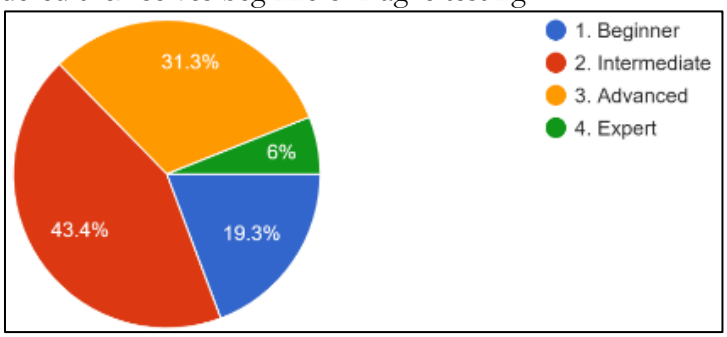

Figure 3: Level of Familiarity of Respondents.

These findings provide valuable insights into the demographics and satisfaction levels of the respondents, shedding light on the diverse perspectives and experiences within the software houses surveyed.

Questionnaire Distribution:

To ensure wide distribution and maximum participation, we utilized the convenience of Google Forms for designing our questionnaire. The form was shared through various channels, including email, WhatsApp groups, LinkedIn, Instagram, and Facebook. To reach out to different organizations, we directly contacted Human Resources (HR) personnel through WhatsApp and LinkedIn, seeking their assistance in distributing the questionnaire among relevant team members.

This multi-channel approach allowed us to reach a diverse range of participants and collect responses from individuals across various software houses in Pakistan. By leveraging popular communication platforms and engaging directly with HR professionals, we aimed to enhance the visibility and accessibility of our questionnaire, facilitating a broader response rate and ensuring representation from different organizations and roles.

Collection of Responses:

We successfully collected a total of 83 responses from various organizations across Pakistan which are members of P@SHA. This widespread participation reflects the diverse perspectives and experiences within the industry.

Results:

In the analysis of survey responses, we carefully examined the collected data to extract meaningful insights from the participants' feedback. By capturing a snapshot of the responses,

we utilized a visual representation of the diverse perspectives shared by professionals in the field. These responses serve as a valuable resource for understanding the current practices, challenges, and preferences related to agile testing methodologies.

One of the main research objectives of our survey was to determine the popularity of different agile testing methodologies employed in software houses across Pakistan. The survey responses revealed the following distribution: Test-driven development (TDD) was reported as the most widely used methodology, with 72.3% of respondents adopting it. Behavior-Driven Development (BDD) accounted for 20.5% of responses, while Acceptance Test-Driven Development (ATDD) was reported by 32.5% of participants. Exploratory Testing and Session-Based Testing were found to be utilized by 18.1% and 16.9% of respondents, respectively. These findings provide valuable insights into the prevalence of various agile testing methodologies in the Pakistani software industry, with TDD emerging as the dominant approach as shown in Figure 4.

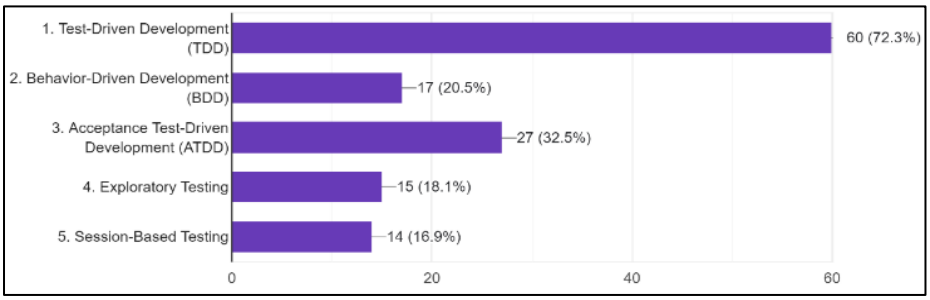

Figure 4: Respondents implemented agile testing methodologies.

In Figure 5, the survey results indicated the primary reasons behind the selection of agile testing methodology. The following statements reflect their responses: Nearly 53% of respondents opted for a methodology aimed at improving collaboration and communication between team members, as well as increasing customer satisfaction by delivering value early and frequently. Almost 27 respondents (representing a specific number) selected the methodology to enhance the adaptability and flexibility of the development process. Nearly 24 respondents prioritized achieving faster feedback and a shorter feedback loop. A total of 28 respondents selected the methodology to promote a test-driven development approach to code quality. Almost 16 respondents opted for the methodology to align with the industry's best practices and standards. However, 15 respondents indicated that they chose the methodology to address specific project requirements and constraints.

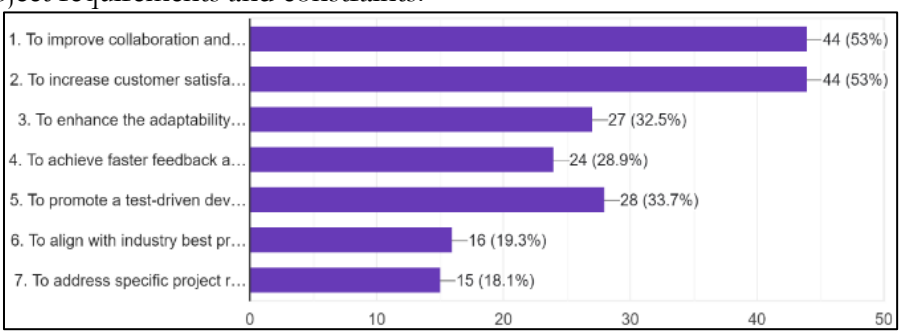

Figure 5: Reason behind the selection of agile testing methodologies

Figure 6 focuses on the factors influencing the respondents' choice of tools to support their agile methodology, several key findings emerged from the survey. Firstly, 49.4% of the respondents acknowledged the significance of the availability of free or open-source options as a determining factor in their decision-making process. Additionally, 45.8% of the respondents highly valued the ease of use and user-friendliness of the tools they selected. Compatibility with their programming language or technology stack was sought by 31.3% of the participants, indicating its importance in tool selection. Moreover, 37.3% of respondents emphasized the

crucial role of strong community support and active development. Integration capabilities with other tools in their development ecosystem were considered by 26.5% of the respondents. Furthermore, 15.7% of participants valued the reputation and reliability of the tool, while 20.5% took into account recommendations from colleagues or industry experts. These findings provide insights into the factors influencing tool selection and highlight the diverse considerations among respondents when choosing tools to support their agile methodologies, 19.3% of respondents considered the cost-effectiveness and affordability of the tools.

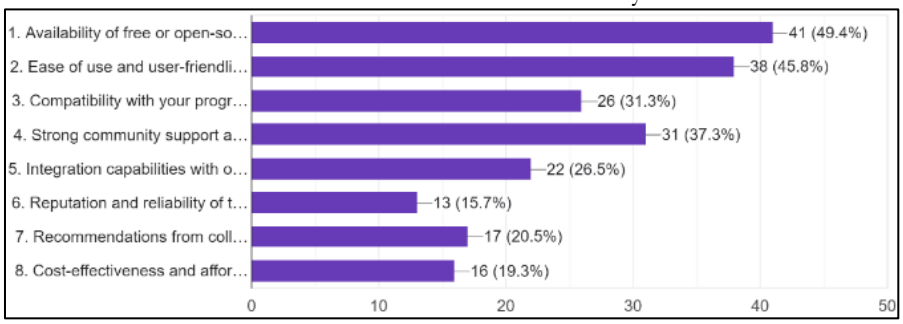

Figure 6: Factors Influenced the selection of agile testing methodologies

Figure 7 illustrates that 51 respondents which account for 61.4%, are satisfied with the efficiency and accuracy of tracking and management during Agile testing. Nearly 26.5% of respondents are Very satisfied and 12% of respondents are neutral. The majority of respondents expressed satisfaction with the accuracy and efficiency of Agile testing methodologies, indicating the overall success of these techniques.

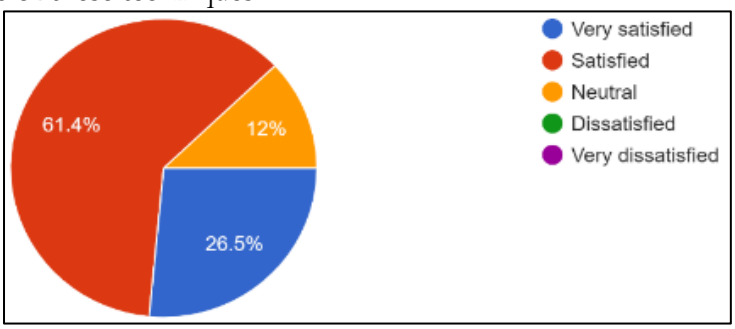

Figure 7: Level of Satisfaction with the efficiency and accuracy of defect tracking and management during Agile testing

Figure 8 demonstrates the impact of agile testing methodologies on collaboration and communication between testers, developers, and other stakeholders of the project. Almost 98% of the respondents agree that agile methodologies help in improving communication and collaboration.

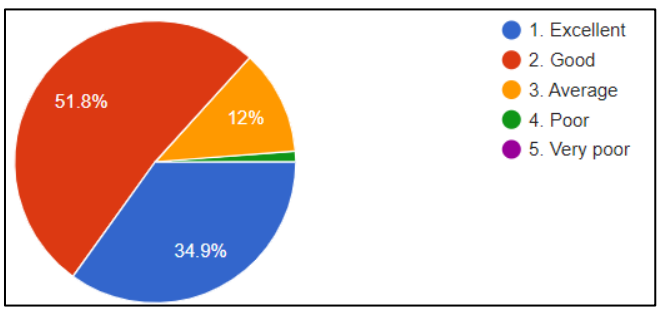

Figure 8: Ratings of the collaboration and communication level between testers, developers, and stakeholders in agile projects

Figure 9 displays that 48.2% of respondents have observed significant improvements in decreasing the number of defects with the adoption of agile testing methodologies. Almost 42.2% of respondents have observed the moderate improvements. This finding underscores the

significant factor contributing to the popularity of agile testing methodologies, which is the observed decrease in the defect rate during the early stages of development.

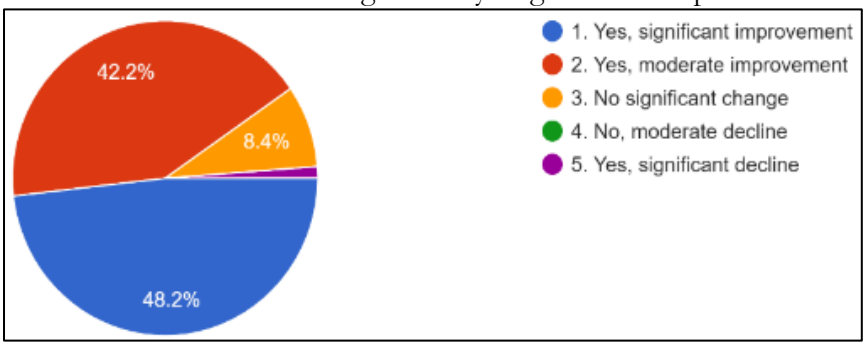

Figure 9: Decrease in the No. of defects or improvement in defect resolution time with the adoption of agile testing methodologies

Figure 10 discusses the different challenges that respondents faced while implementing agile testing tools or methodologies. Lack of clear requirements and user stories and Insufficient collaboration between team members lie in the top two positions, respectively. Difficulty in prioritizing and managing test cases, Limited availability of resources for testing, Inadequate training and understanding of Agile testing practices, Challenges in adapting to frequent changes and iterations, and Ineffective communication with stakeholders come after the above two in the listed order.

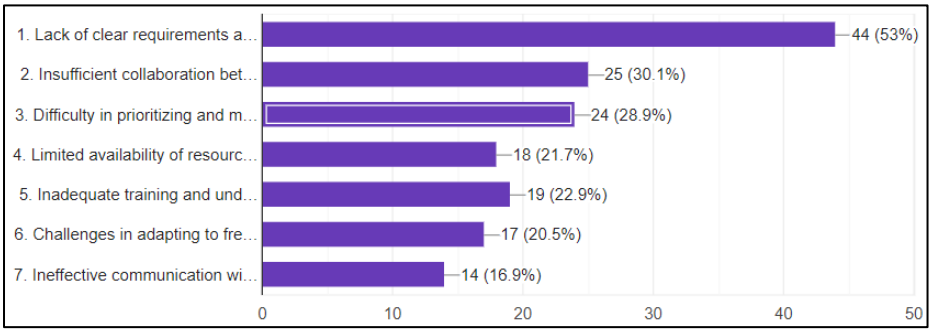

Figure 10: Challenges encountered while implementing agile testing methodologies

Figure 11 provides an overview of how the respondents prioritize and address the defects identified during agile testing. Around half of the respondents prioritize them based on the severity level and a quarter of them go for immediate resolution, the rest of them either schedule those for future releases.

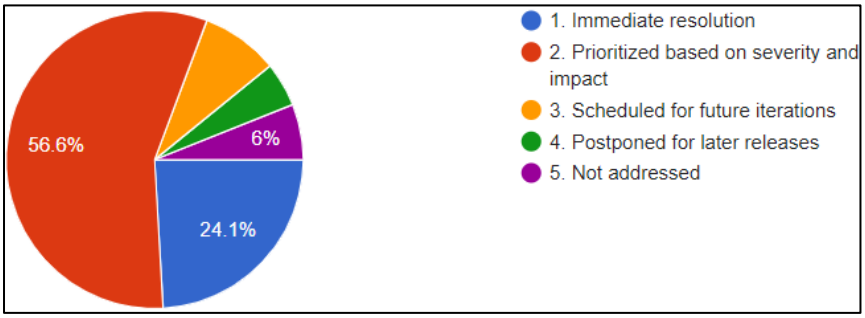

Figure 11: Prioritize and address the defects identified during Agile testing

Figure 12 gives us insights into the metrics respondents used to measure the success of the agile testing methodologies. 60% of the respondents measure success in terms of the number of defects found and fixed. 36% of the respondents measure it in terms of customer satisfaction whereas 33% evaluate it in terms of team productivity. 38% measure it by looking at test coverage and time to market, respectively.

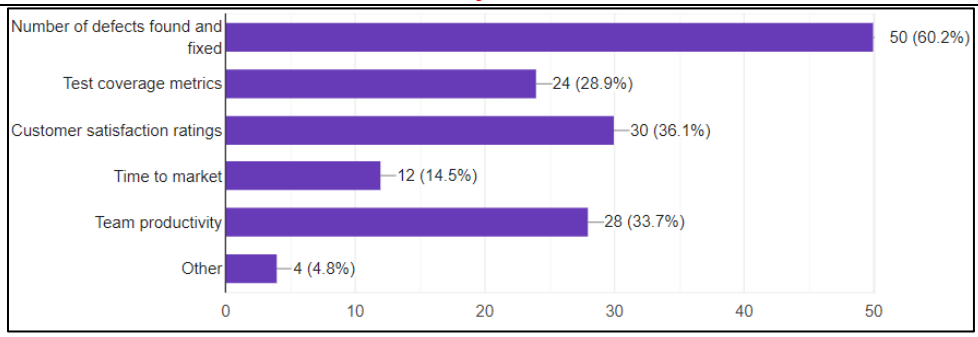

Figure 12: Metrics to measure the success of agile testing methodologies

Figure 13 shows that 85% of the respondents agree that agile testing methodologies helped them in achieving faster feedback and shorter feedback loops. Around 4% disagreed with this whereas 10% were not sure about it. Respondents also show their consent (70%) towards the higher rate of defect detection and resolution with Agile testing compared to traditional testing methodologies as shown in Figure 14.

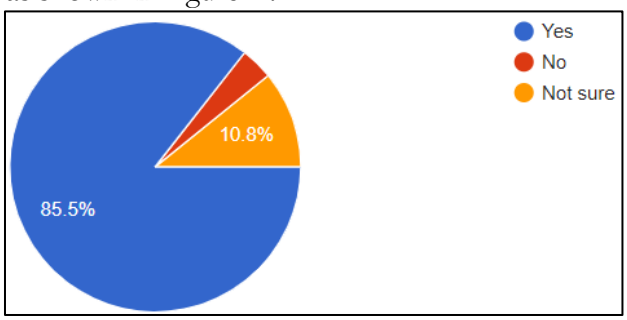

Figure 13: Role of agile testing methodologies in achieving faster feedback and shorter feedback loops

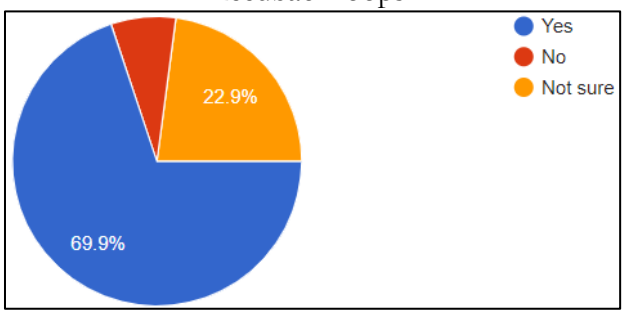

Figure 14: Higher rate of defect detection and resolution with agile testing compared to traditional testing methodologies

Figure 15 depicts that approximately 84% of the respondents agree that agile testing methodologies do adapt to changing requirements and project scope very well. A nominal number of respondents consider this poor.

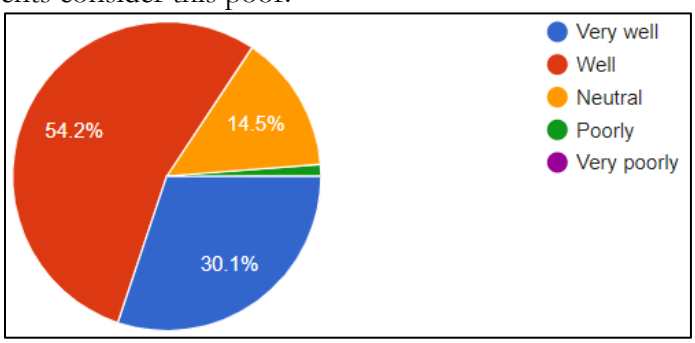

Figure 15: Rate of agile testing methodologies adaption to changing requirements and project scopes

Discussion:

The survey responses provided significant insights into the popularity of Agile testing methodologies in Pakistan software houses. Among the respondents, the most widely used Agile testing methodology was Test Driven Development (72.3%), followed by Acceptance Test-Driven Development (ATDD) (32.5%), Behavior-Driven Development (BDD) (20.5%), Exploratory Testing (18.1%), and Session-Based Testing (16.9%). This indicates the preference for Agile methodologies that promote collaboration, early feedback, and continuous testing. Regarding the reasons for choosing specific Agile testing methodologies, the majority of respondents (53%) cited the aim to improve collaboration and communication between team members. This emphasizes the significance of effective teamwork and communication in Agile development environments. Additionally, 53% of respondents identified the goal of increasing customer satisfaction by delivering value early and frequently. This reflects the customer-centric approach of Agile testing, aligning with the Agile principle of continuously delivering valuable software.

The analysis of factors influencing the choice of tools for supporting Agile methodologies revealed that respondents considered various factors. The availability of free or open-source options (49.4%) ease of use and user-friendliness (45.8%) were identified as key factors. Strong community support and active development (37.3%), compatibility with programming languages or technology stacks (31.3%), and recommendations from colleagues or industry experts (20.5%) were also taken into account. These findings highlight the importance of considering tool-related factors that facilitate Agile testing processes and enhance efficiency. In terms of measuring the success rate of Agile testing, respondents employed multiple metrics. The number of defects found and fixed (60.2%), customer satisfaction ratings (36.1%), and team productivity (33.7%) were the most commonly used metrics. This indicates the emphasis on quality assurance, customer satisfaction, and productivity as indicators of Agile testing success.

The survey results also revealed positive perceptions regarding the effectiveness of Agile testing methodologies. A majority of respondents (69.9%) observed a higher rate of defect detection and resolution with Agile testing compared to traditional testing methodologies. Furthermore, Agile testing methodologies were perceived to adapt well to changing requirements and project scope, with 84.3% of respondents rating their adaptability as either very well or well. These findings contribute to the existing body of knowledge on Agile testing methodologies and provide insights for software houses in Pakistan and beyond. They highlight the importance of collaboration, customer satisfaction, and continuous improvement in Agile testing practices. The results also emphasize the need for suitable tool selection and the use of appropriate metrics to measure Agile testing success.

Further research and exploration in this area focuses on addressing challenges faced during Agile testing implementation, improving training and understanding of Agile methodologies, and investigating the impact of Agile testing on different types of software projects. Overall, this study demonstrates the significance and effectiveness of Agile testing methodologies in software development, emphasizing their ability to enhance collaboration, customer satisfaction, and defect detection and resolution, while adapting to changing project requirements and scope.

Conclusion and Future Work:

In conclusion, this research study examined Agile testing methodologies and their implementation in software houses in Pakistan. The survey responses and analysis shed light on the prevalent Agile testing techniques, factors influencing methodology selection, metrics used to measure success, and the effectiveness of Agile testing in comparison to traditional methodologies. The findings indicate a strong preference for Agile methodologies, with Test Driven Development being the most widely used approach. The reasons for adopting Agile testing methodologies

include improving collaboration and communication, increasing customer satisfaction, and achieving faster feedback and shorter feedback loops. The study also highlighted the significance of factors such as tool availability, ease of use, and community support in the tool selection process. Moreover, the research revealed positive perceptions regarding the effectiveness of Agile testing, as a majority of respondents observed a higher rate of defect detection and resolution compared to traditional methodologies. Additionally, Agile methodologies were perceived to adapt well to changing requirements and project scope. While this study provides valuable insights into Agile testing methodologies in the context of software houses in Pakistan, there are several avenues for future research.

- Firstly, further investigation can focus on exploring the challenges faced during Agile testing implementation and identifying strategies to overcome them. This would help in addressing specific issues and improving the overall effectiveness of Agile testing practices.
- Secondly, additional research can be conducted to enhance the training and understanding of Agile methodologies among software professionals. This would promote better adoption and implementation of Agile testing techniques, leading to improved project outcomes.
- Furthermore, future studies can examine the impact of Agile testing on different types of software projects, considering factors such as project size, complexity, and domain. This would provide a more comprehensive understanding of how Agile testing methodologies can be tailored to specific project contexts.
- Lastly, the research can be expanded to compare the experiences and practices of software houses in Pakistan with those in other countries. This would facilitate cross-cultural insights and help identify global trends and best practices in Agile testing.

By addressing these areas, future research can further contribute to the advancement and refinement of Agile testing methodologies, ultimately leading to improved software quality, customer satisfaction, and project success.

Appendix:

Q1. Designation: Please select your current designation.

- Software Tester/Quality Assurance Engineer
- Developer/Programmer
- Project Manager
- Business Analyst
- Other

Q2. What is your level of familiarity with Agile testing methodologies?

- Beginner
- Intermediate
- Advanced
- Expert

Q3. Which Agile testing methodologies have you implemented or used in your software development projects? (Select all that apply)

- Test-Driven Development (TDD)
- Behavior-Driven Development (BDD)
- Acceptance Test-Driven Development (ATDD)
- Exploratory Testing
- Session-Based Testing

Q4. Why have you chosen to use the above agile methodology in your software development projects? (Select the most applicable option)

- To improve collaboration and communication between team members
- To increase customer satisfaction by delivering value early and frequently
- To enhance the adaptability and flexibility of the development process
- To achieve faster feedback and shorter feedback loops
- To promote a test-driven development approach and ensure code quality
- To align with industry best practices and standards
- To address specific project requirements and constraints.

Q5. What factors influenced your choice of tools for supporting your agile methodology? (Select all that apply)

- Availability of free or open-source options
- Ease of use and user-friendliness
- Compatibility with your programming language or technology stack
- Strong community support and active development
- Integration capabilities with other tools in your development ecosystem
- Reputation and reliability of the tool
- Recommendations from colleagues or industry experts
- Cost-effectiveness and affordability

Q6. How satisfied are you with the efficiency and accuracy of defect tracking and management during Agile testing?

- Very satisfied
- Satisfied
- Neutral
- Dissatisfied
- Very dissatisfied

Q7. Have you observed a decrease in the number of defects or an improvement in defect resolution time with the adoption of Agile testing methodologies?

- Yes, significant improvement
- Yes, moderate improvement
- No significant change
- No, moderate decline
- Yes, significant decline

Q8. What challenges have you encountered while implementing Agile testing methodologies?

- Lack of clear requirements and user stories
- Insufficient collaboration between team members
- Difficulty in prioritizing and managing test cases
- Limited availability of resources for testing
- Inadequate training and understanding of Agile testing practices
- Challenges in adapting to frequent changes and iterations
- Ineffective communication with stakeholders

Q9. How do you prioritize and address the defects identified during Agile testing?
Immediate resolution

- Prioritized based on severity and impact
- Scheduled for future iterations
- Postponed for later releases
- Not addressed

Q10. Have Agile testing methodologies helped in achieving faster feedback and shorter feedback loops?

- Yes
- No
- Not sure

Q11. How do you measure the success rate of Agile testing in your projects? (Select all that apply)

- Number of defects found and fixed
- Test coverage metrics
- Customer satisfaction ratings
- Time to market
- Team productivity
- Other

Q12. How would you rate the collaboration and communication between testers, developers, and stakeholders in Agile projects?

- Excellent
- Good
- Average
- Poor
- Very poor

Q13. Have you observed a higher rate of defect detection and resolution with Agile testing compared to traditional testing methodologies?

- Yes
- No
- Not sure

How well Agile testing methodologies do adapt to changing requirements and project scope?

- Very well
- Well
- Neutral
- Poorly
- Very poorly

Q14. What improvements or changes would you suggest to enhance the success rate of Agile testing methodologies? (Open-ended)

Q On a scale of 1 to 10, how satisfied are you with the overall outcomes of Agile testing in your projects?

(1 = Not satisfied, 10 = Extremely satisfied)

References:

[1] S. O. Barraood, H. Mohd, and F. Baharom, "A Comparison Study of Software Testing Activities in Agile Methods," 2021.

[2] A. R. L. Halounová, C. Heipke, "XXIII ISPRS Congress, Proceedings and Results – Volume XLI, Part A, 2016", [Online]. Available: https://isprs-archives.copernicus.org/articles/XLI-A/1/2018/

[3] J. Yu et al., "An Agile Testing Framework of Four Quadrants," J. Phys. Conf. Ser., vol. 1792, p. 12004, 2021, doi: 10.1088/1742-6596/1792/1/012004.

[4] J. C. S. Coutinho, W. L. Andrade, and P. D. L. Machado, "Requirements engineering and software testing in agile methodologies: A systematic mapping," ACM Int. Conf. Proceeding Ser., pp. 322–331, Sep. 2019, doi 10.1145/3350768.3352584.

[5] "Software Engineering: Sommerville, Ian: 9780133943030: Amazon.com: Books." Accessed: May 22, 2024. [Online]. Available: https://www.amazon.com/Software-Engineering-10th-Ian-Sommerville/dp/0133943038

[6] L. Pemmasani, "Study of Software Testing Techniques in Agile Development".

[7] P. Singh and P. Patel, "Impact of agile testing over traditional testing," Int. J. Converging Technol. Manag., vol. 1, no. 2, 2015.

[8] R. K. Lenka, S. Kumar, and S. Mamgain, "Behavior Driven Development: Tools and Challenges," Proc. - IEEE 2018 Int. Conf. Adv. Comput. Commun. Control Networking, ICACCCN 2018, pp. 1032–1037, Oct. 2018, doi: 10.1109/ICACCCN.2018.8748595.

[9] R. Ramler and C. Klammer, "Enhancing Acceptance Test-Driven Development with Model-Based Test Generation," Proc. - Companion 19th IEEE Int. Conf. Softw. Qual. Reliab. Secure. QRS-C 2019, pp. 503–504, Jul. 2019, doi 10.1109/QRS-C.2019.00096.

[10] B. Losada, J. M. López-Gil, and M. Urretavizcaya, "Improving agile software development methods using user objectives: An end user guided acceptance test-driven development proposal," ACM Int. Conf. Proceeding Ser., Jun. 2019, doi: 10.1145/3335595.3335650.

[11] M. Souza, A. C. Dias-Neto, I. K. Villanes, and A. T. Endo, "On the exploratory testing of mobile apps," ACM Int. Conf. Proceeding Ser., pp. 42–51, Sep. 2019, doi: 10.1145/3356317.3356322.

[12] R. Copche et al., "Exploratory testing of apps with opportunity maps," ACM Int. Conf. Proceeding Ser., Nov. 2022, doi: 10.1145/3493244.3493248.

[13] J. Itkonen, "Empirical studies on exploratory software testing," 2011.

[14] A. A. M. Qamar, Nosheen, "Evaluating the impact of pair testing on team productivity and test case quality–A controlled experiment", [Online]. Available: The discussion of results, particularly the degradation plots and RMSE values, lacks depth. More detailed interpretation of what these results mean in practical terms would be beneficial.

[15] M. M. Moe, "Comparative Study of Test-Driven Development TDD, Behavior-Driven Development BDD and Acceptance Test–Driven Development ATDD," Int. J. Trend Sci. Res. Dev., vol. Volume-3, no. Issue-4, pp. 231–234, Jun. 2019, doi: 10.31142/IJTSRD23698.

[16] I. K. Raharjana, F. Harris, and A. Justitia, "Tool for Generating Behavior-Driven Development Test-Cases," J. Inf. Syst. Eng. Bus. Intell., vol. 6, no. 1, pp. 27–36, Apr. 2020, doi: 10.20473/JISEBI.6.1.27-36.