SOFTWARE TESTING PRACTICES IN IT INDUSTRY

MS INFORMATION TECHNOLOGY

ASRA MAJEED
(MSIT-S19-021)
Session [2019-2021]

FACULTY OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY
THE SUPERIOR COLLEGE, LAHORE, PAKISTAN

SOFTWARE TESTING PRACTICES IN IT INDUSTRY

A thesis submitted in partial fulfillment of the requirement for the Degree of Master of Science in Information Technology

Submitted By
Asra Majeed
(MSIT-S19-021)

Research Supervisor
Dr. Waseem Iqbal

FACULTY OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY
THE SUPERIOR COLLEGE, LAHORE, PAKISTAN

DECLARATION

I hereby declare that the work being presented in this thesis entitled, "**SOFTWARE TESTING PRACTICES IN IT INDUSTRY**", submitted to the Faculty of Computer Science and Information Technology, Superior College, during the period from, 2019 to, 2021. This thesis is the presentation of my research work. Every effort is made to explicitly demonstrate this wherever contributions from others are involved, with due regard to the literature, and acknowledgment of joint studies and discussions. I further announce that this work is the product of my investigations, except where sources are found and the work of others is exempt from plagiarism.

Lahore, June 2021 Signature: _____

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

_____

Dr. Waseem Iqbal

Research Supervisor

The undersigned hereby certify that they have read and recommended the thesis entitled **SOFTWARE TESTING PRACTICES IN IT INDUSTRY** by **Asra Majeed (MSIT-S19-021)** for the degree of Master of Sciences in Information Technology.

_____

Dr.

Research Supervisor

_____

Dr.

Thesis Committee Member

_____

Dr.

Thesis Committee Member

_____

Dr.

Head of Department (IT)
The Superior College, Lahore.

DADICATED TO

This thesis would be incomplete without a mention of the support given me by my **PARENTS, BROTHER, and TEACHERS** to whom this thesis is dedicated. They were my own “soul out of my soul” who kept my spirits up when the muses failed me. Without them lifting me up when this thesis seemed interminable, I doubt it should ever have been completed.

ACKNOWLEDGEMENTS

All acclamation, appreciation and praise for the ***ALMIGHTY ALLAH***, the most gracious and compassionate, whose blessing and exaltation flourished my thought and thrived my ambition to have the cherished fruit of my modest efforts in the form of this manuscript from the blooming spring of blossoming knowledge. My special praises for the ***HOLY PROPHET MUHAMMAD (Peace Be Upon Him)*** who is forever a torch of guidance for the entire humanity.

With a proud sense of gratitude, I acknowledge that this manuscript has found this shape under the kind supervision, inspiring guidance and sympathetic attitude of **Dr Waseem Iqbal** Assistant Professor, Faculty of Computer Science & IT The Superior College Lahore. His thorough analysis and meticulous critique improved not only the quality of this exposition, but also my overall understanding of this research project.

I offer my great sense of gratitude to my supervisory committee **Dr. Arfan Jaffar**,

Dean Faculty of Computer Science & IT The Superior College Lahore and **Dr. Imran Khan,** Head of Department of Faculty of Computer Science & IT The Superior College Lahore, for providing valuable suggestions, competent guidance and boosting up my morale during the conduct of this study.

Last but not least, I wish to express my humble obligations to my great parents, who always raised their hands for my success and gave me confidence and atmosphere that initiated me to achieve high academic goals. I feel it incomplete if I do not extend my thanks to my affectionate and loving father, ***Abdul Majeed Awan*** *and* my loving mother**,** ***Nargis Bano*** for his love, affection and prayers which enabled me to acquire this long-adhered aim.

Contents

Table of Figures

List of Tables

Abstract:

Software testing is fundamental and key procedure in software development life cycle of software IT industry, in which assurance of software product and identification of errors with different design. Software testing en-vaulted the practices of service-oriented software engineering in IT industry which covered the white box testing, black box testing, unit testing, integration testing, system testing and acceptance testing and also verification and validation on which different results was obtain through online survey and experiments. We have planned well-structured research on software testing practices for service-oriented software engineering by proposing a well-defined questionnaire to determine the importance, significance problems and practices to obtain the objective of this research. We conduct online survey through google form the participants during survey was the IT managers, Android Developers, IT executive, IT lecturer, IT students, Software developers and many other people from different organizations and software houses. It helps software tester that the understand all problems in better way. In this Research Different empirical studies, different aspects including bugs, mutation for generation of test have examined. The aim of this research is to identify the problems, advantages and practices for software testing. In the recent research software testing conducted by different software companies. Software companies mainly focus on technology and methodology, automation tools, training testing and collaboration with the software industries. Selenium tool is used for software testing activities. The main issue is found during software testing is lack of knowledge and collaboration of software testers among all of them. And also, time and cost. In this research we also revealed that the software testing Is expensive and time-consuming process. In future we will focus and study about the time consumption and cost of software during software testing.

Keywords: software testing, software testing practices, Automated software testing

1 Introduction:

Software testing is an essential and expensive procedure in the life cycle of software. Testing is an essential aspect of the process of software design. The testing is used to guarantee that both activities and non-criteria are met by a software product. The increasing scope and diversity of software has made software testing necessary, since faults may have catastrophic consequences [1][4]. Software testing, A collaborative knowledge-intensive activity is a subclass of software engineering. The validation and verification of high quality software products should be performed throughout the development process, as software development is the most vulnerable task[2]. In addition, Poor testing of software typically results in significant risks and repercussions. For example, the US National Institute of Standards and Technology (NIST) study of 2002 finds that there are considerable economic costs for the lack of software testing architecture in the United States. The IT sector worldwide faces enormous obstacles when it comes to developing good goods on time and on budget. The Standish Group is a widely renowned technological development and value analysis market research firm, which regularly releases a series of studies entitled CHAOS[1] on this issue. Many Sri Lankan software businesses are the world's top IT consulting and outsource service providers. Therefore, these software businesses have to reach or surpass the quality criteria of prominent software publishers to achieve worldwide success, in particular in banking and financial, insurances, healthcare, communications and media. In such circumstances, these offshore enterprises must embrace technological progress in order to use testing methodologies to generate software products of high quality and reliability [2].

Automotive is one of the domains of application that has seen the greatest growth in technical innovation in recent years, especially for software-intensive elements. Car OEMs (original equipment manufacturers) increasingly develop their automobiles into sophisticated electronically controlled systems from mechanical gadgets. Today, electronic systems, more complicated and CAN linked (Govern Area Networks), control 85% of the functionality of the car. Software (with increasing demand both in size and in complex) is thus a key auto part since it is part of integrated systems for the electronic handling of remotely controlling a wide range of vehicle activities (navigation and infotainment included). Over the past 15/20 years the number of ECUs from inexpensive car types has expanded markedly [3].

1.1 Service Empirical Studies of Software Testing:

Different empirical studies have examined different test aspects, including coverage, mutations, bug discovery and automated techniques for test generation. Although several of these studies focus at test cases navigated, most test data are still produced by hand today; the research mentioned above only look at artefacts created by practitioners (such as code and bug reports), rather than surveying them. [4].

Testing is a crucial aspect of the SQA process, as it is the primary activity for finding and resolving technical faults in software source of code. It evaluates the general usage, performance, safety and compatibility of products. It is not only vital to guarantee quality, but also to build software. Software testing is considered as a set of duties, software-related methods and technologies termed for an umbrella activity Software related tools may be regarded as a systematic technique under

an umbrella activity [5]. Testing is one of the costliest components of almost any software development life cycle. Test problems are associated with cost of low quality, malfunctioning programs and bugs, which produce large extra charges for software manufacturers throughout the development cycle. The cost of testing is increasing; the software industry has recognized a need to reduce the increasing expenses of test environment management [6].

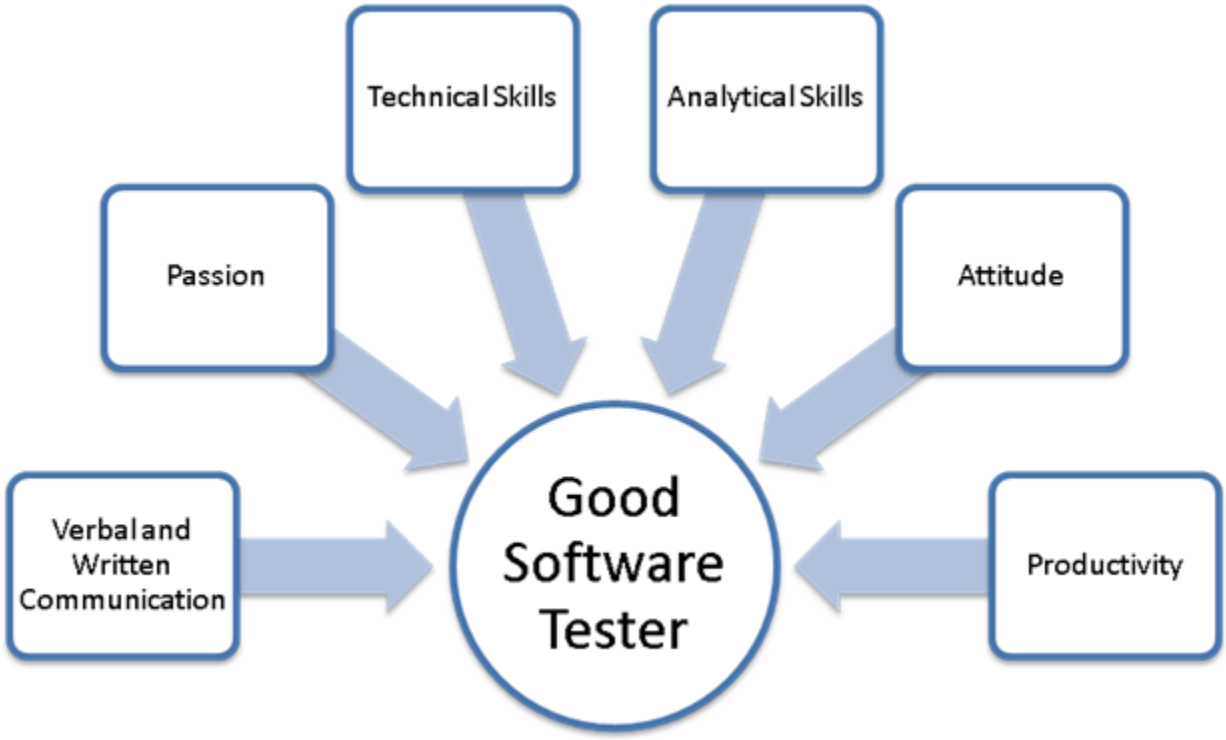

Figure 1 Service Empirical study of software testing

A defined testing plan for building safety-critical software systems should outline the techniques for finding, tracking, analyzing, and removing dangers connected with the system. Despite best build-then-test methods, software-related system faults have progressively dominated System integration rework effort and acceptance testing failures. According to various safety-critical systems assessments, 80% of all defects are not found until the system is integrated or later [7]. Our study's goal was to look at software companies' testing techniques, Tools and models of development processes for updating industrial practices. [8]. Various approaches have been created for improving software testing procedures (STPI) to help companies to evaluate and enhance their testing procedures. To enhance the software test process of a single company [9].

1.2 Automated Software Testing:

Testing may be categorized into two fundamental categories: conventional and computerized. The tester usually follows a stated testing strategy, which includes a collection of test cases, to

guarantee that the test is comprehensive. Automated software testing covers software test automation [10]. Computer programming testing activities serve a key function.

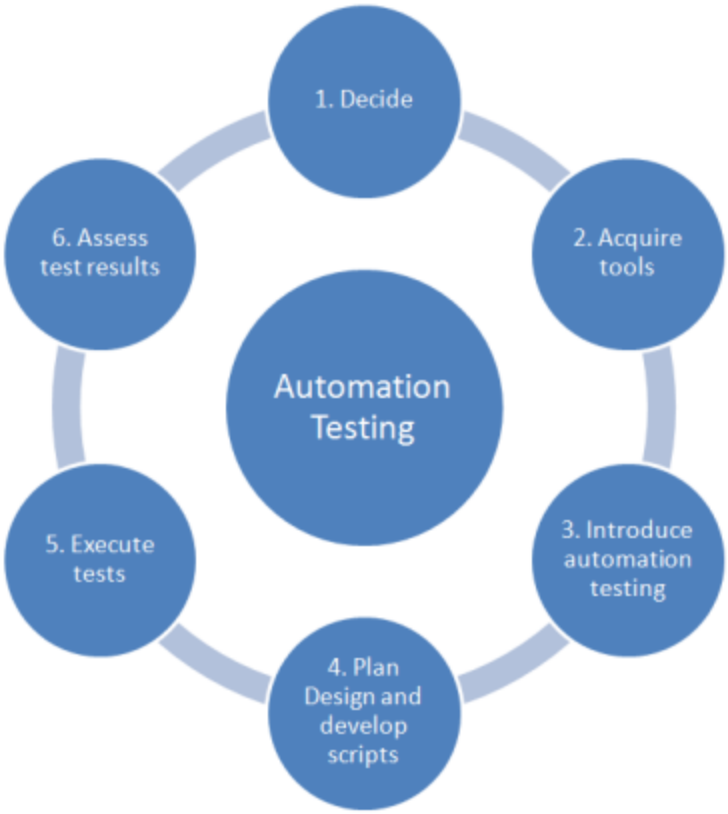

Figure 2. Automated Software Testing

Development process as it represents a major quality assurance technique to help the discovery of software product flaws in the production environment prior to its deployment. Software testing methods should be developed at Multiple level (i.e. unit, integration, system, acceptation), using various approaches and viewpoints (e.g. feature, structural and glitch) (performance, usability, security, functionality, and so on). In addition, software development process by skilled staff undertakes these tasks should also be performed to a software product generally backed by a suitable computer infrastructure [11].

1.3 Software Quality Assurance in Software Testing:

Software Quality Assurance (SQA) is a set of ways to ensure that processes lead to the creation of a high-quality product. Quality assurance is given less weight by software development companies because it is the first line of defense when deadlines are missed.[12]. In recent years, software products have been rapidly developing and successful in easing end-user needs to execute various tasks utilizing computers. The end-user frequently desires to have software items implemented as

quickly as possible. They require a built software solution with new features in a short amount of time. It is important to perform testing over and over and quest for undercover future defect [13]. Software testing is one of SE's most active areas. Since 1994, According to our search, about 101 secondary research in the topic of software testing were published. It is important to undertake a tertiary study with so many secondary studies in this topic in order to acquire a clearer image of the research environment. A tertiary examination is a secondary examination research (or a systematic review of systematic reviews) [14]. Risk-based testing (RBT) is a well-known and successful method for tackling the problem of continually dwindling test capacity, which has recently attracted much attention. It is based on the obvious notion of concentrating testing on scenarios which provide the most important software systems conditions. A lot of benefits might come from its correct implementation. RBT enhances resources (money, time, people), reduces risks, supports early identification of crucial areas and gives management decision-making help. [15].

Load practitioners and software engineers in testing and analysis of large-scale software systems. Unlike functional testing, where we have a defined target (pass/fail criteria), load tests might include one or more functional, non-functional and distinct pass/fail criteria [16]. Two big communities are the worldwide software industry and software engineering (SE). Unfortunately, however, the degree of SE cooperative cooperation between industry and universities (IAC) remains quite low relative to the number of activities of both communities[17]. This latest report focuses on business-academic software testing relationships. A number of collaboration events between industry and academics, such as the Tester Workshop: Collaboration in academic and industrial, practice and innovation research [18], follows. Secure software development consists of the aggregation of many secure development activities. This involves best practices for improved software processes and measures to identify and reduce software risks in the SDLC as quickly as feasible. In general, software engineers are not security professionals and confront significant security limitations. Security mechanisms are complicated and it is difficult for the typical developer to grasp how the security criteria of these mechanisms are met and how the objective of safe implementation is achieved. Software engineers are not necessarily security specialists, it is difficult for them to spot possible threats and vulnerabilities during the early development phase [19]. Big data analytics and AI technologies have rapidly advanced and have made several AI-based software and apps widely accepted and utilized in everyday life. AI software and applications are produced on the basis of state-of-the-art machine learning models and methods via extensive data training to incorporate various artificial intelligence characteristics and skills [20]. Many large-scale systems range from e-commerce websites to telecommunications infrastructures have to accommodate tens or millions of user’s concurrent access. Studies suggest that failure in such systems is usually caused by failing to match user requirements rather than by feature bugs [21]. The software development sector is one of the most important and fascinating sectors of IT, which has the ability to enhance the economic growth rate and can make a useful contribution to the economic development of any nation. As an example, we may mimic India's software success [22]. The creation of big software systems is a difficult and error-friendly process.

Errors may arise at any step in the creation of software. These mistakes, commonly called bugs, may lead to large time-and-money losses, if not recognized and deleted as soon as feasible. Software testing plays an important part in the life cycle of software development (SDLC). It stops the spread of mistakes and hence reduces expenses. However, software testing alone might be expensive [23]. Software testing by people educated in doing these frequently supported operations should be applied to a software product.

Suitable computer infrastructure. The disparities in methods for applying global software testing and practices together with the (in)mature level of knowledge in local markets have inspired us to evaluate the significance and the customary use of these testing activities and practices by both nations' software industry [24]. In the quality management of most software products, software testing is critical. Despite the large variety of instruments available, this still requires a lot of human work, i.e. socio-tech rather than merely technological [25]. This paper aims at understanding how software product companies build management strategies in terms of structure, creativity, branding and positioning, competition development, etc. during various phases of the enterprise product development process and suggest best practices that can be adopted to ensure their products are successful in the market [26]. Agile approach is an incremental and iterative process, and is often used as a solution for changes in requirements in Pakistan's industrial projects. Product distribution is done with minor iterations/repeats, however ensuring the product quality is a key and essential aspect as well as a demanding effort. The quality of the product generated utilising agile methods should be ensured. The research will focus on five main components of software testing, namely methodologies for software testing, methods and techniques for software testing, test standards, automated test tools and training and training [27]. Testing and clinical facilities are the most expensive tasks in the program's life cycle. Several studies have demonstrated, however, that investment in testing processes may enhance product quality and cut costs. Software engineers may close the gap between formal training and industry practices by monitoring the industry to generate more industry-sustainable graduates. In software firms, we have investigated how their products are tested and what type of methods they are following [28] to evaluate the current state of industry. Testing and clinical installations are the costliest jobs in the life cycle of the programmed. However, several studies have also shown that corporations may improve product quality by investing in test methods and saving expenses. Educators in software engineering can narrow the gap between actual education and industrial practice by looking at the industry. In order to learn about the present status of the business, we carried out a research in software companies to evaluate how the products are tested and the process models they use [29]. IT is the core of all sectors and IT is the backbone of software. Failure of software may lead to devastating results. Another challenge is the cost of software failure, since software failure was responsible for just $62 billion. The cost of failure in software is more catastrophic, as in real time, finance and the health sector, software is utilized where the consequence of a single mistake may have substantial effects [30]. The most significant aspect of SQA and the most usually conducted SQA activity is Software Testing (ST). It is a product-oriented activity where software is executed and its behavior or results are observed. Although, Software Quality Assurance and Testing (SQAT) is a costly

activity in software development, still it is an integral part and highly important in software development [31]. Software testing is an expensive and important activity to help create and guarantee the quality of any software system. While a wealth of research exists within the field, evidence-based information about testing strategies and solutions is not widely used inside business. Software testing research offers and assesses an extensive array of test procedures, such as models, processes, techniques, tools, metrics, etc., designed to eventually enable software testing in an industrial context [32]. The dominant strategy for software testing and quality assurance has for a long time been manual testing and, in recent years, fresh interest has been shown in techniques such as exploratory tests that emphasize the imagination and experience of testers. Pr-determined series of procedures performed on a high level of system abstraction to check or validate system compliance, i.e. system tests, or as expected in its area marked, i.e. acceptance testing [33], are more typical manual testing. Regression testing is an unresolved and increasingly major topic in the development of industrial software. Regression testing is a key step towards quality assurance and represents a crucial barrier for the smooth development of large-scale software (e.g. continual integration and delivery). Similarly, the regression testing of industrial software (e.g., in product lines / product variations) is non-trivial [34]. Now a day, practically every domain is produced by industrial software that is usually sophisticated and enormous in size. The majority of the software is used openly in the present distributed computing environment, which leads to a series of safety issues. These issues are easily employed by cyber - thieves and the quantity of software assaults increases rapidly [35]. For software projects, the experience of the members of the software design team is crucial, since they affect critical features of software characteristics such as performance, reliability and simplicity directly. Many studies have analyzed the effects on software quality of team members' skills, team performance and the competitive benefits of such talents on the market. As has been shown by various researches, the absence of experienced staff members on Software Development Teams impacts directly the prices, delivery timeframes and even completion of the software projects [36]. The usage of software and its size are both increasing rapidly because of advancements in ICTs, which result in a growth in software complexity. Software companies/sellers are striving, by decomposing the product into numerous components/modules, to address this complexity [37]. Testing is one of the time-consuming operations in the software development cycle. It is one of the key approaches to guarantee software quality and aimed at identifying defects. The relevance of the test is evident in the whole literature, in which multiple accident records caused by software failures occur. There are a number of approaches for testing requirements. However, for numerous technological reasons, the causes of a software-based mishap may arise. Testing relies on the end product, regardless of the methods chosen, to please both the user and developer [38]. SMEs account for up to 85 percent of all software in most nations. However, even tiny businesses require efficient and effective software technology solutions in order to continue and expand. However, the application of software engineering technology is a challenging issue for small and medium-sized enterprises since they generally have limited resources and stringent time limits. Software testing, in particular, is a demanding and costly process which consumes up to 50 per cent of overall

development expenditures and is thus an ideal starting point to improve software and product quality in small and medium-sized businesses (SME) [39].

1.4 Service Oriented Architecture in Software Testing:

Service Oriented Architecture (SOA) has an interesting property that components of software designed and created by separate organizations, are dynamically assembled for more useful purposes. However, the same factors which allow for flexible compositions also restrict the use of certain conventional methods of testing, making it difficult and costly to validate SOA. Web services generally only reveal one interface sufficient to activate it and write some general (black box) tests, but not sufficient to allow a tester to have proper application knowledge and the quality of independence integration in web services. To remedy this shortfall, we propose a way to increase the visibility of web applications for testers by introducing a service that provides coverage information. The Service-Oriented Coverage Testing (SOCT) approach informs the tester about the service they offer without the internally assigned service being disclosed. Testing feedback is provided as a service in SOCT, and the SOA loose connection and neutrality notions are thus maintained. In this article, I motivate, define and use the SOCT approach as an example. I also conduct a research to evaluate the feasibility of SOCT and offer an advance assessment of its sustainability and value [43].

1.5 Validation and verification in software testing:

Despite automation testing developments, manual software tests continue to be a key and extensively used integration testing approach (V&V). Software testing is typically considered as a test case technique that is extensively predetermined utilizing test case design approaches. In this scenario test method (TCT), the goal is to capture the necessary information in the new tests. Even when performed manually, the ultimate completion of the test is considered mechanical. The predefined test cases are performed while the findings are being compared with the expected results. [40].

The aim of this chapter is to examine the problems, advantages and practices that contribute to the improvement of processes in business and academics [41]. The most gen

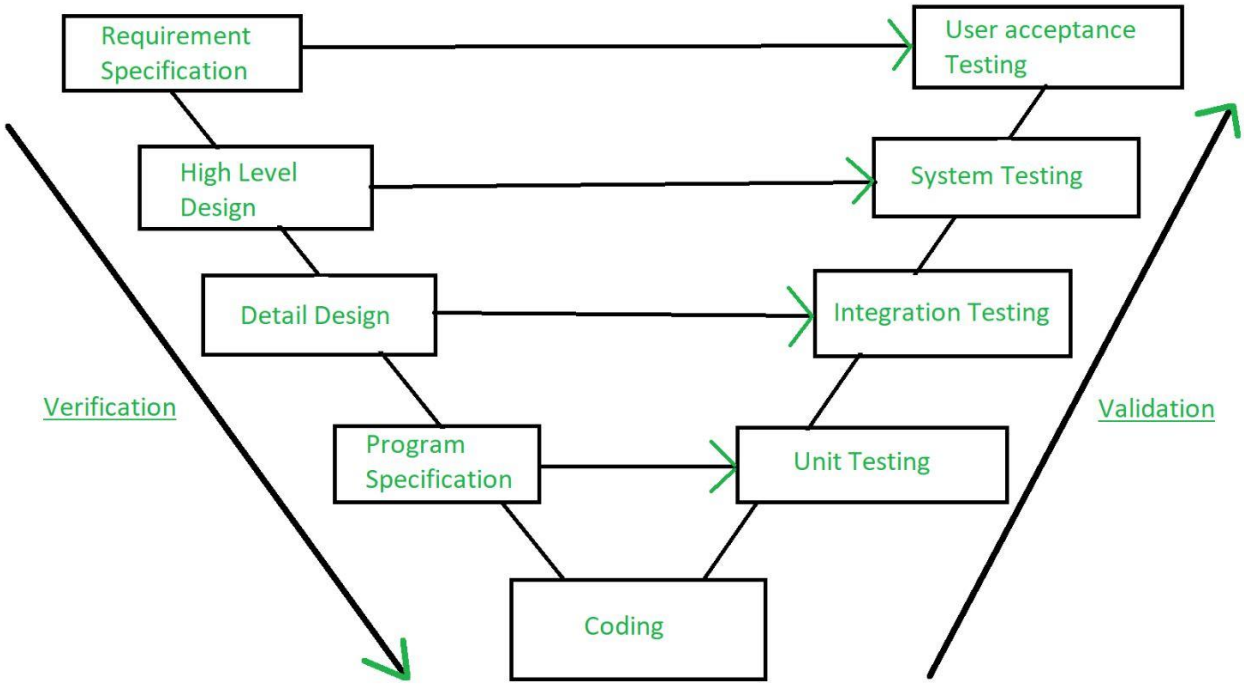

Figure 3 Verification and validation in software testing

early recognized and utilized quality control measure for most systems produced today is software testing. Testing is also the final and most crucial defensive line before releasing the programmed for the detection and removal of faults in many projects. Nonetheless, the many reports of software faults reveal clearly that software testing frequently fails and so significant fault products are released [42].

1.6 Purpose Of study:

The purpose of the study is to determine the major issue facing in Software testing practices in service-oriented software engineering while conducting the survey through google form to collect the persons' perspective of view connected to studies. There were several questions concerning black box testing, box testing, code review, unit testing, user admission testing in provider software engineering.

1.7 Problem Domain:

Many serious problems related to Software Testing Practices associated with software development challenges in services oriented. The Good testing is a major problem in which customers faces that software testing from several viewpoints is a root problem. Due to high cost a large proportion of development problem in software testing. During survey some challenges which are identified as.

1. Test planning and Test strategy not markup during software testing.

2. Test scheduling is just as important as planning. I have noticed that many testers fail to predict the amount of time required for testing,
3. Most of organizations do not follow best practices for Software Testing practices

A lot of research was done on Software testing practices problems such as white box testing, black box testing, timing, testers goals, data, state-based behaviour, constrains, scenarios, validation, verification time management, planning and etc. All these effects the service oriented related software testing phases.

1.8 Objective:

The aim of this qualitative research was to examine and comprehend in practise the challenges of software testing and to identify suggestions for improvement on those questions. The research focused on organizational units developing and testing technical program for automation or telecommunications, which performed an examination of testing processes and which interviewed 26 organizations. Five firms were further chosen for an in-depth conceptual case study from this sample. The study produced assumptions that a design of software projects should foster testing as an architectural characteristic and use experts to improve the implementation of testing. Testing tools based on usability and configurability criteria should also be picked. These data may be utilised to increase the efficiency of software development and to design an organizational testing strategy.

2 Literature Review:

In 2013, Wahid Garous et al. revealed a review of an experiment in provincial programming by experts from Alberta, Canada. To get a broader cross-border view of this point (on the Canadian National Scale), they conducted an updated review that summarized the requirements revised in 2010. Unlike the previous audit across Alberta (53 programming experts), more members (246 experts) in the cross-border review reported on the research plan, implementation, and results. The post pointed out that the necessity of testing has lately been given greater emphasis by the Canadian business. This may be shown via increasing awareness of testing training. In recent years, noting that these courses are becoming increasingly popular. The wider acceptance of agile methods that play an important role in testing also seems to be an important contribution to the popularity of software testing and increasing training in Alberta and Canada [1].

In 2018, Shanmuganathan Vasanthapriyan used empirical research on the current state of software testing practices conducted by software companies in Sri Lanka. They mainly focus on testing methodology and technology, automation tools, value testing, training testing, and academic collaboration with the software industry. Agile methodology has become very popular in software development by software companies. In general, the use of tools such as Selenium for testing activities has aroused great interest [2].

In 2019, Kochhar et al. raised the issue of the quality of test cases and asked them to investigate good test cases. In order to answer this important question, 21 questions and 261 professionals from many large and small companies and platforms in 27 countries/regions were requested to develop and validate 29 hypothesis describing appropriate Test cases and techniques of testing Attributes. These capabilities address a variety of characteristics including test case content, size and complexity, coverage, maintenance and error repair. They have high features and practitioners defend their degree of agreement with them, highlighting recommended practices and trade-overs that must be taken into account in certain test scenarios [4].

In 2013, Wahid Garous et al. revealed a review of an experiment in provincial programming by experts from Alberta, Canada. To get a broader cross-border view of this point (on the Canadian National Scale), they conducted an updated review that summarized the requirements revised in 2010. Unlike the previous audit across Alberta (53 programming experts), more members (246 experts) in the cross-border review reported on the research plan, implementation, and results. The post pointed out that the necessity of testing has lately been given greater emphasis by the Canadian business. This may be shown through increasing awareness of testing. In addition, in recent years the first author has conducted industrial training in Alberta, noting that these courses are becoming increasingly popular. The wider acceptance of agile methods that play an important role in testing also seems to be an important Contributions in Alberta and Canada to the prominence of software testing and training [1]. In 2016, Lami et al. reported the findings of an empirical research to characterize and analyze the shortcomings typically seen in automotive software development. In the realm of motorsport software development, automotive companies are largely required to

increase the quality of their projects by professional software providers using process models such as AutoCAD SPICETM.

It specifically addresses the process related to software testing, which is a step in a larger study the author is doing. This study tries to uncover common software development using data from actual automotive software developing projects vulnerabilities that have a negative impact on Automotive SPICETM, to describe the status of automotive software development practices, and to provide researchers and practitioners. And proposed improvement measures for these weaknesses [3].

In 2019, Saha (T.) et al. use electronic measuring instruments. Bengal Software started touring about 3 years ago, and 2 years ago, it has had a huge impact. Currently, about 500 software companies are registered with the Bangladesh Software and Information Services Association (BASIS). BASIS released a research report on the export revenue generated by software and products, and reported that Bangladesh exported 800 million US dollars (6720 kroner) of software last year [5].

In 2018, Hyninen et al. explored industry practices related to software testing. They have researched software firms to assess products and process models. As an online application of survey methodologies, data collecting is performed. In addition, the data obtained were compared to their past study to determine how the practices of industry had evolved. Based on their findings, enterprises have moved towards more complicated testing and test infrastructures, using even more agile approaches in mission critical software, and lowering the usage of formal modelling techniques. [6].

In 2018, Kassab discovered that software has increasingly become an essential part of critical safety systems. Unfortunately, Current software professional practices for software analysis and quality assurance operations in critical security systems are almost undocumented. In order to rectify the absence of data, software experts were thoroughly examined in an effort to find these practices. They reported the results of the research on the practice status of software testing of critical safety systems in this study, involving the following three aspects: 1) Inclusion of testing activities into the life cycle of software development; 2) Testing methodologies and technology; 3) Testing index and failure management. They also compare software for non-critical safety testing process [7].

Lynn 2018, T. Hinninen, J. Kasurinen et al. They conducted a study to evaluate how software companies test their items and their follow-up process models. Online form of design techniques involves data collecting. They also contrasted the data acquired with our preliminary study to identify the changes in industry practices. As a result, organizations have moved to experimental automation and advanced test infrastructure, and even implemented more efficient processes for mission-critical software, and reduced the use of standard process models [8].

Afzal, W., Alone, S.et al. reported that the Software Test Improvement Method (STPI) is a framework for guiding Enhance their software testing procedure by software development firms. We utilized a systemic literature review (SLR) to identify current STPI approaches (e.g., development thoroughness and availability of information and assessment tools) and their features,

and restricted areas of access). In addition, two alternative methods (TPINEXT and TMMI) were evaluated for their content and the industry of the evaluation results. As a result of this research, we have determined the 18STPI method and its characteristics. The details of TFTPINEXT and TMMI are late. We discovered that many STPI techniques did not give sufficient information or did not offer assessment tools. This makes it hard to apply in the market in many ways. The main similarities found between TPINEXT and TMMI are some differences [9].

In 2016 Garousi, V., & Mäntylä, M. V Consider software test automation to be a solution for cost reduction and Reduction of cycle time in software development. However, automated testing might fail if unit testing is not introduced at the correct time, context and technique. When and what to automate judgments are essential since erroneous choices might result in disappointment and serious errors (resources and efforts). In 2016, Garousi, V and Mäntylä, M. V considered the automation of software testing to be a solution that might cut test costs and decrease development cycles for software. However, in the correct context and in the correct technique, the setting up of test cases may fail if the test automation is not done at the correct moment. It is extremely crucial to choose where and when to automate, since the incorrect selection might be frustrating and huge cost of error (resources and workload). In order to support the decision of when and where to achieve automation, since the advent of evidence-based automation technology, researchers and practitioners have put forward various heuristic guiding principles and factors. As the quantity of such resources rises, it is necessary that the newest practice be classified systemically and a complete overview is provided. To attain these aims, they performed a research on micro literature review (MLR) to see when and how software testing can be automated. MLR is a system literature review form (SLR). In addition to published (official) literature (such as journals and conferences), it also includes gray literature (such as blog articles and white papers). You examined academic material using Google Scholar and investigated present study deals using the Google ordinary search engine. Its MLR and its conclusions are based on 78 sources, 52 of which are grey papers, while 26 are official publications. They separate aspects that determine when and how to automate the issue into five categories using qualitative analysis (coding): (1) testing software factors (SUT), (2) testing related factors, (3) testing tools related Variables related, (4) staff and organizational variables and (5) cross-domain factors, etc. The most prevalent personal variables are: regression assessment (44 sources), economic circumstances (43) and maturity of the SUT (39)[10].

In 2017, Dias-Neto, A.C., Matalonga, S. and others reported the survey findings of practitioners from various nations in software testing practice. They studied these surveys in this study in order to resolve their main problems and replicated a poll of clinicians Uruguay and Brazil (the two software development programmers in South America). A total of 150 software testing practitioners responded to this poll. To enhance previous findings, verify its results with new software tests mentioned in the technological literature. Teams from Brazil and Uruguay noted: (1) The software testing experts useful and relevant information (plans, cases, procedure, results) of the testing facility; (2) System testing and linings are the most useful and relevant (3) Testing and test management tools and fault reports are regarded as important and useful [11].

In 2015, Fawad, M., Ghani, K. et al. found that Pakistan's software industry has grown very impressive in recent years. However, software development businesses must follow tight quality assurance standards in order to sustain this expansion and provide high-quality software. The major goal of this research was to examine existing quality assurance processes in the Pakistani software sector to identify improvements. Consider the key components of quality assurance software, including test methodologies and procedures, software test and tools, performance testing, quality indicators, training and training for quality assurance. Research findings suggest that the present utilization of activities connected to quality assurance is not prevalent. This applies in particular to small software development businesses. It is, nonetheless, encouraging that they take this approach very much [12].

In 2015, MANSOR, Z. and NDUDI, E. E. found that software testing activities are the main challenge in today's software development projects due to the needs of end users. End users need to complete the project in a short time with zero defects and high product quality. Therefore, testing activities should be started as early as possible, because it helps to correct major errors in the early stages of software development and reduce error tracking and reprocessing later. In addition, it is also a challenge for the testing team to clearly understand the purpose of the test and plan how, where and how to test the system. These problems make the software testing process very time-consuming, accompanied by various challenges, which stem from the inability of software testers to perform these challenges.

Complete tasks effectively. Therefore, they investigated the issues, challenges and best practices of software testing activities. Document analysis is done to analyze information. The survey results revealed 9 key issues and challenges in software testing activities. The survey also revealed 9 best practices. The survey results will help the software community, especially the testing team, determine the problems and challenges they may face, and how to perform the testing activities in the best way [13].

In 2016, Garousi, V. and Mäntylä, M. V. found that any novice or industrial practitioner may have difficulty digesting a large amount of Software testing expertise. All the information employed in business, education and research in an ideal world should be known for high evidence. Since choices must not be dependent on individual research, supplementary studies are crucial when data is provided. More than 101 grade school studies in the subject of software testing have been published since 1994, according to their study. Due to the large number of secondary school studies, this field must be reviewed to outline the field of software research. this field. The aim of this study is to develop secondary sources in software testing systematically (classify). It argued that third-level research may be utilized as an abstracted index for the recovery of the most significant research material and therefore enable evidence-driven decision-making in any particular sector of software engineering. [14]

In 2014, Federer (M.), Haisjackl (Haisjackl) et al. were in industry. Due to limited resources, testing had to be carried out under strong pressure. Risk-based testing employs risk to direct the testing process which may employ resources allocation and product risk reduction. Risk evaluation (i.e., risk identification, analysis and evaluation) establishes the relevance of the value of the risk

set in the test and hence defines the quality of the whole risk-based process. They propose a paradigm for risk assessment and integrate it with the current testing procedure. The concept is based on best practises based on risk-based evidence based on techniques of public use and applied to industrial tests [15].

In 2015, Jiang M. and Hassan, A. E. Many large software systems had to handle thousands of simultaneous requests. These systems should be load tested to guarantee they can usually function under load (that is, input the required rate). This study will be of great benefit to load test specialists and software engineering academics interested in big software systems load testing [16].

Garousi, V., Escandar, MM et al. reported in this article will describe the characteristics of collaborative projects of numerous industry R&D colleges in the field of software testing conducted by authors (located in Canada and Turkey), which include the latest system literature The review study identified a number of challenges, models and counter-models in order to contribute to the evidence in the field of IAC so that SE researchers and practitioners can benefit from carrying out the projects of the successful. Generally speaking, The IAC is charged for software engineering and program testing. To fulfil the aforementioned objectives, 10 IAC projects (six completed, two failure and two permanent) were chosen, all in the realm of software testing, where the author has served as a leader or played an active role. Research objects and performs quantitative and qualitative analyzes on selected challenges, models and counter-models. As a result, A number of empirical results and scene-based suggestions are presented in the research. It was noted, for example, that while the IAC project looks to be ideal in many ways, it faces a major challenge (such as a divergence in a confidentiality agreement) that could lead to failure [17].

In 2017, Federer (M.) and Ramler (Ramler, R.) have disclosed a lengthy and outstanding research history on the static and dynamic assessment of software systems, with a concentration on academic and scientific locations. The condition of academic research and the condition and demands of industry practice still remain apart. Bridging this gap is of shared interest, since collaboration between academic and industrial will promote industrial development and innovation and assure industry's relevance to academic research. This feature article focuses on software testing collaboration between business and academics. A number of events connected to collaboration between industry and universities follow, such as seminars in testing: academia, industry, practice and technology of research [18].

In 2018, Maher Z.A, Shaikh H et al. investigated the variables that impact the application of software development methods in the industry. Secondly, this research is founded upon a Unified Model of Acceptance of Technology and Technology Use 2 (UTAUT2), a model to explore the variables influencing developers of software that follow appropriate software development practices [19].

In 2019, with the rapid growth in analytics and big data, the report produced by Gao J.et al. in Tao C., used more and more software services to create smart functions, including computer vision, advice, decision-making, predictions and intelligent decisions based auf its multimedia input. However, increasing quality problems have contributed to inaccurate test prices for companies and companies. Existing work hardly examines how quality control and record keeping on AI software

may be performed. This paper concentrates on AI software quality certification aspects. This article gives us an insight of AI software testing's new features and needs. Moreover, the current test categories for AI software are described and several test methodologies are explored. In addition, it also explains the evaluation of the test quality and the analysis of the standards. In addition, a practical research on the quality evaluation of the image recognition system is carried out through the deterioration test method. The research results show the feasibility and effectiveness of this method [20].

In 2015 Jiang, Mr. M., & Hassan, A. E. observed that hundreds or millions of large-scale software systems had to service simultaneous queries. These systems should be load tested to verify that they work under stress correctly (i.e., the rate of input requirements). In this work, they investigate the status of research and load testing practice. In three steps of a load test, they analyze and contrast existing techniques: (1) the design of an appropriate load, (2) the performance of a load test, and (3) the analysis of the load test data. This paper is valuable for practitioners and software engineers interested in the load testing of large buildings [21].

In 2014 Sultana, S., Motla et al. reported that the software sector in Pakistan might potentially play an essential role in boosting a stagnant economy. A well-organized framework adapted to the demands of the industry contributes to the development of high-quality goods in budget and schedule. Identify major difficulties in the Pakistani software industry which are considered to hinder the achievement of worldwide quality requirements, incorporate agile principles in order to tackle different management, quality and technical problems and design an appropriate framework. The numerous issues of Pakistan's software sector are highlighted. Several hybrid models are examined in order to assess their strengths and limitations. In conclusion, case studies and expert evaluations are carried out to check the efficacy of the software sector in Pakistan. Our hybrid pattern [22].

In 2015, Mailewa, A., Herath, J., and Herath, S. reported that testing has become expensive due to the time spent. This is because many combinations of functions, completeness, performance, etc. need to be tested. These combinations might be referred to as test cases. The objective of software businesses is to shorten test time to save money and provide consumers with goods more quickly. Two basic approaches to shorten test time are available. Firstly, the number of analytical cases is reduced; secondly, the region of repeated tests is automated. This paper discusses the fundamental principles of testing such as the value and distinction between testing and validation, the newest innovations in agile software testing, testing of the whole SDLC and methodologies, levels and types. Finally, as a recap, the tests will give some ideas for combined tests to lessen the issue of lengthy test times in order to minimize the amount of test cases [23].

In 2017, a total of 150 software testing practitioners reported to Dias-Neto, A.C in the United States. , Matarunga, S. Salili conducted an investigation. Compare the results with previous implementations & additional surveys discovered in the research papers to enhance past results. Participants from Brazil and Uruguay pointed out: (1) The documentation (plans, cases, procedures, results) of the test facility is useful and relevant to software testing practitioners; (2) System testing and regression testing are the most useful and most useful Important (3) The task

of monitoring and managing the testing process and tools for reporting defects are considered useful and important; (4) Software companies usually define the testing process and have a dedicated testing team; (5) Lack of support to measure the industry Testing and coverage tasks; (6) Tools that support the creation, execution or coverage of trial code are rarely used in their organization [24].

In 2015, Deak, A., Stålhane, T., and Sindre, G. Software testing has been noted as a vital to producing a successful and stable program product or service, however testing is typically seen as not exciting compared to design or coding. Like any human endeavor, the outcome of the ultimate output of software relies on human elements. One fundamental difficulty facing software development companies is to identify effective approaches to boost the motivation and work happiness of testing professionals. Their research aims to understand the motivation of professional software testers and to investigate policies and processes defined and implemented in projects. This article presents the results of empirical study based on super, detailed interviews with 36 physicians from 12 companies in Norway. The data collection performed a two-year study to explore the company's tactics for stimulating testers, while also considering the motivations and motives that affect testers [25].

Royal j j. , Kothapallii, the first computer product development company is listed as the main driving force of overall economic growth in developing countries. This provides professionals with quality-of-life incentives to encourage them to accept technological advancement and have the potential to provide jobs for more people. This article aims to illustrate the best management practices based on qualitative and quantitative statistical analysis, and propose a practical method to help software product development companies enter the market. Some of the recommended best practices are effective communication between all participating teams in the software product development process, the use of regression testing, smoke test methods and common sense to verify business recommendations, continuous innovation and growth of features, and market competition [26].

In 2018, Maqbool, B., Rehman et al. S. said that study focused on the five key aspects of software testing, defining the methodologies of software testing and the signs of software testing. Standards, automated tools for testing and training and testing. The study evaluates the applicability of current software testing procedures, provides some ideas and views on the future of application development in Pakistan's IT sector based on the survey findings, and proposed solutions on how to ensure the quality of agile software development. Use different factors [27].

Hynninen, T., Kasurinen et al. The costliest jobs in the software life cycle are quality testing and assurance. However, various studies have also shown that the industry may improve quality and reduce cost by expenditure on development of test techniques. Computer programming educators may bridge the gap in both formal training and industrial practice by following advancements in industry and fostering more graduates in industry. In order to understand the market today, we carried out a study of software companies to analyze how products are tested and which production models they use. According to the findings of the study, firms depend extensively on test

automation and employ extensive testing facilities. Even in mission-critical software, they use agile approaches and eliminate formal recommendations And techniques of evaluation. This document presents various critical learning goals which the industry needs college graduates to fulfil in performance management and software testing. The constructive alignment idea is utilized for proposing learning objectives, instructional techniques and evaluation methodologies that satisfy the needs of industry [28].

The sorts of software engineering methods and technologies utilized in the sector are vital to grasp. Each software team and firm have a variety of kinds and maturities of software engineering methods. Many studies were carried out in various nations and Areas for defining computer science process variants in software firms. The Turkish software company is in the process of creating and understanding the condition of the software development. Their objective is to define and understand the high-level viewpoint of software engineering in Turkey. The study involves procedures of computer programming, design, development, testing, upkeep, configuration management, release planning and technical support. The most recent research ever conducted in the Turkish software business. They conducted an online survey with 46 questions based on our past experience in the Canadian and Turkish environment and in the Software Engineering Knowledge Organization to achieve these goals (SWEBOK). The poll was attended by 202 in-service software engineers in the Turkey software sector. They investigated and described in this article the outcomes of the difficulties. Where applicable, the trends and outcomes of the survey are also compared to the findings of previous research in the Canadian software sector in 2010 [29].

In 2019 Jahan, M. S., Riaz, M. T. et al reported the fundamental Software testing is the product testing activity and a highly integral ingredient of the software development. The quality of any system is known by testing this product. Over a time with the expansion of the IT industry, several new testing and validation procedures are produced. New trends in developed countries like the United States, the United Kingdom etc. are generally welcomed. However, IT is not as good in impoverished countries such as Pakistan. Due to several contracted projects in Pakistan as well as worldwide IT companies. Due to increasing demand for outsourced projects, the software testing is becoming a key part of Pakistan's IT sector. Their team acquired test data in Pakistan's IT industry that will aid the IT industry testers not only in Pakistan, but globally. Researchers may utilize these data to determine the strengths and weaknesses of IT testing [30].

Bhuiyan, S. A. R., Rahim, M. S., et al. have found both obstacles and practices in Bangladesh of SQAT operations. The survey was produced utilizing qualitative approaches for research. The survey was place from August 2017 to January 2018. A total of 47 organizations took part in the poll. Most of the software companies included in the study were Dhaka, the IT industry center in Bangladesh. The survey was established based on four major goals: to characterize businesses, to identify SQAT practices, to capture the essence of the test team, test instruments and to understand the problems of SQAT, training, learning and career prospects. This report discusses the outcomes of the survey and gives recommendations for academics as well as industry for further study. [31]

Petersen, K. et al. Engström, this article discusses the design and assessment of SERP-test, a taxonomy meant to enhance communication between software testing academicians and policymakers. SERP exams may be utilized for direct contact in industrial academic collaboration. It may also facilitate indirect communication between researchers in software engineering and scientists who seek commercial relevance. The SERP exam was built on a methodical and goal-based approach including literature studies and interviews with professionals and academicians. [32].

In 2015 Alégroth, E., Feldt, R., et al. High-quality testing, such as graphical interface (GUI) and acceptance testing is usually accomplished utilizing sometimes costly manual processes, repetitive and error-prone in the software development business today. Test automation has been recommended to overcome these challenges, although testing from either a low system - level abstraction in most automation systems has been attempted. Their suitability for high-level evaluations was thus challenged. However, research suggest that such techniques suffer from limits like GUI responsiveness or compiler optimizations, system dependence, etc. High level test automation options such as registration and replay exist. Visual GUI (VGT) testing is an emerging industrial process with a perception that is more flexible and resilient to certain GUI changes compared to the previous high-level automation (GUI) test approaches. The heart of the VGT is the image recognition utilized to assess and interface with the front-end bitmap layer of the system. Vgt tools can simulate the behavior of end-users on almost any GUI based system regardless of implementing language, system software or platform [33] by combining image acknowledgement with test scripts.

Bin Ali, N., Engström, E. et al. 2019. Various strategies were presented to minimize There is, however, no research on how the industry usefulness and applicability of these methodologies are judged as the quantity of test cases to be carried out in regression tests. They have systematically examined literature for the following two purposes: enabling students to create and present regression tests with a focus on advanced manufacturing relevance and applicability; and second, to enable industry to adopt such research in order to address concerns from the practitioners' perspective. We discovered 1068 publications on regression testing using a reference-based search method. It restricted the scope to include articles with clear debates on relevance and application (i.e. mostly industry stakeholders studies)[34].

Anwer, F., Nazir, M. et al. 2017. The widespread nature and multimedia aspects, the dynamic and responsive behavior, the development of third-party products and the rapidly changing versions of these products might offer security dangers. The current investigation shows that the vulnerability of security to financial applications has increased significantly and continues to expand. It is thus crucial that computer software be safer and more reliable. Security tests disclose the vulnerabilities that may violate the integrity, validity and logical correctness of the state and the angles of attack vectors that take use of such flaws[35].

Florea, R., & Strecke, V. The purpose of this research is to create a test profile by empirically examining the competencies the company now needs. Data were assessed from 400 job ads in 33 countries. They mapped the capabilities to encompass testing, computing and domain

competencies in a taxonomy. They also examined the necessity for education, adequate qualifications and requirements for past experience. Its findings reveal that mainly employers concentrate on test and design skills, test automation, test automation, test environment and reporting. A third of the job advertisers were interested in those who are able to execute test implements [36].

Ilyas, M., Khan, S. U., reported in 2020 Software complexity increases, as contemporary companies rely heavily on large-scale software-intensive systems that work more and more in a continuously available environment. Global software developers (GSDs) are struggling to decrease this complexity by breaking down the goal software product into several components which have been created in-house, outsourced or acquired as commercial off-shelf components (COTSs). These components are then merged into a final functional product [37].

Two Santos, J., Martins, L. E. G., reported in 2019 Software testing is a key step in guaranteeing quality, safety and dependability in safety-critical systems (SCS). Several writers have presented novel ways to improving safety testing procedures in light of current procedures that strive to enhance procedures and make a good contribution to developers of software. This article seeks to examine major techniques to the testing of requirements, notably in the context of SCS safety standards. They examined how these ideas were formed and how they contributed to academics and industry. They assessed the advantages and disadvantages of the techniques and how they linked to the collaborative work of demand engineers and test workers. A systematic literature review (SLR) was conducted, choosing 53 publications from 1990 to 2018. Their study has been done in accordance with the Kitchenham and Biolchini standards. The findings of this SLR show the new software and security system testing research [38].

Felderer, M., & Ramler, R. in 2016. Risk orientation in testing is a significant way of balancing quality, timing and software costs. Risk guidance may assist to concentrate testing effort on essential aspects of a software product particularly for small and medium-sized companies (SMEs) under significant competitive and economic pressure. Although there are various risk-based testing methodologies available, the subject has yet to be addressed in the context of SMEs, where risk is frequently linked to crucial business challenges. This paper fills the gap and analyses the risk orientation status in the testing procedures of SMEs. In addition, the condition of danger testing in SMEs is compared to the situation in big companies. Multiple case study involving 5 SMEs. A prior risk-based testing research in large companies is being utilized to explore the variations between diversity promotion in SMEs and big companies. Studies imply that a strong business focus, use of informal risk concepts, and application of expertise to reduce testing costs and timelines constitute important differences between risk-based testing in SMEs and larger businesses [39].

In 2014 Itkonen, J., & Mäntylä, M. V. It was shown that the testing of manual software is a commonly employed form of verification and validation, which, despite the improvements in test automation, will not vanish. Manual testing is carried out by many practitioners without or before

test cases, i.e. creative, experiential testing, and say it is more efficient than testing with details. In this document, 51 students were presented with a repeated experiment evaluating efficacy, Efficiency and por unity of testing and ET (TCT). The findings of the first study are confirmed: (1) the effectiveness of identification of defects between ET and TCT is unchanged; (2) the efficiency of ET is greater when less design effort is needed. Due to slight changes in trial design, they also suggest that greater time pressure might affect the efficacy of the TCT strategy than ET. [40].

In 2016 K presents different perspectives of being an integrated scientist in industry who contributes to the advancement of software processes. The effort on improving the process centered on helping firms migrate from planned procedures to agile and lean activities. They will explore the obstacles, critical behaviors, and associated advantages that have been identified as embedded researchers in close collaboration with industry. This experience report is based on examples from many published cases [41].

Mohacsi, S., et al. provide in 2019 insights from a major energy project. For many energy companies, the mandatory shift from analogue energy meters to smart metering technologies is a major problem. Besides technical challenges relating to meters and transmission technologies, internal business process adaptation in addition to developing backend software, it might be harder than intended. In software and system testing, the criticality, size and complexity of the studied project are represented in where the underestimated effort, errors and incorrect judgments have created major problems. We detail the test difficulties identified and the underlying reasons in our study. They next evaluate the detected flaws with a collection of well recognized tests and anti-patterns [42].

Review of Literature with description of Authors Name, Objectives, Search Techniques, Problem Statement and Result:

Authors [ref]	**Objective of the study**	**Search Techniques**	**Problem statement**	**Result**
Vahid Garousi (2014)	Regional test automation survey findings among practitioners	TLD,TDD,Junit	the majority of Canadian firms spent less than 40% of their efforts (budget and time) on testing during development	authors They have noted that the necessity of testing recently is paid greater attention by the Canadian sector.
Shanmuganathan(2018)	Study shows the use of an	TDD,UML	Which test tools are	The findings of this

	Empirical research carried performed by software businesses in Sri Lanka on the present situation of the practise of software testing.		automatic (e.g. - in-house tools created, commercial tools etc.)	research would allow the three main software firms to identify test software that is necessary for development of software
Saha,(2019)	The automated test tools make testing simpler and more user-friendly for test automation testers.	STMTS	This study will help industry experts to grasp the current position in Bangladesh	The software business has become crucial by using youngsters and using current tools and approaches to generate high quality. software.
T. Hynninen (2018)	The purpose of the survey was to study software testing industry practises. We have analysed software companies to evaluate how they test their products. And whatever types of	CMMI	Complicated testing tools cause test configuration errors.	The findings of the study provided in this article show that software testing procedures in the previous 8 years have experienced some significant modifications.

	process they follow.			
Muhammad Shah Jahan(2019)	offer information of the procedures utilised for testing software in Pakistan's IT industry,	GQM,JUNIT		Understand the characteristics of IT tests
S. M. Abdur Rouf Bhuiyan(2018)	The objective of this study was to examine the obstacles and practises of SQAT in Bangladesh.	JIRA,JUNIT		the survey results carried out in Bangladesh to discover agile development and testing techniques and difficulties.

Table 1. Review of Literature with description of Authors Name, Objectives, Search Techniques, Problem Statement and Result

3 Software testing:

3.1 What is software Testing?

Software testing refers to the software evaluation procedure with the purpose of finding errors. Software testing is a technology that aims to evaluate a program or product's attributes or capabilities and to determine its quality. Software testing is also used to evaluate software for additional quality elements of software, such as dependability, usability, integrity, safety, capacity, efficiency, portability, maintenance, compatibility etc. [44] A competent test suite finds genuine defects. As the set of defects in a program is frequently unknown, this definition does not work for practitioners who produce test suites or for researchers who create and evaluate tools for the generation of test suites.[45]

3.2 Software Testing Process:

We have planned well structed research on Software Testing practices for service-oriented software development by proposing a well-defined questionnaire to determine the importance, significance, problems and practices to gain the objectives of research. It helps software Tester to better understand the problem. Participant involved in requirement engineering software engineers, Managers, Testers and users.

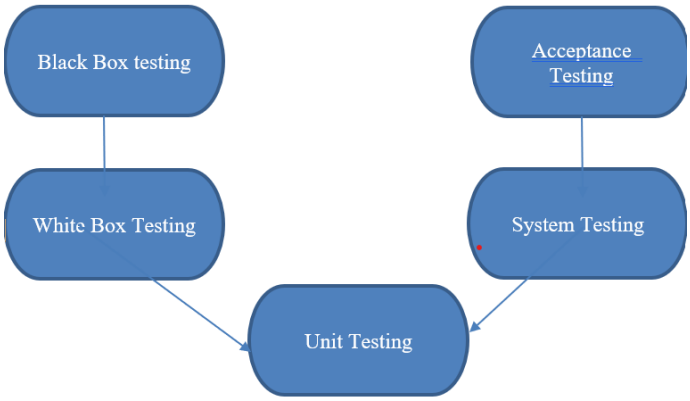

Figure 4. Software testing process

3.3 Optimize Software Testing Process:

The STLC stages find and correct errors using different software testing methodologies. This document covers the neces ıout which no software life cycle will be effectively completed. Primarıly, testıng provıdes a critique or a comparison which assesses the state conduct of the system against its specs, methods, principles, features and applicable

standards. The testing method of software may be adjusted to client or project requirements. The optimization method that you may apply during software testing is analyze, plan and prepare, execute and close [47] This article describes how testing is vital in the software development process as a whole. The whole cycle is known as the life cycle of software development (SDLC),in this process, software testing phase runs in parallel via a testing life cycle (STLC) through which software is tested step by step. Different techniques of testing are used on different phases of STLC [46]

- S1: Requirement Analysis
- S2: Test Preparation.
- S3: Developing Test Cases
- S4: Setup of the Test Environment
- S5: Execution of Tests
- S6: End of the Test Cycle

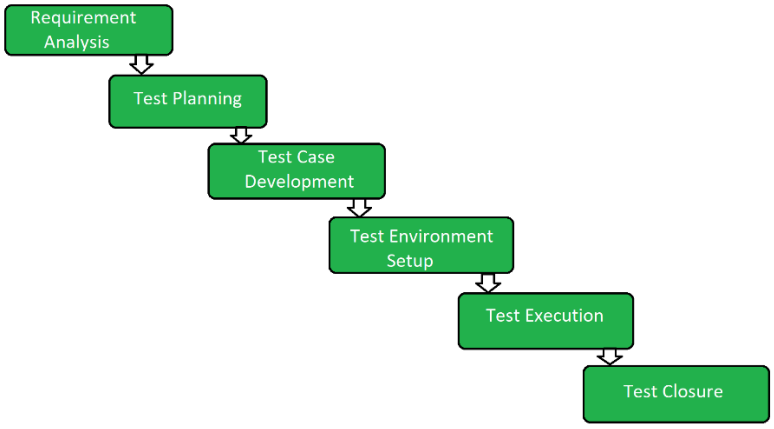

Figure 5. Optimize software testing

3.4 Software Testing Research Roadmap:

A roadmap offers the way from the red dot "you are here" to a desired location. The research plan for software testing is structured as follows: • the red point of the selected destination is a set of (four) dreams: we use this term to indicated that these are exponential goals at the end of four identified research advance paths. · The red dot "you are there" consists of the highest achievements of past research (but note that certain efforts are still in progress). By definition, they are unachievable and their worth remains precisely as the attractive pole for valuable, far-sighted research; • in the center, present and future test research faces problems, at a more or less mature level and with more or less prospects for success. These obstacles are the directions to be taken on the path to the dreams, and they form the key portion of the plan. We have placed the new and continuing avenues for study in the middle with more mature themes, successes at the left and the

ultimate aims, the aspirations, at the right. Four horizontal strips show the paths of dream research found, namely:

- Universal testing theory
- test-based modelling;
- 100 percent automated testing,
- The paths are arranged as follows:

the theory is at the base of the accepted models, which are in turn necessary for automation, which is essential for cost efficient testing engineering. The horizontal difficulties cover six vertical strips correlating to the WHY and HOW of the software testing confronts. WHY and WHEN (in no specific order). The problems of software testing are found in this plan, based vertically on the long-term aspirations to which they are mostly aiming, and horizontally on which question or issues of the software testing characterization they mostly focus. In the remaining article, the features (realizations, problems, aspirations) of this plan will be discussed. This plan is commonly compared with its 2000 Harrold predecessor which we shall call FOSE2000 in the future.

4 REAEARCH METHODOLOGY

This chapter covered the following parameter of study by using a model which is consist of
1. Questionnaire

2. Online survey

3. Survey result in statistical form (Pie charts/tables/Graphs/Histograms etc.)

5. Propose customized model of software testing for SOSE

6. A case study of service-oriented software testing

The research methodology model is given below

4.1 QUENTIONARE:

White Box Testing:

1) Do you check that checking white boxes might be fairly complex?
2) Do you confirm if a software program is intended for white box testing?
3) Check that White Box test scenarios can be automated easily?
4) Are you looking for hidden mistakes to test the code optimization?
5) Make sure you can start testing early in the SDLC even if there is no GUI available?

Black Box Testing:

1) Do you make sure that the system's needs and specifications are checked first?
2) Do you ensure that the software tester creates test cases using the inputs you've chosen?
3) Do you agree that black box testing makes it easier to test module communication?
4) Do you make sure that the causes and consequences of a decision table are mapped out in a matrix?
5) Do you make certain that the emphasis of black-box testing is on the product?

Unit Testing:

1) Do you make sure that good unit tests act as documentation for your project?
2) Do you make sure that all unit tests aid in the early detection of problems and cost savings during the development life cycle?
3) Do you make sure that unit testing helps programmers to fine-tune their code and confirm that the module functions correctly?
4) Do you make sure that unit testing aids developers in comprehending the testing code base and allowing them to make modifications as needed?

Integration Testing:

1) Make sure you have an adequate architecture / technology design document that properly defines the relationships between each unit?

2) Do you make sure that every unit is tested before you start testing for integration?
3) Do you make sure you have a comprehensive management system for software configuration?
4) Do you ensure that interface testing is performed after or in tandem with system testing?
5) Do you ensure that without integration testing system tests will be time-consuming?

System testing:

1) Do you check that the end-to-end system is to be evaluated?
2) Do you check that the system test verifies the whole software product, which is completely integrated?
3) Do system tests validate the user's testing of the application?
4) Do you verify that system testing on the final software product is done by a quality laboratory agent before it is launched into the market?
5) Do you check that in system testing Time available for testing?

Acceptance Testing:

1) Do you check that the Acceptance test approach is defined?
2) Do you check that environments are sorted?
3) Do you verify that output is generated?
4) Do you verify that relevant test data is identified?
5) Do you check that all bugs should be fixed?

4.2 Online survey:

A survey is a powerful tool to research the best techniques and problems in previous and present time study. The research was conducted to collect the data from different organization such as software development companies, user customers, different working companies to equated the software testing practices, maintenance, according to their requirements. A research methodology was use to judge the software testing; a questionnaire was used to get response from different IT organizations through a survey. In Pakistan a few organizations follow the software testing process. We conducted interviews, hard copy and google forms. The main goal of this survey to indicate the test practices in customization of software testers which are used in IT industry. A survey conducted digitally in which Service oriented Software engineers, Software Quality Assurance, Software Quality Engineer, Developers, PHP Developers, MS IT Students, IT Lecturer, IT Executive, IT Manager, Web Developers, Network Engineer and Software Houses showed their freedom to choose the best quality option related service-oriented software engineering showed in graph and online conducted more than seventy peoples to filled the questionnaire through google docs. Different results were allocated due to survey in Pakistan. The research will be helpful to indicate the weakness and strengths and also boost IT industry. Respondent’s result showed in After collecting the data of questionnaire, the statistical analysis was use to get the result.

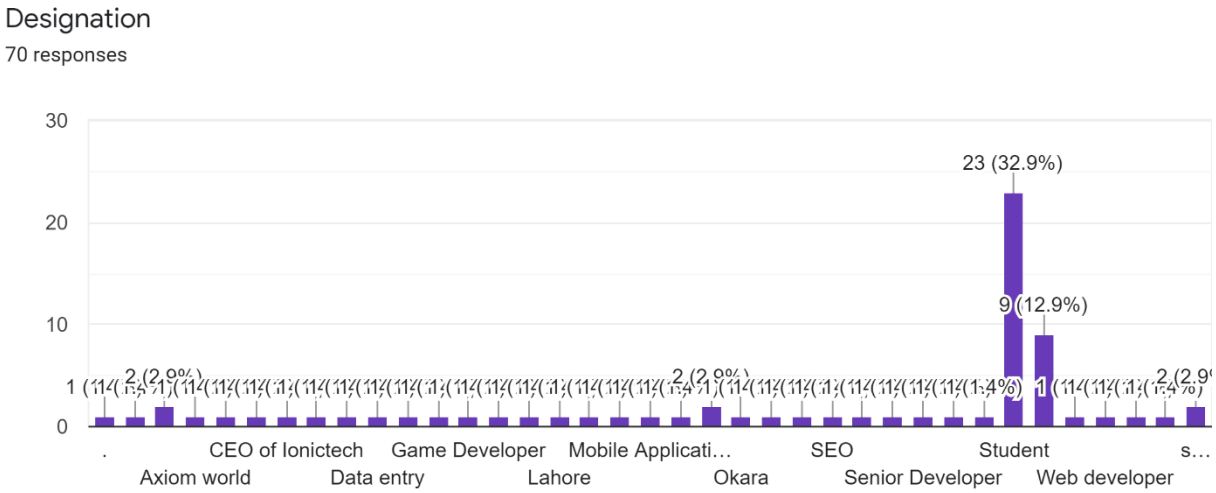

Figure 6 Designation

4.3 SURVEY RESULT IN STATISTICAL FORM(PIECHARTS/TABLES/GRAPHS/HISTOGRAMS ETC.)

Result is major part of research which help to identify the problem and recommendation. The pie graphs of all questions were obtained after online survey. A pie graph is a special chart uses pie slices to determine relative size of given data. Result obtained from online survey showed in detail in chapter result and discussion.

4.4 PROPOSE CUSTOMIZED MODEL OF SOFTWARE TESTING

service-oriented software testing process model is given below that exists the finding and explored the standards which specify the working and behaviour of model.[48]. The model explains the different activities such as;

- Black box testing
- White box testing
- Unit testing
- System testing
- Acceptance testing

4.5 A case Study Of service-oriented software testing

Software testing has long been acknowledged as one of the most expensive and dangerous processes involved in the development and maintenance of any software system. Because of the significant changes that this architectural style brings about in both the system and the software business/organizational models, adequately testing a system becomes more difficult with SOA.

The inherent dynamic nature of SOA, as well as the clear separation of responsibilities between consumers, suppliers, owners, and developers of a service and the piece of software that underpins it, present significant testing issues. In particular, as a result of automated service discovery and ultra-late binding, the complete configuration of a system is known only at the time of execution, which makes integration testing more difficult. Similarly, per-use-charge of a service has an impact on unit testing and Quality of Service testing of services and their compositions, among other things. Despite the fact that SOA testing is a relatively new field of study, numerous contributions have been made to the literature, primarily in the areas of unit testing of services and orchestrations, integration testing, regression testing, and testing of non-functional properties of services and orchestrations. There are also various studies in the literature that look into ways to increase the testability of services and service-centric systems. Nonetheless, numerous issues remain unresolved, necessitating further investigation: The use of a combination of testing and run-time verification It is necessary to supplement testing-based validation with runtime verification due to the dynamic nature of SOA due to its dynamic nature. On the one hand, testing is unable to cope with certain aspects of a service-centric system validation, primarily because it is impossible to test all—often unexpected—system configurations. On the other hand, testing is capable of dealing with certain aspects of a service-centric system validation. Although capable of dealing with the inherent dynamic and adaptability of SOA, run-time monitoring is unable to provide users confidence that their system will behave as expected before it is actually deployed into the wild. Further study is required to thoroughly appreciate the function of testing and monitoring in the validation of a service-centric system, as well as to create systematic methods for combining them with the goal of boosting confidence while decreasing the cost of validation Improving the testability of the system.

For system integrators and, in general, users, a service is simply an interface, which makes it difficult to implement standard white-box coverage methodologies. Many services, such as booking a restaurant table, have persistent impacts in the real world, making stress testing methodologies infeasible. Even worse, many services have persistent repercussions in the actual world, such as booking a restaurant table. In order to solve the lack of observability and the expense of repeated invocations, it may be necessary to publish a (state-full) model of the service and provide a testing interface that allows users to query and alter the state of the service without having to touch the real world. Additional research is required in order to develop the appropriate formalisms for expressing the models and to assist in the standardization of the interfaces. Because developing models is time-consuming and error-prone, methods for reverse engineering them from observations of service behavior must be developed System validation for completely decentralized systems. Today, orchestration is the most widely used approach to service composition. It entails the use of an engine that executes a process description and coordinates the invocation of services in order to create a cohesive service composition. Systems that use this method are intrinsically distributed in nature. Services are provided by administrative domains that are separate from one another Control, on the other hand, remains centralized. Nowadays, new types of compositions are emerging, such as peer-to-peer choreography, which entirely

decentralizes control as well as services, allowing for a more decentralized control structure. In addition to opening up new possibilities, fully decentralized compositions introduce additional problems to testing and monitoring, such as the propagation of a query via a network of active services that may subscribe to it based on an introspective knowledge of their capabilities. SOA has tremendous potential for lowering the costs and hazards associated with enterprise systems, increasing the efficiency and agility of businesses, and mitigating the impact of change. Many of the benefits of SOA, on the other hand, become challenges when it comes to testing services and service-centric systems. In order to meet these challenges, a coordinated combination of testing, run-time monitoring, and exception management is required.

4.6 White box testing:

White Box Testing is a software testing approach where software internal structure, architecture and code are checked for input flow and design, readability and security. We perform a questionnaire in software testing to assess the significance of software testing for providers engineering. Various questions like as were intended

4.6.1 Do you check that checking white boxes might be fairly complex?

This study is connected to white box in software tests, where testing may be understanded by 71 individuals in white boxes where 50.7% of the respondents agreed to testing white boxes might be fairly complex, but 22.1% less so. The result is shown in figure 7.

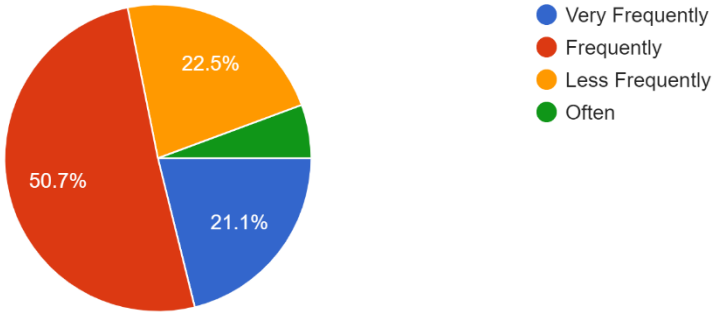

Figure7. White box fairly complex

4.6.2 Do you confirm if a software program is intended for white box testing?

This poll is connected to software white box testing, where White Box testing is to be performed on the software application of 71 persons, where 50.7 percent frequently replied to software program intended for white box testing but 22.5 percent less frequently did so on a software application.

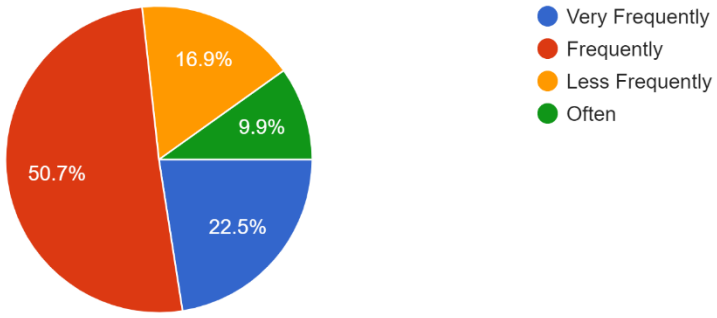

Figure 7. software program intended for white boxes

4.6.3 Check that White Box test scenarios can be automated easily?

This survey relates to the testing of white boxes in software testing in which situations of white box may be automated by 71 individuals 53.5% replied frequently White Box test scenarios can be automated easy, but often 7%. The result is shown In figure 9.

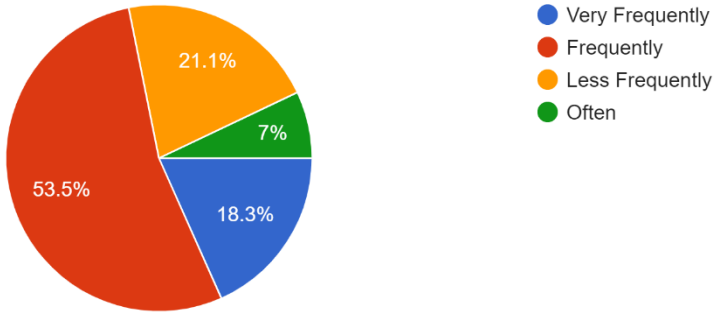

Figure 8. Test Scenarios Automation

4.6.4 Are you looking for hidden mistakes to test the code optimization?

This survey relates to the testing of white boxes in software testing in which situations of white box may be automated by 71 individuals 42.3% replied to the Code Optimization regularly by detecting hidden mistakes but 25.4% less often. The result is displayed Total replies in Figure 10 were 71.

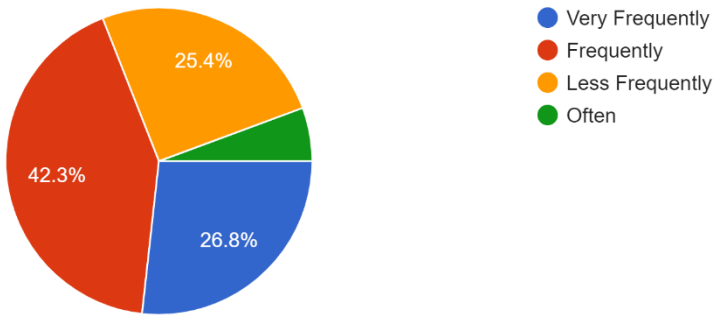

Figure 9. Code Optimization Test

4.6.5 Make sure you can start testing early in the SDLC even if there is no GUI available?

This study refers to white box in software testing where testing may commence early in the SDLC, even if 71 people can't understand the GUI when testing in the SDLC may start early even if the GUI is not available but is 7% often. This survey was indeed related to white box testing. The result is displayed Total replies in Figure 11 were 71.

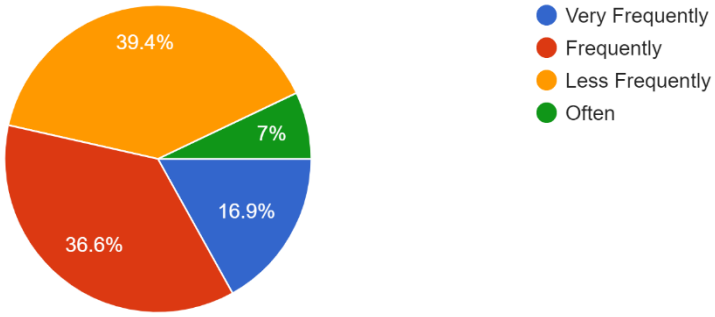

Figure 10. Software testing in SDLC

4.7 Black box testing:

It is described as a testing approach in which the function of the Application Under Test is examined. In software testing we conduct a questionnaire to evaluate the importance of software testing for services-oriented engineering. Different questions were planned such as

4.7.1 Do you make sure that the system's needs and specifications are checked first?

This poll is linked to Software Testing for the Black Box testing of the system's needs and specifications are checked first that is understood by 71 people, in which 47.9% of the people

often agreed to system requirements first and requirements are examined in black box testing but 12.7% less often. The figure 12 shows a total of 71 replies.

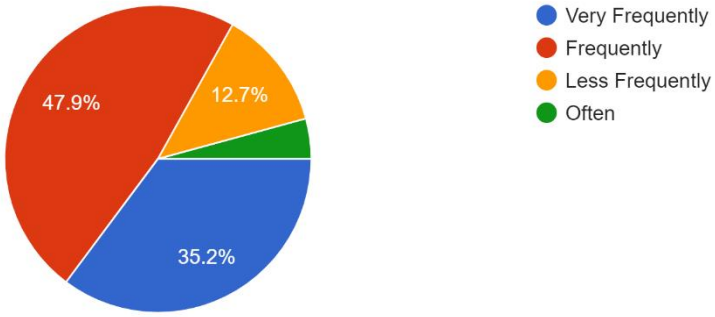

Figure 11. System needs and specification first

4.7.2 Do you ensure that the software tester creates test cases using the inputs you've chosen?

This research is related to testing Black Box in software, where Black Box testing should occur in a software system Testing which has 71 persons. 62% of individuals agreed that the software tester creates test cases using the inputs but 9.9% less often. The result is shown in figure 13 total responses were 71.

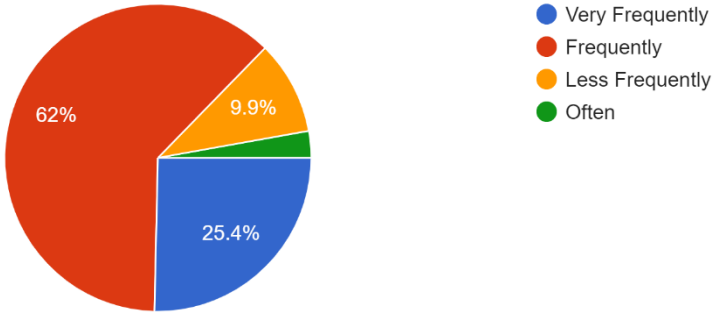

Figure 12. Software tester creates test cases

4.7.3 Do you agree that black box testing makes it easier to test module communication?

This poll is related to Black Box testing in product testing in which 71 persons execute Black Box testing with a software application in which 52.9% answered often, while 27.1% less often than that black box testing makes it easier to test module communication The results indicate a total of 71 replies in Figure 14.

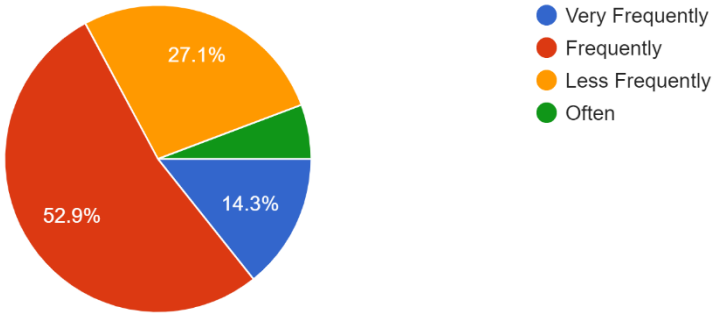

Figure 13.Test Module Communication

4.7.4 Do you make sure that the causes and consequences of a decision table are mapped out in a matrix?

This research concerns Blackbox software testing in a software system that 71 people comprehended and in which 56.3% typically indicated that a statement table puts reasons and its consequences into a matrix, while 15.5% less often. The figure 15 shows a total of 71 replies.

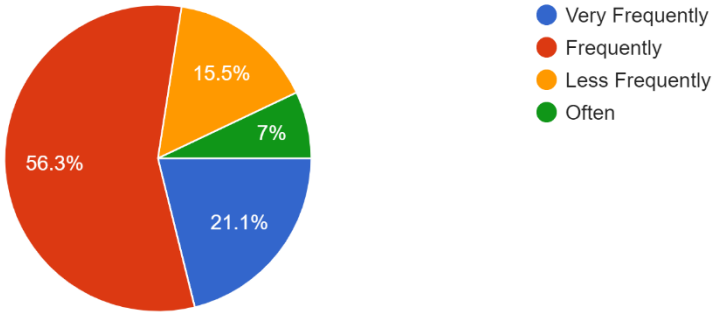

Figure 14.Consequence of decision table

4.7.5 Do you Ensure the focus of black-box testing is on the validation of your functional requirements?

This poll is connected with software testing Black Box testing where Black Box tests will be done on a software system understood by 72 individuals. 60.6% of individuals often stated that black-box testing focuses on validating their functional requirements, whereas 11.3% less often. The result is shown in figure 16 total responses were 71.

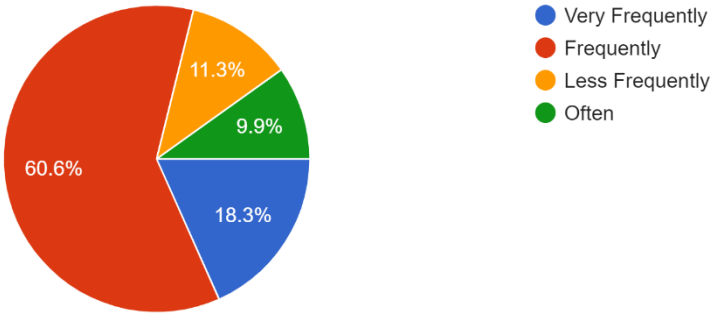

Figure 15 Validation of your functional Requirement

4.8 Unit testing

UNIT TESTING is a software test which tests each program component or unit. The goal is to check each unit of software code for its projected performance. Unit Testing takes place throughout the development (code phase) of a developer application. In software testing we conduct a questionnaire to evaluate the importance of software testing for services-oriented engineering. Different questions were planned such as

4.8.1 Do you make sure that good unit tests act as documentation for your project?

This study is connected to testing process in software tests where unit test act as a documentation of your project 52.1 percent often agreed that the excellent unit tests serve as construction documents but 38% very often. The result is shown in figure 17 total responses were 71

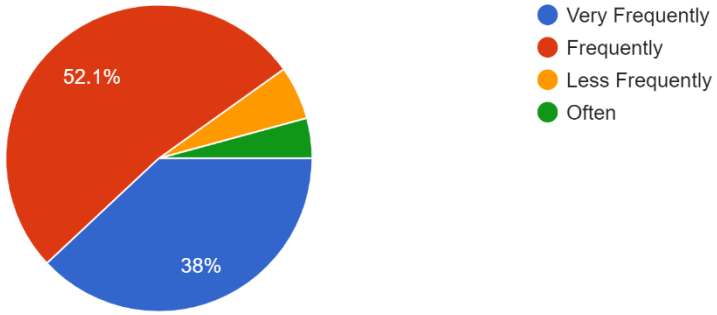

Figure 16. Unit Test documentation

4.8.2 Do you make sure that all unit tests aid in the early detection of problems and cost savings during the development life cycle?

This study is connected to testing process in software tests where unit testing should be performed on software applications understood by 71 people where 56.3% frequently answered the people agreed that all unit tests aid in the early detection of problems and cost savings during the development life cycle but 11.3% less frequently. The result is shown in figure 18 total responses were 71

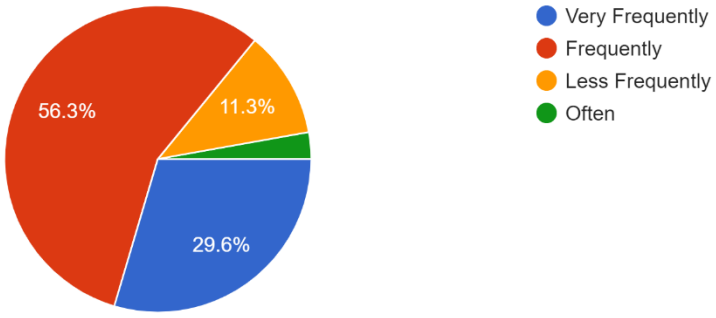

Figure 17 Early problem detection of problem and cost

4.8.3 Do you make sure that unit testing helps programmers to fine-tune their code and confirm that the module functions correctly?

This study is related to software unit testing in which the unit testing should take place on an user interface, which was understood by 71 people, where 52.1% of people often agreed that that unit testing helps programmers to fine-tune their code and confirm that the module functions correctly, but 7% less frequently. The result is shown in figure 13 total responses were 71.

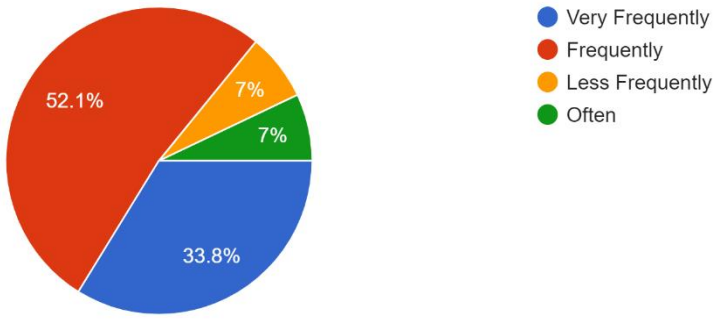

Figure 18. Module functions correction

4.8.4 Are you ensuring that Unit Testing allows developers to comprehend and make rapid changes to the test code base?

The poll relates to unit testing in software development where unit testing should be performed on a software application in which 33.8% have regularly said that the testing process helps developers understand the code base and allows them to make rapid changes. but 7 % less frequently. The result is shown in figure 20 total responses were 71.

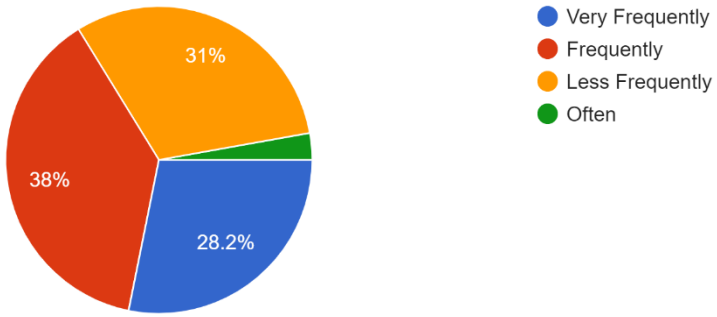

Figure 19 Rapid changes to test code

4.8.5 Do you make sure that you can test project portions without waiting for others to be finished through unit testing?

This study is connected to the unit testing in software tests in which the software testing should be done on a piece of software understood by 71 individuals in which 46.5% of people often agreed that unit testing allows you to test sections of the project before waiting for others but 15.5% less often. The result is shown in figure 21 total responses were 71.

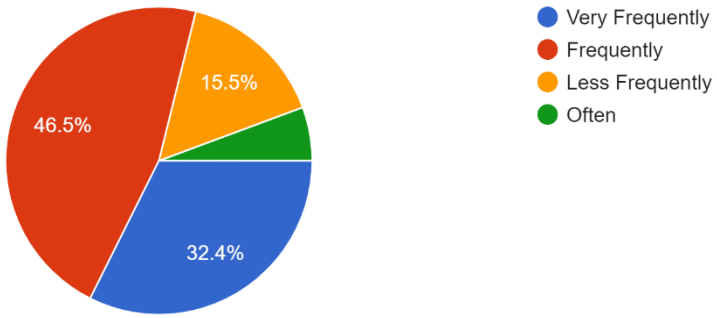

Figure 20 Test project portions

4.9 Integration testing

INTEGRATION TESTING is a level of software development that combines and tests separate units / components. This test level is intended to identify interaction defects between integrated components. We perform a questionnaire for software testing to assess the value of software development for services-oriented engineering. Different questions were planned such as

4.9.1 Make sure you have an adequate architecture / technology design document that properly defines the relationships between each unit?

The study concerns integration tests in software tests where software testing should be carried out on a software application. 50.92 percent replied frequently to people who agreed that a proper Architecture/Technical Design Document should contain a clearly defined but 7 percent fewer close discussions between each unit. The result is shown in figure 22 total responses were 71.

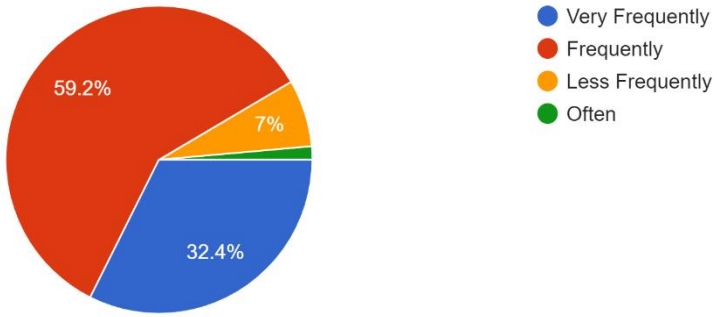

Figure 22Architecture Design document properly defines

4.9.2 Do you make sure that every unit is tested before you start testing for integration?

This poll is connected to software integration testing where software integration tests should be done on software program where 52.1% of people often agreed that each unit was tested before the integration testing started, but 11.3 percent less often. The result shows a total of 71 replies in Figure 23.

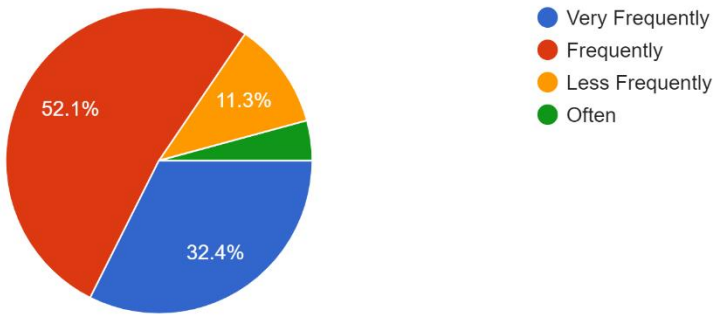

Figure 23. Unit is tested before you started integration testing

4.9.3 Do you make sure you have a comprehensive management system for software configuration?

This survey is connected to software integration testing where software testing should be performed on software application understood by 71 people where 59.2% frequently answered that you have a comprehensive management system for software configuration The result is shown in figure 24 total responses were 71

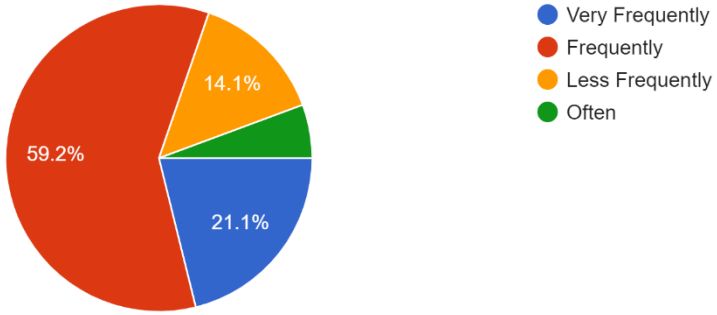

Figure 24. Comprehensive management system for software configuration

4.9.4 Do you ensure that integration testing is performed after or in tandem with system testing?

This survey concerns software integration testing in which integration testing should take place in software applications understood by 71 people, in which 56.3 percent frequently replied that integration testing should be done after testing process or in parallel with the software, but 18.3 percent less often. The result is shown in figure 25 total responses were 71

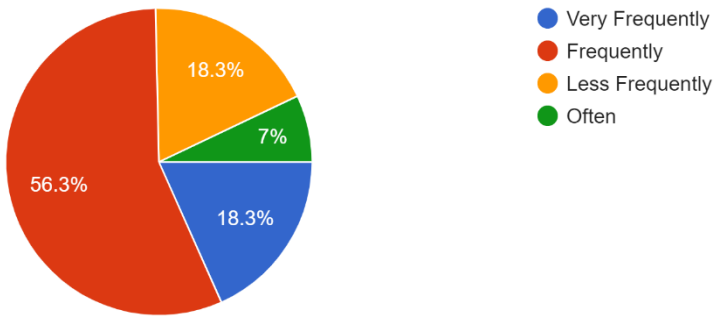

Figure 25 integration testing perform after in tandem with system testing

4.9.5 Do you ensure that without integration testing system tests will be time-consuming? This survey is related to Integration testing in software testing where conduct that Integration testing in software testing should be done on a software application understood by 71 people where 54.9% frequently answered the people agreed that without integration testing system tests will be time-consuming but 12.7% less frequently. The result is shown in figure 26 total responses were 71

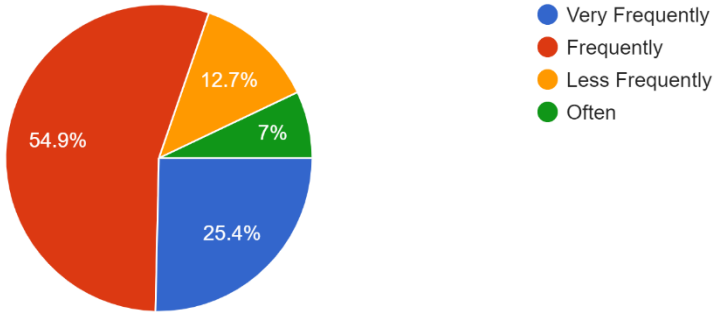

Figure 26. test time will be consuming

4.10 System testing:

SYSTEM TESTING is a software-level test that checks the whole product. The objective of a network test is to evaluate system requirements in every way. Usually, the software is merely one part of a larger computer system. In software testing we conduct a questionnaire to evaluate the importance of software testing for services-oriented engineering. Different questions were planned such as

4.10.1 Do you check that the end-to-end system is to be evaluated?

This poll covers software testing, where system testing should be performed in software development indeed be conducted in a software program understood by 71 individuals where 53.5% of the respondents agreed often that system testing should review the end-to-end system. but 14.1% less frequently. The result is shown in figure 27 total responses were 71

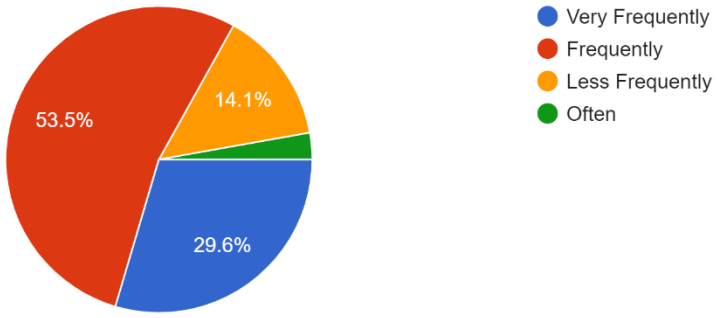

Figure27. End to end system evaluation

4.10.2 Do you check that the system test verifies the whole software product, which is completely integrated?

This study relates to software testing system testing in which system testing is carried out on a user interface understood by 71 individuals where 60.6% of people have often said that the system testing verifies the entire and completely integrated software product.t but 9.9% less frequently. The result is shown in figure 28 total responses were 71

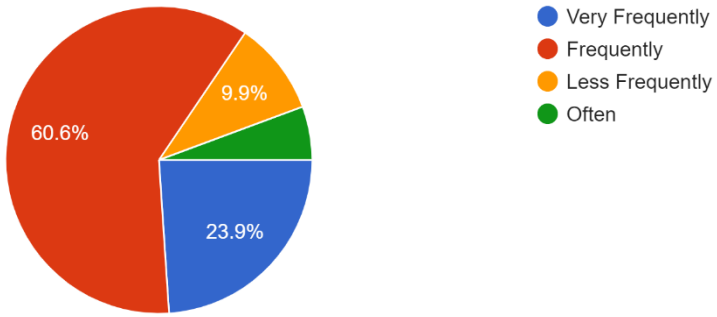

Figure 28. software product verification

4.10.3 Do system tests validate the user's testing of the application?

This poll is about software testing system tests where software testing should take place on a software application known by 71 individuals, where 54.9percent said regularly that

system tests validate the user's testing of the application but 9.9% less frequently. The result is shown in figure 29 total responses were 71

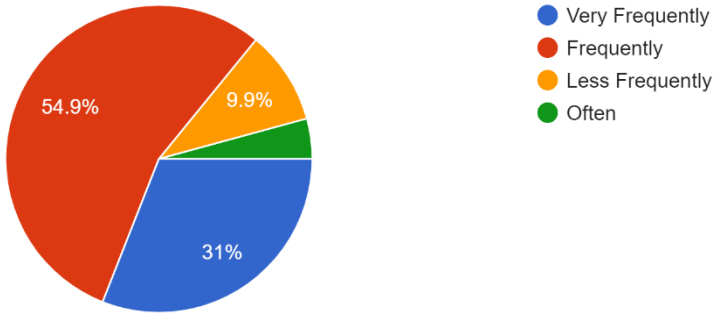

Figure29. Users testing of the application

4.10.4 4. Do you verify that system testing on the final software product is done by a quality laboratory agent before it is launched into the market?

This survey is concerned with software test system testing in which software testing is conducted on a software application that has been understood by 71 individuals, where 46.5% of respondents frequently agreed that the system testing done by a professional software testing agent is done before marketing, but 14.1% less often. The result is shown in figure 30 total responses were 71

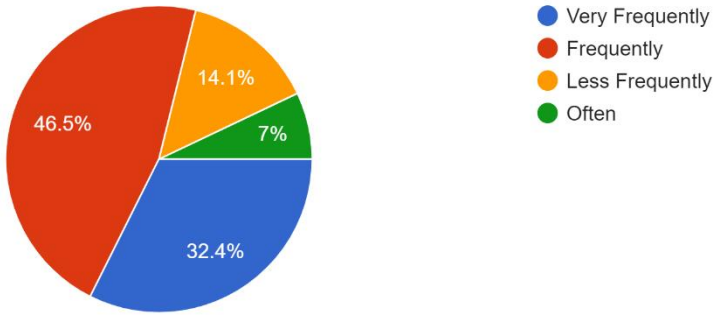

Figure 30 software product done by quality laboratory

4.10.5 Do you check that in system testing Time available for testing?

This survey is related to system testing in software testing where conduct that in system testing Time available for testing understood by 71 people where 54.9% frequently answered the people agreed that in system testing Time available for testing but 16.9% less frequently. The result is shown in figure 31 total responses were 71

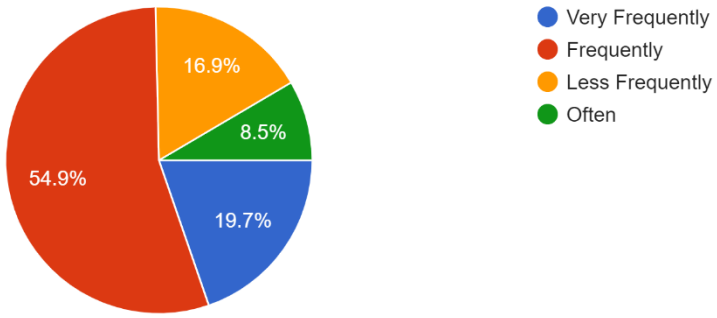

Figure 31.Testing time availability

4.11 Acceptance testing:

ACCEPTANCE TESTING is an acceptability system testing application development level. This test aims to check the company criteria of the system compliance and to determine if delivery is satisfactory. We perform a questionnaire for software testing to assess the value of project management for services -Engineering focused. Different questions were planned such as

4.11.1 Do you check that the Acceptance test approach is defined?

This survey is related to Acceptance testing in software testing where conduct that Acceptance test approach is defined understood by 71 people where 50.7% frequently answered the people agreed that the Acceptance test approach is defined but 11.3% less frequently. The result is shown in figure 32 total responses were 71

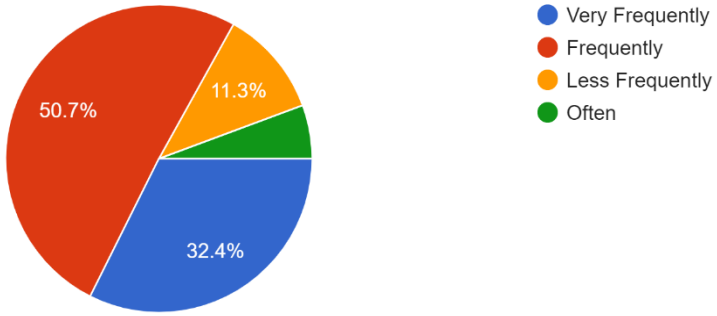

Figure32. Acceptance test Approach

4.11.2 Do you check that environments are sorted?

This survey is related to Acceptance testing in software testing where conduct that environments are sorted understood by 71 people where 59.2% frequently answered the people agreed that environments are sorted in acceptance testing but 12.7% less frequently. The result is shown in figure 33 total responses were 71

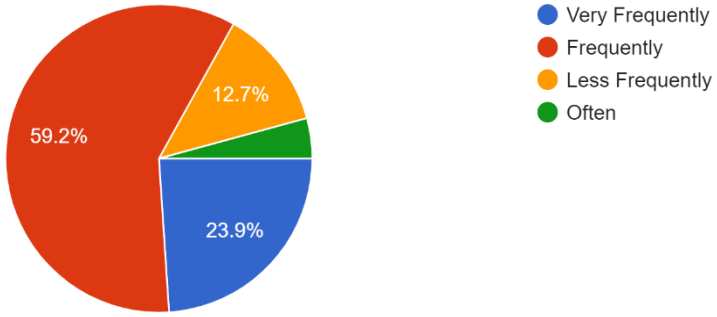

Figure 33 Environment are sorted

4.11.3 4.Do you verify that relevant test data is identified?

This survey is related to Acceptance testing in software testing where conduct that relevant test data is identified understood by 71 people where 39.4% frequently answered the people agreed that relevant test data is identified in acceptance testing but 12.7% less frequently. The result is shown in figure 33 total responses were 71.

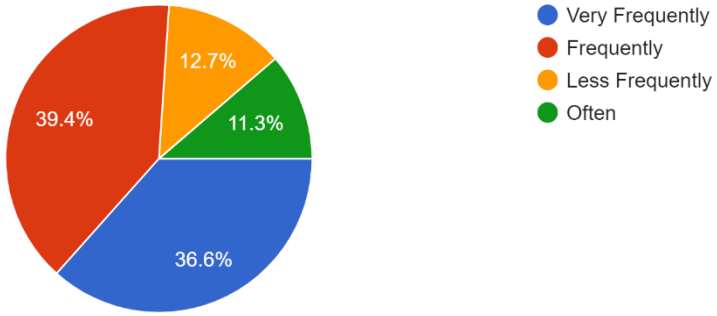

Figure33. Test data identification

Conclusion and Future work:

This Research shows that the software testing practices as a service Oriented Is necessary For Every Software Testing Company This Research shows that the software testing practices as a service Oriented Is necessary For Every Software Testing Company. We also study about service-oriented architecture Service oriented architecture software testing is a new field of study and have many contributions about the areas of unit testing, integration testing and regression testing. In this study we discus software testing practices in white box testing, black box testing, unit testing, integration testing concludes that:

Most of people have no idea about software testing practices during software testing behind some issues they have lack of knowledge about software testing as well as they have no collaboration with all of others software testers 21% not satisfied with white box test cases can automated easily but 50% mostly answered that in white box testing test cases can be automated easily.

Most of people face problem during the survey of black box testing that the black box testing is valid for the functional requirements 60% often stated that it is valid for functional requirements.

References:

[1] Garousi, V., & Zhi, J. (2013). A survey of software testing practices in Canada. *Journal of Systems and Software*, *86*(5), 1354-1376.

[2] Vasanthapriyan, S. (2018, July). A study of software testing practices in sri lankan software companies. In *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)* (pp. 339-344). IEEE.

[3] Lami, G., Biscoglio, I., & Falcini, F. (2016, June). An empirical study on software testing practices in automotive. In *International Conference on Software Process Improvement and Capability Determination* (pp. 301-315). Springer, Cham.

[4] Kochhar, P. S., Xia, X., & Lo, D. (2019, May). Practitioners' views on good software testing practices. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)* (pp. 61-70). IEEE.

[5] Saha, T., & Palit, R. (2019, December). Practices of Software Testing Techniques and Tools in Bangladesh Software Industry. In *2019 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)* (pp. 1-10). IEEE.

[6] Hynninen, T., Kasurinen, J., Knutas, A., & Taipale, O. (2018, May). Software testing: Survey of the industry practices. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 1449-1454). IEEE.

[7] Kassab, M. (2018, January). Testing Practices of Software in Safety Critical Systems: Industrial Survey. In *ICEIS (2)* (pp. 359-367).

[8] Hynninen, T., Kasurinen, J., Knutas, A., & Taipale, O. (2018, May). Software testing: Survey of the industry practices. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 1449-1454). IEEE.

[9] Afzal, W., Alone, S., Glocksien, K., & Torkar, R. (2016). Software test process improvement approaches: A systematic literature review and an industrial case study. *Journal of Systems and Software*, *111*, 1-33. Stoyanova, V., Petrova-Antonova, D., & Ilieva, S. (2013, March). Automation of test case generation and execution for testing web service orchestrations. In *2013 IEEE Seventh International Symposium on Service-Oriented System Engineering* (pp. 274-279). IEEE.

[10] Garousi, V., & Mäntylä, M. V. (2016). When and what to automate in software testing? A multi-vocal literature review. *Information and Software Technology*, *76*, 92-117.

[11] Dias-Neto, A. C., Matalonga, S., Solari, M., Robiolo, G., & Travassos, G. H. (2017). Toward the characterization of software testing practices in South America: looking at Brazil and Uruguay. *Software Quality Journal*, *25*(4), 1145-1183.

[12] Fawad, M., Ghani, K., Shafi, M., Khan, I. A., Khattak, M. I., & Ullah, N. (2015). Assessment of Quality Assurance practices in Pakistani Software Industry. *University of Engineering and Technology Taxila. Technical Journal*, *20*(2),89.

[13] MANSOR, Z., & NDUDI, E. E. (2015). Issues, Challenges and Best Practices of Software Testing Activity. In *Proc. 14th Conf. Appl. Comput. Eng.(ACE15), SouthKorea* (pp.42-47).

[14] Garousi, V., & Mäntylä, M. V. (2016). A systematic literature review of literature reviews in software testing. *Information and Software Technology*, *80*, 195-216.
[15] Felderer, M., Haisjackl, C., Pekar, V., & Breu, R. (2014, October). A risk assessment framework for software testing. In *International Symposium On Leveraging Applications of Formal Methods, Verification and Validation* (pp. 292-308). Springer, Berlin, Heidelberg.
[16] Jiang, Z. M., & Hassan, A. E. (2015). A survey on load testing of large-scale software systems. *IEEE Transactions on Software Engineering*, *41*(11), 1091-1118.
[17] Garousi, V., Eskandar, M. M., & Herkiloğlu, K. (2017). Industry–academia collaborations in software testing: experience and success stories from canada and turkey. *Software Quality Journal*, *25*(4),1091-1143.
[18] Felderer, M., & Ramler, R. (2017). Special issue on collaboration in software testing between industry and academia. *Software Quality Journal*, *25*(4),1087-1089.
[19] Maher, Z. A., Shaikh, H., Khan, M. S., Arbaaeen, A., & Shah, A. (2018, November). Factors Affecting Secure Software Development Practices Among Developers-An Investigation. In *2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS)* (pp. 1-6). IEEE.
[20] Tao, C., Gao, J., & Wang, T. (2019). Testing and Quality Validation for AI Software–Perspectives, Issues, and Practices. *IEEE Access*, *7*, 120164-120175.
[21] Jiang, Z. M., & Hassan, A. E. (2015). A survey on load testing of large-scale software systems. *IEEE Transactions on Software Engineering*, *41*(11), 1091-1118.
[22] Sultana, S., Motla, Y. H., Asghar, S., Jamal, M., & Azad, R. (2014, February). A hybrid model by integrating agile practices for pakistani software industry. In *2014 International Conference on Electronics, Communications and Computers (CONIELECOMP)* (pp.256-262).IEEE.
[23] Mailewa, A., Herath, J., & Herath, S. (2015, April). A Survey of Effective and Efficient Software Testing. In *The Midwest Instruction and Computing Symposium. Retrieved from http://www.micsymposium. org/mics2015/ProceedingsMICS_2015/Mailewa_2D1_41.pdf*.
[24] Dias-Neto, A. C., Matalonga, S., Solari, M., Robiolo, G., & Travassos, G. H. (2017). Toward the characterization of software testing practices in South America: looking at Brazil and Uruguay. *Software Quality Journal*, *25*(4), 1145-1183.
[25] Deak, A., Stålhane, T., & Sindre, G. (2016). Challenges and strategies for motivating software testing personnel. *Information and software Technology*, *73*, 1-15.

[26] Royal j. j.,Kothapallii, s. An empirical approach to recommend best practices for successful software products.

[27] Maqbool, B., Rehman, F. U., Abbas, M., & Rehman, S. (2018, January). Implementation of Software Testing Practices in Pakistan's Software Industry. In *Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences* (pp. 147-152).

[28] Hynninen, T., Kasurinen, J., Knutas, A., & Taipale, O. (2018, July). Guidelines for software testing education objectives from industry practices with a constructive alignment approach. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (pp. 278-283).

[29] Garousi, V., Coşkunçay, A., Betin-Can, A., & Demirörs, O. *A Survey of Software Engineering Practices in Turkey (Technical report-Extended version)*. Technical report.

30 Jahan, M. S., Riaz, M. T., & Abbas, M. (2019, September). Software testing practices in IT industry of Pakistan. In *Proceedings of the 6th Conference on the Engineering of Computer Based Systems* (pp. 1-10).

[31] Bhuiyan, S. A. R., Rahim, M. S., Chowdhury, A. E., & Hasan, M. H. (2018). A survey of software quality assurance and testing practices and challenges in bangladesh. *International Journal of Computer Applications*, *975*, 8887.

[32] Engström, E., Petersen, K., bin Ali, N., & Bjarnason, E. (2017). SERP-test: a taxonomy for supporting industry–academia communication. *Software Quality Journal*, *25*(4), 1269-1305.

[33] Alégroth, E., Feldt, R., & Ryrholm, L. (2015). Visual gui testing in practice: challenges, problemsand limitations. *Empirical Software Engineering*, *20*(3), 694-744.

[34]Bin Ali, N., Engström, E., Taromirad, M., Mousavi, M. R., Minhas, N. M., Helgesson, D., ... & Varshosaz, M. (2019). On the search for industry-relevant regression testing research. *Empirical Software Engineering*, *24*(4), 2020-2055

[35] Anwer, F., Nazir, M., & Mustafa, K. (2017). Security testing. In *Trends in Software Testing* (pp. 35-66). Springer, Singapore.

[36] Florea, R., & Stray, V. (2018, May). Software tester, we want to hire you! an analysis of the demand for soft skills. In *International Conference on Agile Software Development* (pp. 54-67). Springer, Cham.

[37] Ilyas, M., Khan, S. U., & Rashid, N. (2020). Empirical Validation of Software Integration Practices in Global Software Development. *SN Computer Science*, *1*, 1-23.
[38] dos Santos, J., Martins, L. E. G., de Santiago Júnior, V. A., Povoa, L. V., & dos Santos, L. B. R. (2019). Software requirements testing approaches: a systematic literature review. *Requirements Engineering*, 1-21.

[39] Felderer, M., & Ramler, R. (2016). Risk orientation in software testing processes of small and medium enterprises: an exploratory and comparative study. *Software Quality Journal*, *24*(3), 519-548.

[40] Itkonen, J., & Mäntylä, M. V. (2014). Are test cases needed? Replicated comparison between exploratory and test-case-based software testing. *Empirical Software Engineering*, *19*(2), 303-342.

[41] Petersen, K. (2016). A Researcher's Experiences in Supporting Industrial Software Process Improvement. In *Managing Software Process Evolution* (pp. 235-255). Springer, Cham.

[42] Mohacsi, S., & Ramler, R. (2019, January). Why Software Testing Fails: Common Pitfalls Observed in a Critical Smart Metering Project. In *International Conference on Software Quality* (pp. 73-92). Springer, Cham.

[43] Bartolini, C., Bertolino, A., Elbaum, S., & Marchetti, E. (2011). Bringing white-box testing to service oriented architectures through a service oriented approach. *Journal of Systems and Software*, *84*(4), 655-668.
[44] Sawant, A. A., Bari, P. H., & Chawan, P. M. (2012). Software testing techniques and strategies. *International Journal of Engineering Research and Applications (IJERA)*, *2*(3), 980-986.

[45] Just, R., Jalali, D., Inozemtseva, L., Ernst, M. D., Holmes, R., & Fraser, G. (2014, November). Are mutants a valid substitute for real faults in software testing?. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering* (pp. 654-665).

[46] Singh, G. A STUDY ON SOFTWARE TESTING LIFE CYCLE IN SOFTWARE ENGINEERING.

[47] Hooda, I., & Chhillar, R. S. (2015). Software test process, testing types and techniques. *International Journal of Computer Applications*, *111*(13).

[48] Stoyanova, V., Petrova-Antonova, D., & Ilieva, S. (2013, March). Automation of test case generation and execution for testing web service orchestrations. In *2013 IEEE Seventh International Symposium on Service-Oriented System Engineering* (pp. 274-279). IEEE.