



PRIMARY RESEARCH

Software developemnt life cycle and testing

Abdul Basit^{1*}, Iftikhar Alam Khan², Anmol Zubair³, Khuram Ameen⁴^{1, 2, 3, 4} Department of Creative Computing, Bath Spa University Academic Center RAK, Ras Al-Khaimah, UAE

Keywords

Neural networks
Modern networks
Wireless network
Information processing

Received: 01 August 2019**Accepted:** 16 October 2019**Published:** 23 December 2019

Abstract

The development of big and large scale software is a complicated process. Bugs, errors and faults can occurred at any stage of development and these errors and bugs can be fixed as early as possible to sort out the problems. The process stops the propagation and reduces the costs of verification which might be happened. The engineers those who are able to maintain quality of the software can avoid from bugs in the system. Tasks can be done and verified earlier as while developing the code. The product deployment can be easy if the codes are done properly in the maintenance stage. Developing the codes and testing the software this can be effective for the codes testing schemes. In this research paper we are going to test software testing process which can lead to the deployment and testing of the software. The testing of software leads to the characteristics of the testing of software properly. The quality activities are involved with two steps. The first problem is discussed as revealing the faults and then the other one is accessing the faults. One process is ended, the quality cannot be measured. The quality can be measured at the earliest stage of the development of the software. But the quality is forced to maintain during the whole development process. Sometimes important classes with faults are difficult to find out. Software is excellent from which all errors and bugs are removed.

© 2019 The Author(s). Published by TAF Publishing.

I. INTRODUCTION

The development and testing of large software is done after the development of the software. Testing process is done according to the different software developers plan and schedule of software project management techniques and methodology. The software is developed in tasks and these tasks are mainly play very important key role in testing the software. The classification of the software is not sharp when the most of the activities of the software take part in different modules. In some other cases the development of software and testing of the software is not easy for the team. So the developer's team checks the scenario of the complete software design structure and the do the testing of the software properly [1]. All quality related activities are often done during testing phase. The quality of testing the software actually is not a phase but to make sure the quality of the software should be well enough to meet the requirements of the company. The quality and testing of the software make sure the product is developed, would work properly. The testing of software is always with different

stages but mostly with the feasibility study. This feasibility stage leads the software development and testing according to the standards [2]. These standards lead to the software testing and development according to the designed phase and software products can be tested. While testing the software, the design specification is very important for the inspection and analyzing the possible results of testing the software product. The testing of the software is too much costly if this product is testes after the development of the software and then these bugs reveal lead to the software test again. This process is very costly. The testing of software is executed many times by different phases. These different phases can be of different stages of the software development [3]. There are many stages which are involved in testing of software. the specific testing of the software involved include many costs. The selection of testing of suitable set of quality is little bit more complex and difficult to make the good approach towards testing the developed product. Testing is also involved along with strong validations in the software application. If validations are properly assigned to

*Corresponding author: Abdul Basit

†email: abdul@bathspa.ae

the application interface then errors and bugs can be easily handled in the software application. The validations in the testing of the software depends strong knowledge of the design structure along with background knowledge of programming and other languages skills. The quality activities are involved with two steps. The first problem is discussed as revealing the faults and then the other one is accessing the faults. One process is ended, the quality cannot be measured. The quality can be measured at the earliest stage of the development of the software. but the quality is forced to maintain during the whole development process. Sometimes important classes with faults are difficult to find out. Software is excellent from which all errors and bugs are removed.

II. RESEARCH METHODOLOGY AND RELATED WORK FOR SOFTWARE TESTING

A. Research Methodology

After explaining what the process of SDLC is now the process of Research Methodology is explained. To start with the term research is defined. Research is a process which may be defined as a careful investigation or inquiry especially through search for new facts in any area of knowledge. Research methodology is a way to systematically solve the research problem [4]. It may be understood as a science of studying how research is done scientifically. In it the various steps that are generally adopted by a researcher in studying his research problem along with the logic behind them is studied. The process of research methodology can be seen in Figure 2.

Mostly the tests are generated from stages of specifications and integration of the software. There some activities which have aimed to correspond the application structure. By these activities, we can check the application. The implementation with verification can be changed. There are some gaps help in finding the errors. So these errors are not easy to traceable. But the application structure in testing the modules is used properly. The activities which are done for the software testing may give validation verification for the software development [5]. These activities are done for the validation verification. There is difference between verification and validation. These activities can be different. These can be illustrated by the diagram. Validations and verifications are can be compliment of each other. Verification can be included in different stages. The verification actually is a process which can complete the validation process. The main validations can concentrate on the final products. The final products can be certain types of software development. Software testing may be called the code

testing.

B. Problem Statement

A research in the field of computer technology when comes up with an answer it makes use of Software Development Life Cycle and testing. Model for its implementation. This paper is going to show how this very statement works.

C. Research Objectives

1. The whole purpose of system development is the enhancement of the productivity of the organization and the group of people working in that organization, as system development got bigger there was a need to systemize the process of system development and come up with a set of steps that are required for any system.
2. SDLC analysis is conducted for the purpose of studying a SDLC and testing or its parts in order to identify its objectives. It is a problem solving technique that improves the SDLC along with testing and ensures that all the components of the system work efficiently to accomplish their purpose.
To determine specific needs of SDLC.
- 3 Discuss approaches and tasks of SDLC.
- 4 Evaluate tools and techniques.
- 5 Use appropriate methods and techniques to design software

D. Proposed Solution

Now since the problem statement is well understood the next step is to identify and understand the various terminologies. The work begins with explaining the overall process of System Development Life Cycle process followed by the overall process of Research Methodology and testing. The tests are related to the application environment and developing technologies in coding session. There are some problems in the databases of the software which cannot store the results properly and then the application design may be changes to remove the problem. The problems in software testing can be in the form of anomalies. These anomalies are actually happened I software and databases because of this data cannot be store into the databases. The number of faults can be traced properly and to remove early [6]. The slightly errors can make the whole application structure can be can be changed. To avoid from this situation we need to remove the errors earlier as possible as to fix the errors. The modules are tested and then verified in the application and structure of the application. The modules give the proper testing while software is run.

III. EXPERIMENT AND RESULTS

To start with the problem of the research is defined. The problem can be stated as in a given image as it is required to separate the desired image from the background of the image. Now stating the problem according to the research methodology it is required to understand and analyse the basic principle. When we are analyzing the basic principle we are following the theory analysis of Research Methodology and following the requirement analysis phase of SDLC. This SDLC model-2010 has been applied while developing both the software. viz., Spare parts cost Optimisation and Academic-HR Manager. The following are the results of applying the new software model to the following software. viz., a) Spare parts cost Optimisation and b) Academic-HR Manager. The estimated reliability of software developed was compared with and without the use of the proposed new SDLC model and the results are compared and have been

So during this phase we understand the concept of datamining, image processing and image segmentation. Further when the theoretical relevance is tried to understand, the step by step procedure of image segmentation is well understood which then allows to create an algorithm which would be needed to implement the image segmentation. This part comes in the design phase of Waterfall model of SDLC process.

The anomalies are removed from the databases structure to enhance the capacity of the software structure [7]. The increase in faults ratio may the application structure undeveloped and proper. Different types of testing are introduced. These testing's can take part to remove the application errors from the base structure [8]. The IBM world best company in manufacturing the computer units all over the world has introduced the removal of errors system from the applications. The detail procedure for testing the systems and their software all the companies have their own schedule and ways to remove the errors from the application. The allocation of different tasks to remove the errors from the application and design code structure may be different for the different companies. There are some responsibilities which

are provided to the software developer team to do test of removing of errors. There are some large organizations which can split the units into different parts to take the plan of the application structure and testing the code properly [9]. If these modules are developed then these modules are split into different tasks. These tasks can help in removing the errors step by step. If the modules are separated then the quality of the software testing increases and modules are easy to handle.

The application interface and environment can provide the information on faults and risks while testing the program. A lot of organizations collect general test suits which lead to the simple categories of testing the codes and the applications. There are lot of weaknesses of the developing technologies and programming languages which can lead to different faults and risks while testing the codes and structure of the application. The C++ and C# languages while memory management always comes with reputed risks which can lead to complicated and dangerous risk of memory leaks. These can be limited but cannot be voided from these types of situations. Coding coverage actually indicates the different regions of code in different segments. These regions of codes cannot be properly tested [10]. This may suggests different tests to repeat again and again. These test cases are generated if the specifications are available. When the code is ready, the code is put for execution. This reduces the testing time after coding. Early code generation is very effective point for the validation of the verification. The experience of testing the code to remove the errors becomes very easy in testing phase. The validations play very key role in catching the errors. If specifications are used for the coding purpose then validations give certain problems. The test cases are needed to be executed in the absence of complete application software of code validations and verification along with specifications. This leads to the incremental testing modules and segments of the code testing system. If modules are isolated then decision of incremental may divide the modules into different segments [11]. The execution of the whole system is main requirement of the testing phase.

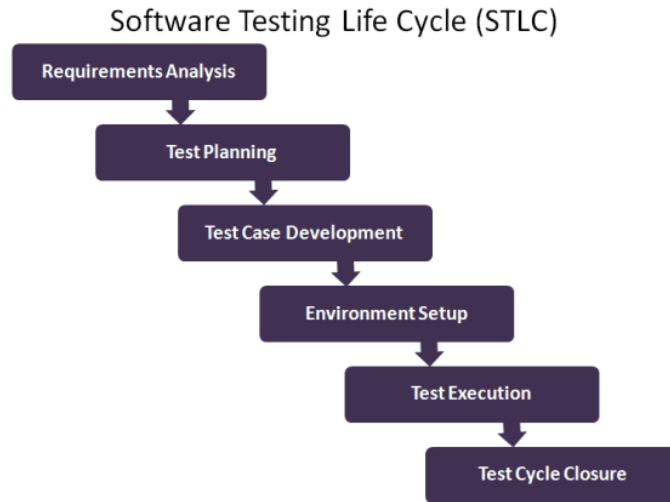


Fig. 1. STLC

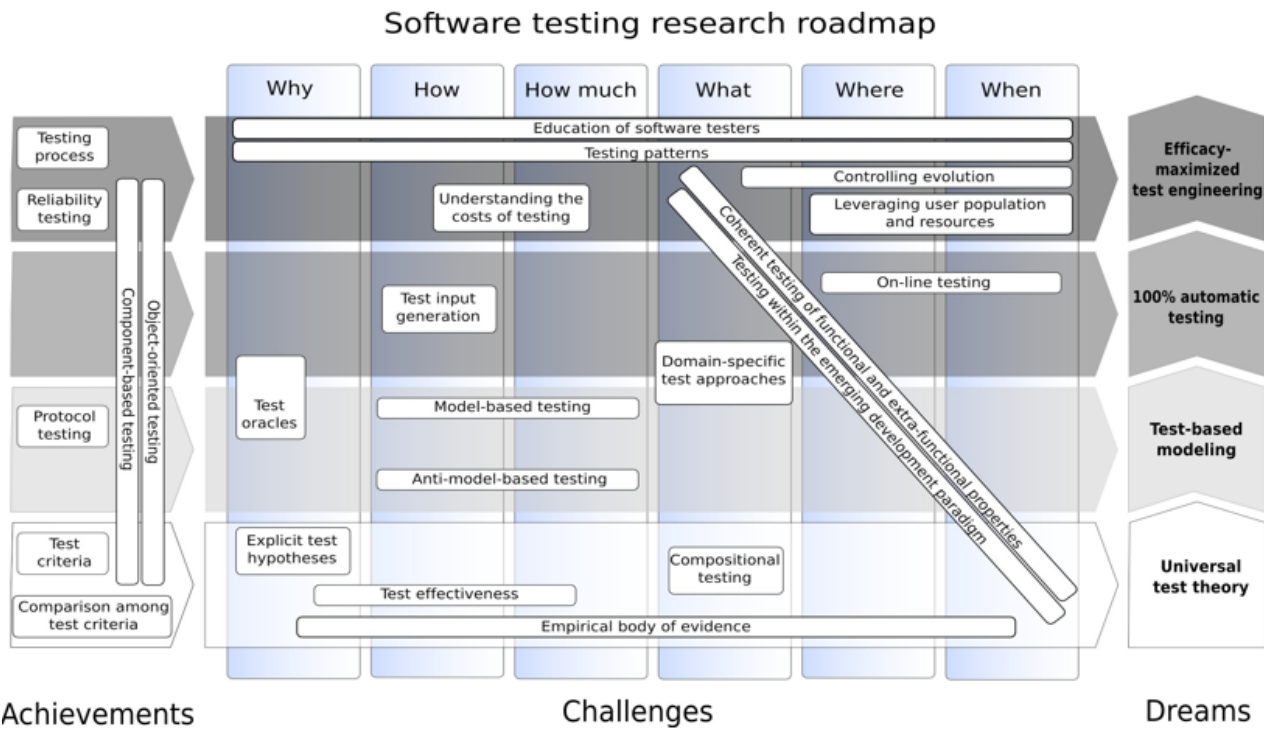


Fig. 2. Software testing road map

IV. SOFTWARE TESTING RESEARCH ROADMAP

A roadmap provides directions to reach a desired destination starting from the “you are here” red dot. The software testing research roadmap is organized as follows:

The “you are here” red dot consists of the most notable achievements from past research (but note that some of these efforts are still ongoing);

The desired destination is depicted in the form of a set of (four) dreams: we use this term to signify that these are asymptotic goals at the end of four identified routes for re-

search progress. They are unreachable by definition and their value exactly stays in acting as the poles of attraction for useful, farsighted research;

In the middle are the challenges faced by current and future testing research, at more or less mature stage, and with more or less chances for success. These challenges constitute the directions to be followed in the journey towards the dreams, and as such they are the central, most important part of the roadmap.

The roadmap is illustrated in Figure 1. In it, we have sit-

uated the emerging and ongoing research directions in the center, with more mature topics -the achievements- on their left, and the ultimate goals -the dreams- on their right. Four horizontal strips depict the identified research routes toward the dreams, namely:

1. Universal test theory;
2. Test-based modeling;
3. 100% automatic testing;
4. Efficacy-maximized test engineering.

The routes are bottom-up ordered according somehow to progressive utility: the theory is at the basis of the adopted models, which in turn are needed for automation, which is instrumental to cost-effective test engineering [11].

The challenges horizontally span over six vertical strips corresponding to the WHY, HOW, HOW MUCH, WHAT, WHERE, and WHEN questions characterizing software testing faces (in no specific order).

Software testing research challenges find their place in this plan, vertically depending on the long term dream, or dreams, towards which they mainly tend, and horizontally according to which question, or questions, of the introduced software testing characterization they mainly center on.

In the remainder of this paper, we will discuss the element (achievements, challenges, dreams) of this roadmap. We will often compare this roadmap with its 2000's predecessor by Harrold [12], which we will refer henceforth as FOSE2000.

V. QUALITY PLAN OF SOFTWARE TESTING

The quality of the software is designed by the quality plan of the engineers. The initial plan is different for the different code testing sessions. The initial plan can be tested by the quality of the designed cycle. This strategy of testing codes of the software is done in system testing sessions. According to the experience of the quality assurance, the quality of the company software is tested in design phase cycle [13]. The quality of testing the software code depends on the initial stages of the coding schemes and infrastructure of the software development methods. This strategy of the infrastructure describes the quality plan of the software testing schemes. The initial plan of the software testing scheme is incremental knowledge of the projects which are done carefully to redefine the structure of the schemes [14]. The detail module testing of the software codes can be different for the different coding sessions. The initial plan of software testing is taken into account and segments. After this, modules are tested. If the plans are not detailed, then software testing is stopped. These plans make different segments of testing. If the plan is not in detail then it is rescheduled [5].

A quality plan of software testing system includes all information of software testing. These segments and modules are tested easily. These programs are monitored properly. Testing is shown in single quality. These activities are controlled by software testing system. There some activities which have aimed to correspond the application structure. By these activities, we can check the application. The implementation with verification can be changed. There are some gaps help in finding the errors. So these errors are not easy to traceable. But the application structure in testing the modules is used properly [10]. The activities which are done for the software testing may give validation verification for the software development. These activities are done for the validation verification. There is difference between verification and validation. These activities can be different. These can be illustrated by the diagram. Validations and verifications are can be compliment of each other. Verification can be included in different stages. The verification actually is a process which can complete the validation process. The main validations can concentrate on the final products. The final products can be certain types of software development. Software testing may be called the code testing.

A. The Resources Allocated to Software Testing

1) *Tested items for software:* These items are modified and tested according to the schedule. These items are given schedule to maintain the quality of software testing. There some activities which have aimed to correspond the application structure. By these activities, we can check the application. The implementation with verification can be changed. There are some gaps help in finding the errors. So these errors are not easy to traceable. But the application structure in testing the modules is used properly [10]. The activities which are done for the software testing may give validation verification for the software development. These activities are done for the validation verification. There is difference between verification and validation. These activities can be different. These can be illustrated by the diagram. Validations and verifications are can be compliment of each other. Verification can be included in different stages. The verification actually is a process which can complete the validation process. The main validations can concentrate on the final products. The final products can be certain types of software development. Software testing may be called the code testing.

2) *Features to be tested in software testing:* In this module, features are tested. For testing these items, first of all features related to software is tested. Features which are not

tested in software.

Features are indicated first which are not to be tested. This feature helps the software I testing. The completeness of the plan for testing the software is done in early stage. There some activities which have aimed to correspond the application structure. By these activities, we can check the application [15]. The implementation with verification can be changed. There are some gaps help in finding the errors. So these errors are not easy to traceable. But the application structure in testing the modules is used properly. The activities which are done for the software testing may give validation verification for the software development. These activities are done for the validation verification. There is difference between verification and validation. These activities can be different. These can be illustrated by the diagram. Validations and verifications are can be compliment of each other. Verification can be included in different stages [13]. The verification actually is a process which can complete the validation process. The main validations can concentrate on the final products. The final products can be certain types of software development. Software testing may be called the code testing.

3) *Approach for testing the software:* The approach is chosen for testing purpose. All modules related to this are inspected and tested. Specific testing techniques are done. There some activities which have aimed to correspond the application structure. By these activities, we can check the application. The implementation with verification can be changed. There are some gaps help in finding the errors. So these errors are not easy to traceable. But the application structure in testing the modules is used properly. The activities which are done for the software testing may give validation verification for the software development. These activities are done for the validation verification. There is difference between verification and validation. These activities can be different. These can be illustrated by the diagram. Validations and verifications are can be compliment of each other [15]. Verification can be included in different stages. The verification actually is a process which can complete the validation process. The main validations can concentrate on the final products. The final products can be certain types of software development. Software testing may be called the code testing.

4) *Pass/Fail criteria for software testing:* Define the criteria for software testing is defined properly and then launched different schedules are covered. In this schedule, moderate impact faults are finding out and then remove the errors after testing. Suspension and resumption for testing software

Software testing purpose, there are conditions which can easily help in testing the software. Some activities are proper done. Some time the failure rate of software is increased [8]. Sometimes failure rate prevent the software for execution. Then testing is resumed if software is working properly. There are some checks which are used to test the software. There some activities which have aimed to correspond the application structure. By these activities, we can check the application. The implementation with verification can be changed. There are some gaps help in finding the errors. So these errors are not easy to traceable [5]. But the application structure in testing the modules is used properly. The activities which are done for the software testing may give validation verification for the software development. These activities are done for the validation verification. There is difference between verification and validation. These activities can be different. These can be illustrated by the diagram. Validations and verifications are can be compliment of each other. Verification can be included in different stages. The verification actually is a process which can complete the validation process. The main validations can concentrate on the final products. The final products can be certain types of software development. Software testing may be called the code testing.

5) *Monitoring the quality process of software testing:* The quality of software testing is monitored and reveals different deviations of the software programmers. Quality assurance of software testing is proper done under different conditions. These are the possible scenarios which control the software testing system. There are two forms which take part for supervision [16].

The quality process must be according to the prescribed standards. The quality process should describe the contingencies properly. There are two forms which can properly monitor the application. The supervision of the information is collected by the software testing. The quality engineer tries to make the software testing in running form. The activity of the software makes the testing properly [17]. The plans for the software can make the testing form change into testing the codes. There are some problems in the databases of the software which cannot store the results properly and then the application design may be changes to remove the problem. The problems in software testing can be in the form of anomalies. These anomalies are actually happened I software and databases because of this data cannot be store into the databases. The number of faults can be traced properly and to remove early. The slightly errors can make the whole application structure can be can be changed. To avoid from this situation we

need to remove the errors earlier as possible as to fix the errors. The modules are tested and then verified in the application and structure of the application. The modules give the proper testing while software is run. The anomalies are removed from the databases structure to enhance the capacity of the software structure. The increase in faults ratio may the application structure undeveloped and proper [5]. Different types of testing are introduced. These testing's can take part to remove the application errors from the base structure. The IBM world best company in manufacturing the computer units all over the world has introduced the removal of errors system from the applications. the detail procedure for testing the systems and their software all the companies have their own schedule and ways to remove the errors from the application. The allocation of different tasks to remove the errors from the application and design code structure may be different for the different companies [18]. There are some responsibilities which are provided to the software developer team to do test of removing of er-

rors. There are some large organizations which can split the units into different parts to take the plan of the application structure and testing the code properly. If these modules are developed then these modules are split into different tasks. These tasks can help in removing the errors step by step. If the modules are separated then the quality of the software testing increases and modules are easy to handle. The software development codes and session may be based on different abstraction levels on modules. There may be different modules which can do the testing of the software [9]. There are different symbols which can be use to illustrate the program testing. Specifications lead to the excellent development of software testing. This can be possible by introducing the code design sessions in the programming session. There are some modules which can be integration level. The integration level can also be used for the software testing. There are some specific levels of the software development which can pass the structure of the code session [5].

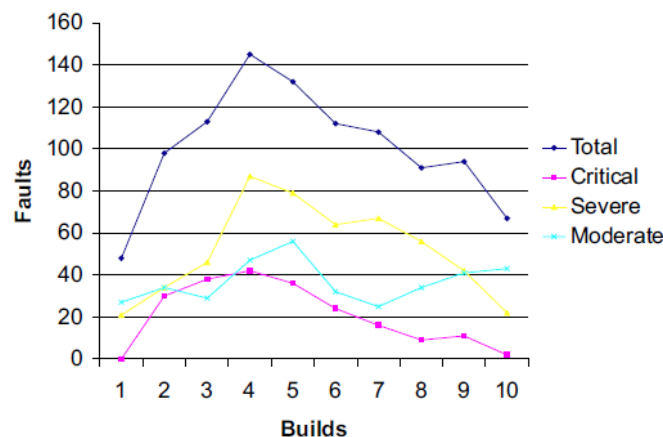


Fig. 3. Graph showing error rate

The faults which are found in the software testing system may lead to the integration properly. The categories can be different for different codes testing schemes. According to the above diagram which clearly mention the faults and their testing to prove the software system and quality. The diagram shows the quality of testing the software up to which extent the software testing system is working [19]. There are some problems in the databases of the software which cannot store the results properly and then the application design may be changes to remove the problem. The problems in software testing can be in the form of anomalies. These anomalies are actually happened I software and databases because of this data cannot be store into the databases. The number of faults can be traced prop-

erly and to remove early. The slightly errors can make the whole application structure can be can be changed. To avoid from this situation we need to remove the errors earlier as possible as to fix the errors. The modules are tested and then verified in the application and structure of the application. The modules give the proper testing while software is run [12]. The anomalies are removed from the databases structure to enhance the capacity of the software structure. The increase in faults ratio may the application structure undeveloped and proper. Different types of testing are introduced. These testing's can take part to remove the application errors from the base structure. The IBM world best company in manufacturing the computer units all over the world has introduced the removal of errors system from the

applications. the detail procedure for testing the systems and their software all the companies have their own schedule and ways to remove the errors from the application. The allocation of different tasks to remove the errors from the application and design code structure may be different for the different companies. There are some responsibilities which are provided to the software developer team to do test of removing of errors. There are some large organizations which can split the units into different parts to take the plan of the application structure and testing the code properly. If these modules are developed then these modules are split into different tasks. These tasks can help in removing the errors step by step. If the modules are separated then the quality of the software testing increases and modules are easy to handle [20]. There some activities which have aimed to correspond the application structure. By these activities, we can check the application. The implementation with verification can be changed. There are some gaps help in finding the errors. So these errors are not easy to traceable. But the application structure in testing the modules is used properly. The activities which are done for the software testing may give validation verification for the software development. These activities are done for the validation verification. There is difference between verification and validation. These activities can be different. These can be illustrated by the diagram. Validations and verifications are can be compliment of each other. Verification can be included in different stages. The verification actually is a process which can complete the validation process. The main validations can concentrate on the final products. The final products can be certain types of software development. Software testing may be called the code testing.

6) *Verification and validation in software testing system:* There some activities which have aimed to correspond the application structure. By these activities, we can check the application. The implementation with verification can be changed. There are some gaps help in finding the errors. So these errors are not easy to traceable. But the application structure in testing the modules is used properly [21]. The activities which are done for the software testing may give validation verification for the software development. These activities are done for the validation verification. There is difference between verification and validation. These activities can be different. These can be illustrated by the diagram [22]. Validations and verifications are can be compliment of each other. Verification can be included in different stages. The verification actually is a process which can complete the validation process. The main validations can concentrate on the final products. The final products can be cer-

tain types of software development. Software testing may be called the code testing. Codes testing may be included in the development stage of the code design session. This can include development stages. The validation activity leads to the software design. The validations of the software can be repaired if any problem happened in the code session. There are some properties which are natural for the software code development. Code session has great impact on web session and website development codes [15]. The software development codes and session may be based on different abstraction levels on modules. There may be different modules which can do the testing of the software. There are different symbols which can be use to illustrate the program testing. Specifications lead to the excellent development of software testing. This can be possible by introducing the code design sessions in the programming session. There are some modules which can be integration level. The integration level can also be used for the software testing. There are some specific levels of the software development which can pass the structure of the code session.

VI. FUNCTIONAL AND MODEL BASED TESTING IN SOFTWARE TESTING

This session can be applied to the all sessions of programming and development stages. These development stages can be achieved and places in separate sessions. There can be certain changes from module to the system level. System level can be different for other programming testing modules. These modules are based on software testing and coding methods. This can be applied on description of programming behavior. A function specification can be such type which can support to the coding session. There are some problems in the databases of the software which cannot store the results properly and then the application design may be changes to remove the problem. The problems in software testing can be in the form of anomalies. These anomalies are actually happened I software and databases because of this data cannot be store into the databases. The number of faults can be traced properly and to remove early [22].

The slightly errors can make the whole application structure can be can be changed. To avoid from this situation we need to remove the errors earlier as possible as to fix the errors. The modules are tested and then verified in the application and structure of the application. The modules give the proper testing while software is run. The anomalies are removed from the databases structure to enhance the capacity of the software structure [13]. The increase in faults ratio may the application structure undeveloped

and proper. Different types of testing are introduced. These testing's can take part to remove the application errors from the base structure. The IBM world best company in manufacturing the computer units all over the world has introduced the removal of errors system from the applications. The detail procedure for testing the systems and their software all the companies have their own schedule and ways to remove the errors from the application. The allocation of different tasks to remove the errors from the application and design code structure may be different for the different companies. There are some responsibilities which are provided to the software developer team to do test of removing of errors. There are some large organizations which can split the units into different parts to take the plan of the application structure and testing the code properly. If these modules are developed then these modules are split into different tasks. These tasks can help in removing the errors step by step [23]. If the modules are separated then the quality of the software testing increases and modules are easy to handle.

A functional testing is useful for the software testing. A functional specification can be behavior of program testing. If it is taken for the code session and code session can lead to the good programming session. Specification can be expressed in natural language and in computer language to identify the problems of the coding session. There is also code session for the infinite machine which can give different numbers. The unified modeling languages are also included in the sessions to make the software codes and programming session [24].

When UML is done then specifications are done. Functional testing techniques can improve the stages of coding sessions. Functional testing is shifted to the program testing system. The test designers can concentrate on codes to make the program clear from error at earlier stage [15]. So the developers and designers can be successful in removing the errors. There are some models which can take part in automatic testing of codes. The codes and programs which are done for the software designing are actually used for programming structure. When specifications of testing the software development are done then software developing stages go on smooth. If the software design is done for the functionality of the software which can take part in codes testing system then this process can be successful. If a shopping cart is develop then that time a system design is done for the progress of the software structure. The software for web testing tool can be many stages and tested with manifesto. There is a diagram which can be helpful in designing the software structure. If any errors are found then it's very

easy for us to do software testing. The software development codes and session may be based on different abstraction levels on modules [25]. There may be different modules which can do the testing of the software. There are different symbols which can be use to illustrate the program testing. Specifications lead to the excellent development of software testing. This can be possible by introducing the code design sessions in the programming session. There are some modules which can be integration level. The integration level can also be used for the software testing. There are some specific levels of the software development which can pass the structure of the code session.

A. Testing Levels in Software Testing

The software development codes and session may be based on different abstraction levels on modules. There may be different modules which can do the testing of the software. There are different symbols which can be use to illustrate the program testing. Specifications lead to the excellent development of software testing. This can be possible by introducing the code design sessions in the programming session [26]. There are some modules which can be integration level. The integration level can also be used for the software testing. There are some specific levels of the software development which can pass the structure of the code session.

B. Module Testing in Software Testing

Modules are the components which can verify the isolation step of the developers into different tasks and sub tasks. There are also some single modules which can make different of development structure. It is very important to remove faults and errors by module testing.

C. Integration and Component Based Software Testing

The quality of software testing for the single module may be enough but the quality of the software final stage may require different stage [7]. There is chance of failure of modules which can lead to the low quality. There is a difficult process to find out he faults but these faults can be traced in latter session of module component software testing.

D. Structural Testing in Software

There is also testing of structural testing which can do the testing of software level at initial stage. Measuring of codes is also very important in coverage of millions of lines of codes and identified by the code session. On the other hand, in structural design and testing is very important for code session modules [14]. On the other hand identify the features of software testing. In structural testing, there can be

easily extent the structural class for the testing of codes. The software development codes and session may be based on different abstraction levels on modules

VII. CONCLUSION AND FUTURE WORK

We have gone through various articles based on SDLC and found that there exists number of models to develop software. Each software has some strengths and weaknesses. This paper provides a necessarily abbreviated discussion of the history, state, and issues with SDLC and software analysis and design methodologies to determine future needs to improve quality and usefulness throughout the organization. Future research was identified and discussed in the following areas: A need to define the relative importance of key drivers of successful applications, specific techniques and processes for developing usable interfaces, best practices in servitizing applications development, SERVQUAL modifications to include IT services evaluation and to tease out the nuances between system and service in web sites, application use and satisfaction relationship elaboration, common methodological checklists of items for application development consideration, methods to move new techniques into industry practice, checklists for managerial roles in applications development, usability and user. Models are adopted on the basis of the requirements. Software testing has become very active area and session for the testing of programs codes. Now a day's quality engineers can lead to the different tools structure. There is great benefit from other software testing techniques. Engineers define many results for this software testing techniques. They can use many useful tools to check the errors from the codes at design time and run time. However, there can be a lot of challenges for the developers to develop the codes in the programs and check them nicely [27]. The quality activities are involved with two steps. The first problem is discussed as revealing the faults and then the other one is accessing the faults. One process is ended, the quality cannot be measured. Nowadays we are watching that technology is evolving at a very high speed. Keeping this in mind the developers have to choose the particular model to make the desired software. Correct selection of the development model can lead to the fast delivery and quality product

The quality can be measured at the earliest stage of the development of the software. But the quality is forced to maintain during the whole development process. Sometimes important classes with faults are difficult to find out. Specifications lead to the excellent development of software testing. This can be possible by introducing the code design sessions in the programming session. It is obvious that

those goals in such roadmap which have been settled as the dreams are destined to remain so. However, in a research roadmap the real thing is not the label on the finish, but the pathways along the traced routes. So, what actually is important that researchers focus on to sign progress are those called the challenges, and certainly the roadmap provides plenty of them, some at a more mature stage, other just beginning to appear. What is assured is that software testing researchers do not risk to remain without their job [13]. Software testing is and will continue to be a fundamental activity of software engineering: notwithstanding the revolutionary advances in the way it is built and employed (or perhaps exactly because of), the software will always need to be eventually tried and monitored. And as extensively discussed in this paper, for sure we will need to make the process of testing more effective, predictable and effortless (which coincides with the ultimate of the four testing dreams). There are some modules which can be integration level. The integration level can also be used for the software testing. There are some specific levels of the software development which can pass the structure of the code session [28]. If these applications are failed to fulfill the requirement of the company then these are again retested to remove the errors but this type of exercise is not good in the software development field. Users can tolerate only few things while testing, otherwise they reject the testing of the application. Reviews and inspections are very good in testing the software. The platform over which the software development and testing reside continues to evolve and remains exceedingly eminent. However, something so crucial and critical like Testing comes often quite late in the process of Software Development. There should be a maximum interaction between specification writers and Testers for better understanding and early review, which may fix ambiguity problems and consequently result in saving the cost of later fixing of the software. Testers after being clear about the specifications and requirements should hand over developers a certain lightweight test model, so they make sure the primary specification are met before handling the project for official testing [5]. Use of simulation tools can immensely help the testers in creating the similar environment in which the product is destined to run, certain exception testing and methods for the exception handling can be best determined. While testing the product in the similar testing environment for which the product is meant for, and that can be easily done by integrating the simulation within the Testing process. Hence, the future work in relevance with the testing process will be much more technology dependent harnessing the simulation and automated testing

model based approach, not only expediting the testing life cycle but also providing optimum bug prevention and efficient quality assurance.

A. Scope for Further Study

The dependent processes like incident management, configuration management and release management could be explored within the scope of development and maintenance. Further one to one mapping between the phases of SDLC and testing with these processes can be established.

REFERENCES

- [1] G. Booch, *The unified modeling language user guide*. New Dehli, India: Pearson Education India, 2005.
- [2] D. Avison and G. Fitzgerald, "Information systems development: Current themes and future directions," *Information and Software Technology*, vol. 30, no. 8, pp. 458-466, 1988. doi: [https://doi.org/10.1016/0950-5849\(88\)90142-5](https://doi.org/10.1016/0950-5849(88)90142-5)
- [3] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, and R. Jeffries. (2001) Manifesto for agile software development. [Online]. Available: <https://bit.ly/3G1uoSB>
- [4] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, *Agile software development methods: Review and analysis*. Espoo, Finland: VTT Electronics, 2017.
- [5] X. Chen and P. Sorenson, "Toward TQM in IT services," in *Proceedings of the ASE Workshop on Automating Service Quality*, Atlanta, Georgia, 2007.
- [6] D. Alonso-Ríos, A. Vázquez-García, E. Mosqueira-Rey, and V. Moret-Bonillo, "Usability: A critical analysis and a taxonomy," *International Journal of Human-computer Interaction*, vol. 26, no. 1, pp. 53-74, 2009. doi: <https://doi.org/10.1080/10447310903025552>
- [7] J. Baschab and J. Piott, *The Executive's Guide to Information Technology*. New York, NY: Wiley, 2017.
- [8] S. Brinkkemper, "Method engineering: Engineering of information systems development methods and tools," *Information and Software Technology*, vol. 38, no. 4, pp. 275-280, 1996. doi: [https://doi.org/10.1016/0950-5849\(95\)01059-9](https://doi.org/10.1016/0950-5849(95)01059-9)
- [9] S. Conger, "Information technology service management and opportunities for information systems curricula," in *Strategic Information Systems: Concepts, Methodologies, Tools, and Applications*. Hershey, PA: IGI Global, 2010, pp. 2606-2616.
- [10] S. Conger and B. Landry, "Problem analysis: When established techniques don't work," in *Proceedings of the 2nd Annual Conf-IRM Conference, Dubai*, 2009.
- [11] E. Brynjolfsson and L. M. Hitt, "Computing productivity: Firm-level evidence," *Review of Economics and Statistics*, vol. 85, no. 4, pp. 793-808, 2003. doi: <https://doi.org/10.1162/003465303772815736>
- [12] T. Guimaraes, C. P. Armstrong, and B. M. Jones, "A new approach to measuring information systems quality," *Quality Management Journal*, vol. 16, no. 1, pp. 42-51, 2009. doi: <https://doi.org/10.1080/10686967.2009.11918217>
- [13] S. Conger, "IT Infrastructure Library ITIL v3," in *The Handbook of Technology Management*, H. Bigdoli, Ed. New York, NY: John Wiley & Sons, 2010.
- [14] B. Brewin. (2010) Glitch prompts VA to shut e-health data exchange with defense. [Online]. Available: <https://bit.ly/3fYxagU>
- [15] M. Gallagher, A. Link, and J. Petrusa, *Measuring Service Sector Research and Development*. Washington, DC: National Institute for Science and Technology, 2005.
- [16] B. Boehm, "A view of 20th and 21st century software engineering," in *Proceedings of the 28th International conference on Software Engineering*, Shanghai, China, 2006. doi: <https://doi.org/10.1145/1134285.1134288>
- [17] A. Cater-Steel and M. Toleman, "Education for IT service management standards," *International Journal of IT Standards and Standardization Research (IJITSR)*, vol. 5, no. 2, pp. 27-41, 2007. doi: <https://doi.org/10.4018/jitsr.2007070103>
- [18] C. U. Ciborra, "Notes on improvisation and time in organizations," *Accounting, Management and Information Technologies*, vol. 9, no. 2, pp. 77-94, 1999. doi: [https://doi.org/10.1016/S0959-8022\(99\)00002-8](https://doi.org/10.1016/S0959-8022(99)00002-8)
- [19] F. Brooks, Jr., *The Mythical Man-Month*. Reading, MA: Addison-Wesley, 1975.
- [20] C. Goldenstern. (2010) Closing the 21st century service capability gap. [Online]. Available: <https://bit.ly/3u4I90o>
- [21] S. D. Galup, R. Dattero, J. J. Quan, and S. Conger, "An overview of IT service management," *Communications of the ACM*, vol. 52, no. 5, pp. 124-127, 2009. doi: <https://doi.org/10.1145/1506409.1506439>

- [22] C. U. Ciborra, "The platform organization: Recombining strategies, structures, and surprises," in *Bricolage, Care and Information*. London, UK: Springer, 2009.
- [23] D. E. Avison and G. Fitzgerald, "Where now for development methodologies?" *Communications of the ACM*, vol. 46, no. 1, pp. 78-82, 2003. doi: <https://doi.org/10.1145/602421.602423>
- [24] D. Avison and S. Gregor, "An exploration of the real or imagined consequences of information systems research for practice," in *Proceedings of the 17th European Conference on Information Systems*, Verona, Italy, 2009.
- [25] S. Alter, "Viewing systems as services: A fresh approach in the IS field," *Communications of the Association for Information Systems*, vol. 26, no. 1, p. 11, 2010. doi: <https://doi.org/10.17705/1CAIS.02611>
- [26] A. Bertolino, "Software testing research: Achievements, challenges, dreams," in *Future of Software Engineering (FOSE'07)*, Minneapolis, MN, 2007. doi: <https://doi.org/10.1109/FOSE.2007.25>
- [27] S. Conger, *Process Mapping and Management*. New York, NY: Business Expert Press, 2010.
- [28] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75-105, 2004.