



A Brief Survey on Introducing Software Testing in the Beginning of the Course

Vinutha.H¹, Rajeshwari LN²
Assistant Professor¹, Student²
Department of ISE
RRCE, Bangalore, India

Abstract:

This paper suggest that learning software testing in the beginning stages of academics makes a student to feel comfortable in writing program and feels working environment familiar by knowing the flow of the program. Following proper way of learning new technology and programming skills makes programmer, user a good programmer. Combining the subjects like programming and testing together as early as possible in the same semester is a tough job for students, lectures and instructors. For computer students, testing is the important skills which are very much essential for every student. By testing the program while writing makes student provides backup of the program and suggestions to improve program process. In order to provide suggestions about the program, the software engineering uses technique called test-driven development which provides quick feedback of the program. Test driven development is the low level language which is in the human readable form. The main of this paper is to introduce software testing courses in beginning of the any computer based courses to make the student a good programmer.

Keywords: Education, unit test, test-driven development, testing skills, low-level features.

I. INTRODUCTION

Regardless of learning software testing is very important in the field of developers, testers, etc. Universities are not providing sufficient education on the testing skills to the students. Students lack in the testing skills because of the same reason. This reason makes student to think about the program again and again which takes long time. Testing is the most required knowledge for every student of computer based courses. Testing encounters how the flow of the program is executing and how the small piece of the program behaves and generates the satisfied outcomes to the users. It also validates and verifies the program and identifies error to provide suggestions about the program to modify the program. Testing is very important in the field of industry for checking the quality. Why to learn testing in the beginning of the courses? Learning testing at beginning of the courses reduces the risk of wasting of cost, time to work again and again on the same topic without knowing the fault in the program and generates error-free program. Considering the life cycle of the Software development, testing can be started at the stage of the requirements gathering process and keeps moving till the development of the software. This paper describes how software testing is important in the career of the software developers, testers, software designers etc. Based on different conducted as per the requirement. Teaching software testing is just not teaching about the flow of the execution to the students, it also a big challenge for teachers, instructors and for the universities. Introducing both the programming and the testing subject together in the same semester needs planned mapping, because universities has their own way of teaching and making the students educated. But once the planning of introducing the software testing in the beginning of the course is successfully done. Then for the teachers and instructor do not face any problem in teaching the testing skills, they make students to evaluate or validate their own program by providing the set of inputs. This makes the student to feel

difficult in the early stages of learning the skills to write the program. In order to overcome this issue, universities or institutions should support the students to learn the testing skills along with programming skills. Firstly, including the software testing in the beginning should be properly determined. Student should provide supporting tools for conducting the testing of their program in the laboratory. However at the same time introducing the software testing course at the beginning is not just teaching the skills, it requires lot of efforts in teaching and adding new subject in between the course or in the packed scheme. From the observation the testing begins after the fruition of coding stage, sometimes it starts even before the primary line of code can be composed. In the life cycle of the regular programming item, testing starts at the phase when the determinations are composed, i.e. from testing the item details or item spec. Once the determinations are surely knew, the required to outline and execute the experiments. Choosing the proper method that lessens the quantity of tests that cover an element is a standout amongst the most vital things that you have to mull over while outlining these experiments. Bugs begin when the experiments are executed. As an analyzer you may need to perform testing under various conditions, i.e. the application could be in the underlying stages or experiencing fast changes, you have not as much as enough time to test, the item may be produced utilizing an existence cycle display that does not bolster quite a bit of formal testing or retesting. Further, testing utilizing diverse working frameworks, programs and the designs are to be dealt with. Revealing a bug might be the most essential and at times the most troublesome assignment that you as a product analyzer will perform. By utilizing different devices and unmistakably imparting to the designer, you can guarantee that the bugs you find are settled. Utilizing computerized apparatuses to execute tests, run contents and following bugs enhances proficiency and viability of your tests. Likewise, keeping pace with the most recent advancements in the field will increase your profession as a product test build. Building

up a decent programming framework is an extremely troublesome assignment. To make a decent programming item, various measures for programming quality credits should be taken into clarification. Framework inconvenience measurement assumes an imperative part in controlling and supervision of programming quality since it typically influences the product quality properties like programming unwavering quality, programming testability and programming practicality. In this way, programming quality confirmation should be tended to keeping in see the new systems, instrument, philosophies and procedures relevant to programming advancement life cycle.

II.LITERATURE SURVEY

An author [1] in this paper describes that checking the quality of the program is dependent of the previously used tools like method used to test the case, procedures, policy, standards. This makes use of the technique called Software quality Assurance, which are framed by combining the methods like controlling of the code, feedback of the code, designing of the program and software. It mainly concentrates on how the testing is used to solve the problem in the program and obtains the required output. By testing the program, quality of the program can be increased and guarantees satisfying output for the users. Here testers need to agree the terms with the customers to produce better system. It also brief about the already available way of teaching skills disadvantages. The disadvantages can be overcome by integrating the new technologies at same platform. Software quality checking mechanism activates the users through different ways of testing mechanisms. At last it describes the importance if the testing to avoid the problems like redundancy of code, long execution speed, lack of knowledge of the students about the testing. In paper [2] author tells about how the software is important in the cycle of developing the any new quality. By conducting the software testing when developing the program to implement the system which finds bugs at every level of development. This method is used and implemented in the incremental development model. It also rechecks the program that has some summary about program flow in order to make the changes or modification in the program for better usage of the code. This technology is used to avoid the execution of the same code which does not have any effect on the minor changes. Software testing is the part of the program meanwhile software updates also plays important role in development of the new system. This paper concentrates majorly o incremental symbolic execution method for parallel development of the system by just changing the code which is already in existing. It also makes use of feedback of the previous program just to avoid the repetitions in the newly development code. This paper [3] tells how the Test Driven Development is useful in the process of the testing by knowing the benefits and parallel development process procedure. This method provides a timely feedback or suggestions about the program in order to increases the efficiency of the code for better development of the system. TDD is the mechanism of developing software using the same test cases again and again. Whenever the new test cases are developed, the developers need to verify the newly introduced test cases on the system. It is based on methodology called agile method that checks the program in detail in order to generate the timely suggestions about the program. The main objective of the TDD process is to use this technology when developing the components in parallel applications mechanism. The development of the component using parallel application uses four major steps namely

partitioning, mapping, agglomerating, and communicating. The testing is done based on how the test cases are written, and they just require a proper coding to solve or to develop new system. In this paper [4] the importance of modern laboratory is how much needed in the field of programming and testing. Using updated techniques in laboratory improves the efficiency of the program, allows unit testing, avoids wasting of the time, and improves rethinking capacities of the programs, designers and testers. This paper guides every university to adopt this technical laboratory with well equipped tools which are very helpful for the lab instructors. Applying knowledge in laboratory makes student to feel comfortable about the working editor. To allocate each system to every students instructor should make a blueprint to connect the system to exchange the date among the students without any network connections. Using developed laboratory boost up the execution and testing speed of the programs and overcomes the traditional way of executions. This kind of laboratory allows student to feel the real requirement of the programs. This paper [5] describes the ability of the individual to learn the testing based on their testing knowledge. Testing is very much needed in the feel of quality assurance, development sector etc. Testing maintains half of the development work. Testing is the knowledge which can be performed by the students with enough programming skills. Student can become a good if he knows enough knowledge about the requirements of the written programs. But the fact is student do not show the interest in learning the testing skills and make it as their profession. Because computer based courses do not add testing subject as core subject, it is added at the end of the course. This negligence makes students to face problem during their project development as well as in future working environment. Students in starting semesters scoreless mark in laboratory because they learn how to write the program, not the testing skills or tracing of program executions. During courses they learn only few testing skills, which are helpful in knowing or learning new testing skills in professional career. The testing skills learnt during course act as basement for all testing. The objective of this paper is that student should be able to hold firmly the concept of testing at least as early as possible and to avoid same set of test cases when testing their programs. If they use same set of test cases they are not producing the effective program for the industries.

III.CHALLENGES

An investigation of this software testing made to analyze the important requirements and their challenges or difficulties in integrating this subject. By collecting the feedback from the teaching staffs and non teaching staff about an idea of including the software testing following challenging topics raised:

A. Decide how and when programming testing to be presented in the educational modules

This task says that how to include testing in the predefined course in which semester and also brief about how to interconnect both software testing and programming subject without making any individual into some confusion, trouble, system management issues, teaching hours issues etc. Planning of introducing the software testing subject at the introductory level should be carefully done. If the plan is done carefully then chances of increasing the risk is for sure for both students and for teaching staffs. Simply including this subject is not easy, because student can acquire the knowledge up to some extension. They cannot learn all at same time. It is good to include this subject when students are able to apply the

programming skills practically. If not, student feels testing is not useful. From teaching and non-teaching staff view point, workload is increased because they need to prepare useful materials, assignments, laboratory allotment.

B. Misusage of resource by the students

Usage of resources should be guided by the instructors to students because of chances of wasting of resources are more. Resources are wasted because of the lack of the testing skills. Initially testing requirements should be assembled before starting the testing process. Student should know the value of testing of the program. By knowing the values of the testing student do not feel difficult or burden in learning the testing skills. Learning proper usage of the resources makes student to use the resources in future very meaningfully without making any loss to the organizations.

C. Testing activities should be conducted in a proper way

Testing activities are conducted using the technique called Test Driven Development. This technique provides the time to time suggestions about the program of the students in order to improve the efficiency of the program by rethinking about the program modifications. Test Driven Development is the most used approach in the field of testing. Because adding the software testing in the beginning of the course causes some difficulties in order to the following activities are conducted:

Design: how the test cases are to be applied.

Test cases: Used to test the program.

Abstract: used to hide the important strategy of the test cases.

Implementation: running the written code by applying the test cases. mrectify the task we completely depend on the Test Driven method. In order to use test driven development technique, student should first gather the required test cases. Once done with the test cases, next is to execute the code to apply the test cases on the program. This process continues until the student obtains the required output. obtains the required output.

IV. ADVANTAGES

The advantages of teaching the software testing at the beginning of the courses are as follows

A. By testing the programs provides timely and useful suggestions about the particular program.

B. Reusability of the code whenever necessary.

C. Conducting testing makes student to know about the program deeply.

D. Requirement of the program should be gathered before the execution starts.

V. CONCLUSION

This paper gives an idea about how to introduce the software testing subject at the courses, what are the difficulties faced by teaching staff, non teaching staff and by students when introducing this subject in the packed way of teaching. Student acquires good knowledge about the programming by the benefits of this subject and improves the quality of the program by replying back about the program improvement using test driven development method. As we discussed earlier, instructor should think about how the lab management can be handled without any confusion. And teaching faculty should explain the concept of testing skills that should reach the student and student must execute the program by applying the theoretical knowledge taught in the classroom. An idea is just to know the value of learning the software testing skills in the beginning stages of computer activities. Testing becomes non teaching subject, learning testing is useful; student improves their programming skill by solving the same program in different ways. Student's testing skills should improve or

increase at the early stages, introducing testing at the beginning will not affect the learning skills of program. Student exhibit their own idea of solving programs, which takes them to feel complex program very easy. This all results in what are the best testing skills or method is good to solve the program without wasting time or over thinking of the problem without wasting time or over thinking of the problem.

VI. REFERENCES

[1]. Nayyar Iqbal, M. Rizwan Jameel Qureshi. "Improvement of Key Problems of Software Testing in Quality Assurance," Department of Computer Science, COMSATS Institute of Information Technology, 2009 nayyar_iqbal@hotmail.com, rjamil@ciitlahore.edu.pk

[2]. Shengjian Guo, Markus Kusano Chao Wang "Conc-iSE: Incremental Symbolic Execution of Concurrent Software," Department of ECE, Department of CS, Virginia Tech Blacksburg, VA, USA, 2016

[3]. John W. Burris, "Test-Driven Development for Parallel Applications," Department of Computer Science and Industrial Technology and Southeastern Louisiana University Hammond, LA, 2017 jburris@southeastern.edu

[4]. Adam Smolarczyk, Désiré Rasolomampionona, "A Modern Laboratory Test Bench for Power Protection Unit Testing," Warsaw University of Technology, Faculty of Electrical Engineering, Institute of Electrical Power Engineering Warsaw, Poland, 2016 Adam.Smolarczyk@ien.pw.edu.pl, Desire.Rasolomampionona@ien.pw.edu.pl

[5]. Odd Petter N. Slyngstad, Jingyue Li, Reidar Conradi, Harald Rønneberg, Einar Landre, Harald Wesenberg, "The Impact of Test Driven Development on the Evolution of a Reusable Framework of Components – An Industrial Case Study," Department of Computer and Information Science (IDI), Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2008

[6]. M. E. Caspersen and M. Kolling. Stream, "A first programming process." ACM Transactions on Computing Education, Mar. 2009.

[7]. S. H. Edwards and Z. Shams. Comparing test quality measures for assessing student-written tests. In Companion Proceedings of the 36th International Conference on Software Engineering (ICSE Companion 2014), pages 354–363, New York, NY, USA, 2014.

[8]. S. H. Edwards and Z. Shams. Do student programmers all tend to write the same software tests? In Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education New York, NY, USA, 2014

[9]. S. H. Edwards, Z. Shams, and C. Estep, "Adaptively identifying non-terminating code when testing student programs," In Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE '14), New York, NY, USA, 2014

[10]. M. Pozenel, L. Furst, and V. Mahnicc, "Introduction of the automated assessment of homework assignments in a university-level programming course". In 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), May 2015.