Integrated Development and Testing Environments (IDTE) – A Research Proposal

1Ranjith Engu, 2 Sharat Chandra Kotakonda

1Blekinge Tekniska Högskola, Karlskrona, Sweden, 2Blekinge Tekniska Högskola, JNTU-H, Hyderabad, Andhra Pradesh, India

E-mail: ranjithengu@lycos.com, sharath.chandd@gmail.com

ABSTRACT

This Paper discusses about the idea of integrating the development and testing Environments which might be useful for the companies to increase communication and collaboration between two different teams working on different environments (i.e testing and development environment), Produce a quality product by decreasing the testing costs and performing repeated testing in different environments using different tools. We have designed some research questions with which we can come to a conclusion whether the IDTE tool will be effective or not. In our research process we are also trying to find out which tools integration makes the IDTE tool effective, in the next article we will be discussing about the benefits and limitations of IDTE tool by analyzing the data obtained from the Literature review.

Keywords: Software Testing, IDTE, Integrated Testing and Development Environments, Software Development, Development Environments, Performance, quality, reliability, developing and testing Environments, Integrated, development, testing, IDTE.

1. BACKGROUND

Testing is the most important part of software development life cycle as it consumes more than 40% of total cost[1]. The quality of the software depends on the amount of testing done. It is known fact that testing and development are two completely different worlds completely separated from each other. This separation is not just departmental, it can also be seen in the tools they use to develop or test the software. The developers use different development environments to develop application and the tester’s use not one but many testing tools for different testing techniques [2]. “Studies suggest that a significant portion of development and maintenance costs go to this retesting, which is known as regression testing”[1]. Harrold M.J has clearly explained and stated that retesting consumes more than 70% of the maintenance cost. This retesting should be done if the software is modified or a new version of the software is released. In order to avoid the regression testing on the software that is not modified, it is very important for the developers and tester to communicate with each other which also helps in increasing the quality of the product [4]. Many bugs are introduced due to lack of communication [3]. According to soner, “Discovering and fixing software bugs is a difficult maintenance task, and a considerable amount of effort is devoted by software developers on this issue. In the world of software one cannot get rid of the bugs, fixes, patches etc. each of them have a severity and priority associated to it. There is not yet any formal relation between these components as both of these either depends on the developer and tester or on customer and project manager to be decided on”[5]. Alternatively to fix these bugs, developers and tester need to meet up frequently and these meeting take lot of time. This communication gap is forcing the developers and testers to compromise software quality for time to market [5]. This separation also leads to reworks and delays time to market.

The testers often mistake a simple bug for defect (i.e. A simple change in setting or reconfiguring can fix that issue) and raise a ticket. This leads to misunderstanding and wastes developer time. The tester’s lack of knowledge in developing environment causes some serious misunderstanding [6]. To clear these misunderstandings developers have to answer a lot of questions, which might be difficult. Some defects are hard to reproduce i.e. say tester finds defects running some test and the raises a defect ticket. But when developer tries to fix it, he can’t reproduce that defect in order to isolate the problem and fix it. In some cases the tester finds a defect and raises a defect ticket and when developer checks it works fine on his side. This is because they are running the tests on different environments. At this situation misunderstandings between developers and testers may arise and may not cooperate with each other, which leads to delay in the project. So there is a need for one common tool which can aid developers and testers and improve the communication between them which in turn improves the quality of the product and time-to market [2]. One of the aims of our research is to increase the communication and collaboration between developers and testers using a common tool. We termed such a tool as Integrated Development and Testing Environment (IDTE). In this research we intend to study the testing techniques that can be integrated and how we can bridge the developers and testers with IDTE, benefits and limitations of IDTE.
2. INTRODUCTION

The quality of the product lies in the fixing the number of bugs during testing. These bugs can be minimized when testing is performed number of times. Continuous testing and re-testing increases the cost of the development process. Following the traditional approaches which separates analysis, design, code and testing phases may cause inconsistency in the development process[11]. Fulfilling the requirements of every team member in the software development life cycle is quite difficult. This is the main focus of our research where we have focused only on two phases (Development and testing) which are very important to produce a quality product through effective communication and collaboration. The use of common tool such as IDTE helps the organizations in improving the communication and in producing the quality product.

In the section 2, we have discussed some aims and objectives of our research following with challenges /Problem focus. In section 3 we explained the process of addressing the research questions, in the later part we have listed our goals and expected outcomes followed by discussion and conclusion.

3. AIM/OBJECTIVES

Here are some of the main aims and objectives of our research study. The purpose of this study is

1) To increase the communication and collaboration between developers and testers using a common tool.

2) To Show the effective usage of the IDTE tool.

3) To analyze and compare different testing techniques that can be integrated with IDE.

4) To compare the benefits and limitations of IDTE and draw conclusions based on the results.

5) To produce a quality product, by giving the developer a chance to test the code and fix the bugs in the early stages of the software development life cycle.

4. CHALLENGE/ PROBLEM FOCUS

The research questions are stated below.

1. What are the major issues or problems that hinder the communication or collaboration among developers and testers?

2. What are the current industry practices or techniques to improve the communication and collaboration between developers and testers?

3) Are there any IDTE in the market? How integrated development and testing environments can help increasing the communication and collaboration between developers and testers?

4) How to measure the extent of communication and collaboration between developers and testers?

Many Researches were carried out by many researchers in order to improve the quality of the product and induce effective communication between the team members.

Though there are many communication tools which help the organization to decrease the communication gap between the team members, they are not enough for understanding each other’s work in the team. From the study of IBM, it is known that providing Common Environments to the developer and the tester helps in understanding the work along with the increase in the communication which in turn helps in producing the quality products by correcting the bugs during the early stages of the development lifecycle[8][9].

Through many decades it is known that the quality of the product lies in fixing the maximum number of bugs, which is possible only through testing. In order to produce a product with minimum number of defects organizations are using wide variety of testing tools in addition with better development tools there are many tools and technologies available in the market[10]. To carry out our research we have to analyze the best tool which suits and performs well when integrated and to prove that these tools are effective when used.

5. RESEARCH METHODOLOGY

The research methodologies can be classified into three types. They are i) Quantitative ii) Qualitative and iii) Mixed methodologies [10]. Our research strategy consists of a Quantitative part and Qualitative part. In our Qualitative work we conduct a literature review, survey to gather information such as The problems/ issues that hinder the communication or collaboration between developers and testers, the possible solutions to those issues or traditional techniques being used to overcome those problems, to identify any software tools which can aid both developers and testers (IDTE). We intend to conduct an experiment to see if those traditional techniques can solve the issues or problems that hinder the communication or collaboration between the developers and testers.

The Quantitative part consists of analyzing the data from the experiment and determining whether or not the traditional testing techniques have increased the communication or collaboration between developers and testers. We determine to what extent the communication between developers and testers has increased. For this we measure the extent of communication between developers and testers. We conduct the experiment again using a common tool (IDTE) and compare the data from the experiment with the previous one to draw conclusions.
5.1 ADDRESSING RESEARCH QUESTIONS

A detailed approach of our research is described below:

We use different approaches to address the research questions mentioned in section 4, and the goal of the research questions that are being addressed is described in the next section.

We intend to find the answer and address the first two research questions by conducting a systematic literature review and survey. Data from this survey will be analyzed and used in our research to answer the above questions.

We intend to review the literature and the internet to see if there is any such tool available in the market. Which can aid both developers and testers in their daily tasks and also help in increasing communication between them. We will gather information on the tool and its features to answer the third question and we will use the tool in our experiment.

To address the last question we intend to review the literature, use some statistical calculation and draw out some means to measure extent of communication and collaboration between developers and testers. With this measure we will compare traditional techniques and the common tools (IDTE) ability to improve communication and collaboration between developers and testers.

We will conduct an experiment and for the experiment we require ten volunteers (Experiment results are described in detail in the final paper). The ten volunteers are divided into 2 groups. Each group is assigned a different task. One group is assigned the work of developing a web site and other to test it. We will apply some of the traditional techniques to increase the communication between developers and testers. For the first part of the experiment 10 webs are developed and tested. Following data is collected No of Defect detected, no of defects fixed, Rework done to fix defects in person-hours or person-days, Defect Detection Ratio, No of Disagreements between Developer and Tester. This data is used to measure the extent of communication between developer and tester. We will conduct the experiment again using the tool (IDTE) and then use the measure to measure the extent of communication. We compare this with the previous one and see which will better increase the extent of communication between developers and testers. If the common tool (IDTE) increases the communication and collaboration better than traditional techniques then we prove that our research work is Successful.

6. GOAL/RESULT

6.1 EXPECTED OUTCOME:

Following are expected outcomes of the study

- The use of IDTE is beneficial to an organization only if the integration is limited to some testing techniques.
- Increase in the software productivity and quality.
- Fill the communication Gap between the developers and the testers in order to produce a Quality product.
- To find the problems or issues that hinders the collaboration and communication between developers and testers.

6.2 CONTRIBUTION

Our potential Contribution from our research is that many organizations will be benefited by introducing a common tool for development and testing environments. It also includes:

- Analysis of Different types of tools and techniques which suits for integration.
- Helps to measure the Extent of collaboration between the developers and testers.

7. DISCUSSION

At this time it is very difficult to predict whether the integration of testing tools with the development IDTEs will be a success case. This was an idea that evolved from real time experience and we all know that many projects undertaken by different companies do not meet the deadlines and this might be due to many reasons, but one of the many reasons is testing [5]. Many companies now a days are focusing on just delivering the projects with a belief that they can produce different versions of the product if it does not succeed or if they cannot meet the deadlines (for delivery), but the important thing here is that, though the product is released in different versions with some changes (ex:removing bugs) to the previous versions, “it must be tested”. Here the question arises, why does the company invest in testing different versions of the same product, why not invest in testing the original product avoiding testing and development cost for different versions, this is because the company personnel are afraid and might believe that testing consumes more than 40% of the total project cost [1]. In order to decrease the testing cost we came up with a research idea of integrating development and testing Environments and believe that this IDTE tool helps in decreasing testing costs and improve communication and collaboration between developers and testers.
Detailed differences about different development and testing environments will be discussed as the research progresses. At the moment it is hard to predict the count of different development and testing tools and their compatibility.

8. CONCLUSION

We strongly agree that no tool can replace the traditional technique such as face to face interaction (that is very effective means of communication), but in case where they are not being used effectively. The alternative is to facilitate the communication and collaboration through a common tool. It can be proved that IDTE fills the communication gap between developers and testers in an organization. From our research study we have found that it not only fills the gap but also improves the productivity of the company as IDTE serves as a common tool for all in an organization.

REFERENCES


n/ITS03001USEN.PDF

http://www.carpathia.com/assets/files/VirtualLabWP.pdf


AUTHORS’ BIOGRAPHY:

Ranjith Engu
Department of software Engineering
Blekinge Institute of Technology,
Karlskrona,Sweden.
Ph: +46734784580

Kotakonda. Sharat Chandra
Department of Software Engineering,
Blekinge Institute of Technology &
JNTU-H,
AndraPradesh, India.
Ph: +919573275896