ISSN 2348 – 8034
Impact Factor- 5.070

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

A PREDICTIVE ALGORITHM FOR QUALITY ANALYTICS IN APPLICATION LIFECYCLE MANAGEMENT (ALM)

Thejashree H[*1], Devi T[2], Pooja V[3] & Latha S[4]
[*1,3&4]UG Scholar, Computer Science and Engineering, Rajarajeswari College of Engineering
Bangalore, India
[2]Asst.Professor, Computer Science and Engineering, Rajarajeswari College of Engineering
Bangalore, India

ABSTRACT

In today's business environment making decision before implementing anything is very important since it predicts what's going to be happen in the future and helps to reduce the chances of loss. ALM is a predefined process that involves many phases. Different tools of ALM can be accessed by the company based on the customer needs to fulfill their requirements. These tools when implemented generates huge amount of structured data which contains meaningful information and decision-making capabilities. Today an integrated view and approach using open source tool is very limited. This makes difficult to predict the possible results and results in delay of making decisions. This project will develop an analytic solution on top of Service X which generates data for the SDLC processes. So the purpose of this project is to develop an algorithm based solution that identifies the high defect rates, missing of release timelines and also reduces the time and cost.

Keywords: *Application Lifecycle Management(ALM), SDLC, Predictive Model.*

I. INTRODUCTION

ALM is a product lifecycle management of computer programs. Basically ALM start as an idea in where requirements are collected and continue till the end of the application, then the feedback is taken for the future work. Phases of an ALM perform their respective functionalities. Three major parts of an ALM along with their respective functionalities are,

Governance:	Making decisions about the application.
Development:	Creating or updating application.
Operations:	application is executed and managed.

The first aspects of ALM are Governance runs across all of them always making decisions about an application. Development is most intensive in the time between idea and deployment. Operations begin before deployment and then runs continuously till applications end of life.

Application governance makes decision about the application. Governance starts with building business module, where an idea for implementing an application is connected with some business strategies.

The second aspect of ALM is development, it is also called as software development lifecycle SDLC) which includes identification of current defects, planning, designing of application, building, testing, deploying and updating the application. The phases are organized as separate steps based on the methods that are used to develop and those method will be integrated.

The operation stage of ALM includes development of the application and maintenance of the technology. While developing the operation is viewed as a separate stage that combines operations and development together into fully integrated and continuous process.

ALM phase operations changes from business to business based on requirements. Test case is conditions or variables to check the functionality of the application.

Predictive analytics contains a variety of techniques from predictive modeling, machine learning, and data mining that make analysis of current and historical data to make statement of what will happen in future. In predictive analytics meaningful information's are extracted this abstracted information's are used to predict pattern behavior and trends. Predictive analytics can also be defined as predicting more detailed level of granularity, generating predictive score for each individual organizational element. This differentiates it from predicting or forecasting. For example, Prediction creates user groups that can be used for targeting with notifications from the firebase console. This helps you engage who are likely to make in-app purchase, and much more.

II. RELATED WORK

Application Lifecycle Management deals with approaches, methodologies and tools for integrated management of all aspects of software development. Its goal is to making software development and delivery more efficient and predictable by providing a highly integrated platform for managing the various activities of the development lifecycle from inception through deployment [2].

The need of ALM systems in modern software development is unquestionable. They support both the management in governance, the development by organizing information, and operation by providing the necessary information. The traceability and consistency is up most important for safety-critical developments, both for developers and assessors [3].

A new software process meta model geared to the utilization of ALM tool data and based on well-established practices and other meta models. Our meta model uses its unified structure to represent and store real project data from various ALM tools and can be used for projects following various methodologies and processes [7].

The application hosting platform supports service composition based on SOA, application deployment and management. The application providers can manage the lifecycle of their applications through the management service portal [8].

III. ENVIRONMENT OF ALM

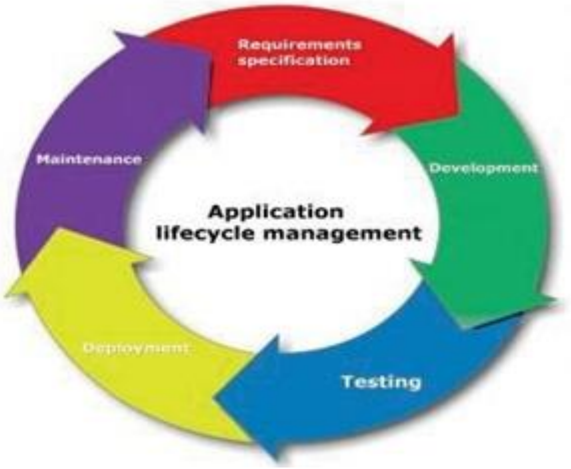

Fig: Environment of ALM

The requirements are collected from the end user, using those requirements an SRS document will be prepared that will be send to the development team the development team will perform designing and planning of the application. The designed part will be send to testing team where it undergoes several testing in this proposed system we are mainly focusing on functional testing and system testing. After testing the tested application will be deployed as a product.

IV. EXISTING SYSTEM

In case of ALM in the current environment there are number of tolls for the development of application. In the business environment the company may choose to use multiple tools that support one another or they may use single tool to fulfill the customer requirement. If we consider an application that needs to be tested and goes through the different testing phase, the testing of applications are done manually by the company employees. And if defect arises then the identification and fixing of those defects are also done manually using some tools which consumes more time and makes delay in making decisions. The existing system may produce high defects rate and cost which may also results in poor quality product.

V. PROPOSED SYSTEM

Today an integrated view and approach using open source tools is very limited because not all the application supports the open source tool, it creates confusion on what functionalities are present in which applications and it difficult the prediction of possible trends resulting in delays for making decisions. A system or solution which collates various project and process data which can be synthesized, analyzed and processed and displayed as a dashboard for various stake holders of the project would add lots of value for application and products development. This project will develop an algorithm based solution to integrate the various process workflows with the help of data generated from the software development process for predicting the quality trends, effort and delivery analysis. The results are planned to be made available as visual representations with continuous monitoring capabilities.

For a specific application the maximum test cases that are written will be 300 to 500. These test cases will be written by the project team person, if a person writes 80 test cases in one day then 500 test cases will be split by six persons to complete the complete application test cases, the test cases are manually written by the project team person. Then the test cases are executed by some testing tool. In our project we are using Test Link software tool for executing the test cases. Once the test cases are executed those may produce some defects for example out of 100 test cases it may produce 10 to 15 defects, so the identification of the test cases which produce defects is done manually by the project team person. It is difficult to identify the particular test case which produces the defects.

The identification of the defect may take more time and it is time consuming. For this purpose the company has to pay for the one who do this work. So our project will provide an algorithm solution for this purpose, which is a predictive algorithm. The algorithm predicts that in which test cases the defects are arises so the algorithm can solve the problem quickly. This reduces the time and cost for the company and provides a quality product to the company.

VI. METHODOLOGY

Software excellence assurance is facilitated by a test management tool which is a web based tool. This environment supports various techniques and reports such as test plans, test suits, test cases, test projects and user management.

Test plan consists assignment of user, results collected from the test (test result) etc. Test plans are the fundamentals for executing some set of collected tests on application. Test plans also includes some collection of test cases, description, name some important events (milestones) etc. Test plans are created by selecting the test plan management in test plan menu which will open another page. The necessary information's are filled by clicking on the create button.

Test cases are written in test suite. All the test cases have certain ID's. Test case is an essential unit of test link. The prefix of the test project is included in test case ID. The test case is created by selecting the test suit in test specification.

Test projects can be defined as a product or solution of the company. The products or solutions change the functionalities of the product but certain functionalities remain same. To create the test project the first step is to login to test link with user id and password. After successful login test link window will open. Test project is created by clicking the test project management.

Test specification structure is partitioned into two parts that are test suite and test cases. This partition is done by test link.

End to end test process in ALM:
The figure shows an end to end process in ALM. The requirements which are required for the product or project are collected from the end user. Then the collected requirements are documented as a software requirement specification (SRS) document. This SRS document will be stored in some ALM requirement tool. The ALM provides a variety of tools that includes Test Link, GitHub, Testopia and RFS. In this proposed system we are using TestLink tool for storing the requirements.

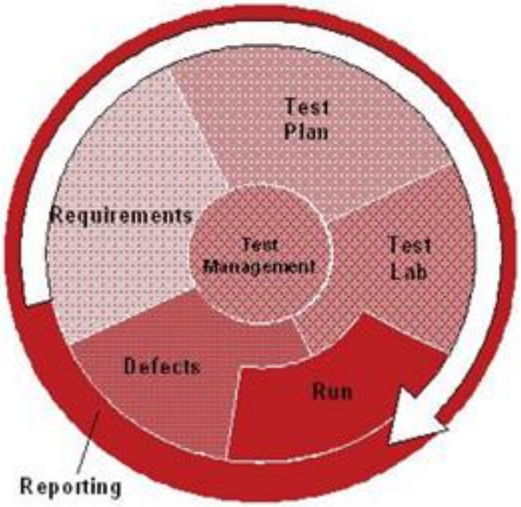

Fig: End to End test process in ALM

Once the requirements are updated to the ALM tool using those required test plans are made based upon the user requirements. Then the test cases are organized in the test plan. In this proposed system again we are using TestLink testing tool for the testing purpose.

Once the test cases are written those test cases will be linked to the requirements and perform the test execution in the TestLink. TestLink produces the results which may include some defects; the defects are logged and managed.

VII. CONCLUSION

In this paper we propose an algorithm based solution that identifies the high defect rates. Predictive modeling uses the test link tested results to predict the possible outcomes. It uses a machine learning concept that automates analytical model building and uses algorithms that continuously determines and learn from historical data and prevents complicated integration. ALM maintains quality as well as controls application development, provides preserved storage for code changes and accountability. It also allows for bugs and errors to be fixed with customer feedback. Since we are using ALM to predict the model which analyses the quality that leads to take faster decisions in software process and avoids major mistakes. This reduces the time and cost for the company and provides a quality product to the company.

REFERENCES

1. *Ki-Sook Chung, SangKi Kim, A study on the application lifecycle management over SOA based application hosting platform, 2010.*
2. *Sebastian Herden, André Zwanziger, Philip Robinson, Declarative Application Deployment and Change Management, 2010.*
3. *Hermann Lacheiner and Rudolf Ramler, Application Lifecycle Management ass Infrastructure for Software Process Improvement and Evolution, 2011.*
4. *Jeong Ah Kim, Seung Young Choi Sun Myung Hwang, Process & Evidence enable to automate ALM, 2011.*
5. *József Klespitz, Miklós Bíró, Levente Kovács, Aspects of improvement of software development lifecycle management, 2015.*
6. *Inigo Goiri, Ferran Juli, Jorge Ejarque, Marc de Palol, Rosa M. Badia, Jordi Guitart, and Jordi Torres , Introducing Virtual Execution Environments for Application Lifecycle Management and Resource Distribution, 20 15.*
7. *Ying Yang, Yong Zhang, Steven X. Ding, and Linlin Li, Design and Implementation of Lifecycle Management for Industrial Control Applications, 2015.*
8. *József Klespitz, Miklós Bíró Augmented, Lifecycle Space for traceability and consistency enhancement, 2016.*
9. *Joseph L. Hellerstein, Optimizing Software Packages for Application Management, 2016.*
10. *Petr Pícha, Pemek Brada, ALM Tool Data Usage in Software Process, 2016.*