ISSN: 0975-8585

Research Journal of Pharmaceutical, Biological and Chemical Sciences

Comparison of Effort Estimation Techniques Using Decision Table with Regression Analysis.

Sapphire Paul J*, Brumancia E, and Monisha GS.

Department of Information Technology, Sathyabama University, Jeppiaar Nagar, Chennai-6000119, Tamil Nadu, India.

ABSTRACT

Software estimation is dependent on prediction properties of system with special care to the development methodology. Effort estimation can be used as input to plans in project, plans in iteration, budget estimation, analyzing investments, terminology in pricing and bidding methods. Software bugs are the commonly occurring problems in an organization. Rework of those bugs involves huge loss in money as well as time. In order to find a solution to this problem, function point method and use case function points are used to estimate the effort or cost in the software development. The unique feature of the function point method is it is more popular and mostly used by all, when compared to the other estimation methods. Function point (FP) counts which is determined at the end is totally based on the user's view of the functions that is to be delivered with the finalized product. Function Point count is used to determine the size of a functional system. The use case is an approach which is utilized to limit the interaction between a user and a customer. A decision table is built which compares both the function point approach and the use case approach to compare which method will make the best feat to do a software business. We take error datasets which are trained datasets from the open source projects like Mozilla to perform the estimation procedure. In this paper effort estimation is done to find out which method would make the best feat in terms of time cost and quality to clear an incoming big. The decision table works on the technique of regression analysis. The approach which produces the best result would be utilized to clear the software bugs.

Keywords: Function Point, Use Case Approach, Regression Analysis, Effort Estimation

Corresponding author

ISSN: 0975-8585

INTRODUCTION

The application of scientific and technical knowledge for project planning, project execution and test methods may be defined as software engineering. It mainly deals with the maturation or the growth process. It also deals with the analysis and its related management. Software development has become a more tedious process due to its increasing size and complexity. Hence more care is required to take care of the growth process. The problems that is being shown in front of a software developer showcases the problems such as cost and degradation. Poor estimation is the major problem in the core of software development. Bad estimation will definitely result in a worst impact in the development process. Proper estimation is the most important task in the evolution process which includes proper planning in the project. Some organization might tend to underestimate a development in the project, which might lead to lack of resources. The worst case is overestimation, which lead to more resources and thus in turn leads to resource wastage.

Tracking the project development process is a challenging task to the organization. Another challenging task is to assign the best task to the best developer based on the estimates. Feedback from the development process is more essential to improve the development and improve the quality of the task. Tracking a system is important in any field, so as is the software development field. Estimation plays a central role in the maintenance of a development process. The mannequin or the method being employed should be well organized and incorporated with the most recent data. Any standard parameters if used so should be calibrated with those collected data.

SYSTEM ARCHITECTURE

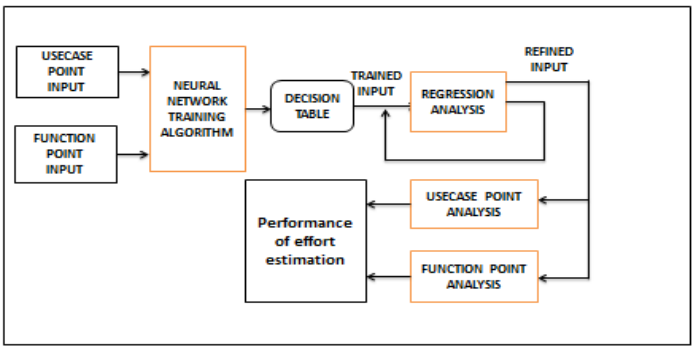

Figure.1. Architecture Diagram

PROPOSED METHOD

In this paper we have proposed a system which is used to estimate the effort that is required to decide which method [1] either the Function Point Approach or the Use case Approach is best suitable to clear an incoming bug. In this paper we have appropriated error datasets from open source project like Mozilla.

We first input trained error datasets to our system and nominate each bug to each developer. The developer logins the system with his credentials and tries to solve [2] the bug and saves the solution of the bug in the bug repository. This database is maintained confidentially by the database managers. This database is more useful to software developers in order to find out the solution of errors occurring during the development. Instead of sitting and reworking the bug again and again, the developer can find the solution of the occurred bug by mining the bug repository. Mining of the database can be done by data reduction or by keywords and attributes.

As stated earlier we have taken error datasets from open source projects like Mozilla [3], to find out the effort needed to solve an incoming bug. They are trained datasets. Incoming bugs are being assigned [4] to testers. The tester in turn tries to solve the bug and stores the result of the bug in a bug repository. The main aim of creating a bug repository is to maintain a database of the errors and their solution so that in future if any bug happens to occur the testers can mine the bug repository and find the solution instead of sitting and decoding. Decoding leads to waste of time and energy. It is not cost beneficiary to the organization.

ISSN: 0975-8585

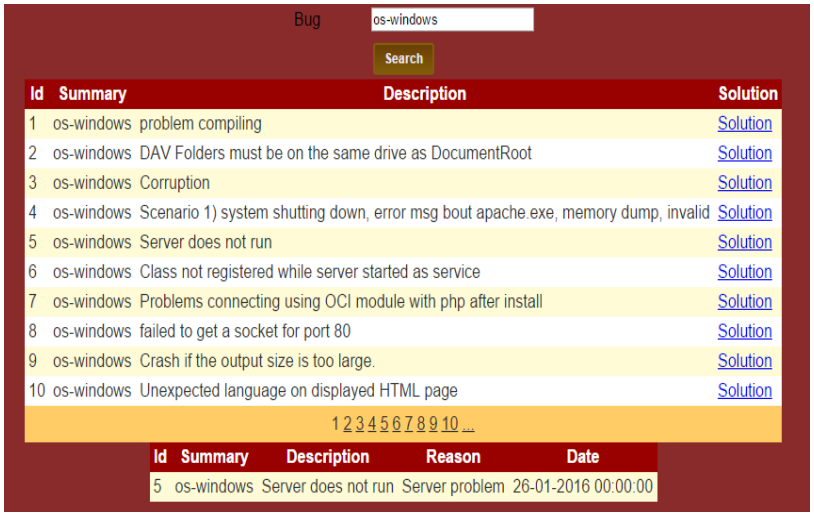

Figure.2. Bug Repository

After finding the appropriate solution for an assigned bug, the next phase is effort [5] estimation. As mentioned earlier, the user tries to find out which is the method to solve the bug, either the function point analysis or the use case analysis.

Function Point Analysis (FPA)

As the system keeps on growing in size, it is really tedious to estimate the size of the software development in the early phase. Function point analysis [6] is a well-known method for estimating the size of a development methodology in advance. Hence in our system we implement the function point analysis in order to estimate the size of the development process and compare with the use case analysis method.

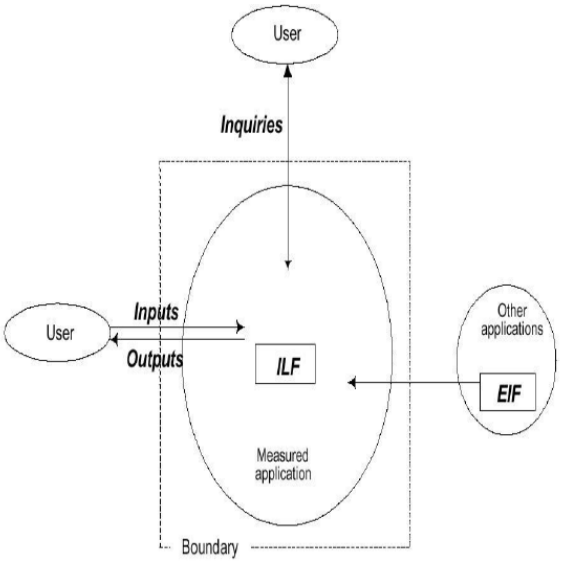

Figure.3. Components in Function Point Counting

Function Point Analysis Steps

1. Get the software requirements from the development team.
2. Identify the functions involved in it and the type of data and the level of complexity involved in it.
3. Add necessary weights for all the complexity levels to it which are predefined according to their difficulty level.
4. Unadjusted function points are computed based on the weights which has been added.

ISSN: 0975-8585

5. Grades are incorporated for the system characteristics which are predefined.
6. Value adjustment factor can be calculated by,

$$Vaf = 0.65 + (Td * 0.0.1)$$

Where,
Vaf = value adjustment factor
Td = Total data

7. Finally function point count is calculated by,

$$FPC = Upf * vaf$$

Where,
FPC = Function Point Count
Upf = Unadjusted Function Points

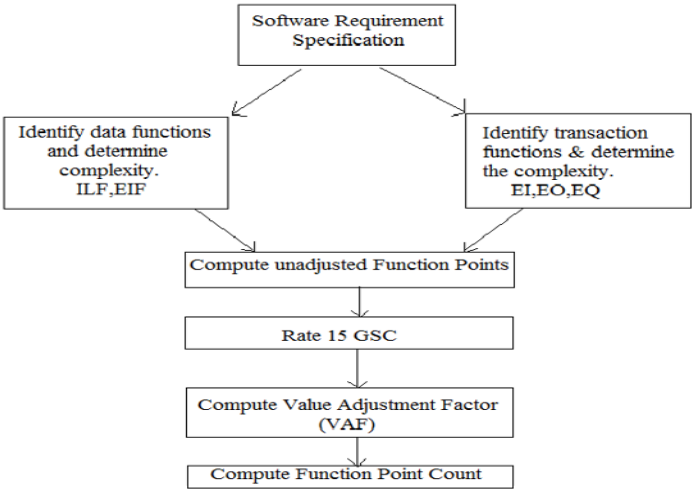

Figure.4. Steps for Counting Function Points

Use Case Analysis

Use case refers to the interaction between a client and a user. Use cases refer to functional requirements of software production. Use case analysis is used to estimate the capacity of a future software system at an early stage which might reduce rework and help in saving cost and time. We have described a simple approach for software effort estimation incorporating use case points. In object oriented analysis use case deals with actors and use cases. We classify the actors such as simple, average and complex. A weighting factor is being assigned to each type of actor which we have specified earlier. Use case analysis also deals with the technical factors that are involved to find out the performance of the system. Technical factors include methods such as its security features, installation methods and its ease of use, distributed system, system portability, end user efficiency, etc. We assign weight factors to each of the factors and compute the system. Environmental factors also play a major role in use case analysis. This also affects the size and complexity [7] of a project. Environmental factors refer to non-technical factors that affect the performance of a system. Environmental factors include methods such as UML familiarity, experience with the application, capability of the lead analyst and their motivation. Use case analysis method is mainly oriented with unadjusted use case points (UUCP) which in turn comprises of unadjusted use case weight and unadjusted use case actor.

The technical factor (TF) which is associated with use case analysis is calculated by using the formula,

$$TF = 0.65 + (TTOTAL * 0.01)$$

ISSN: 0975-8585

Where, TTOTAL refers to the total of all the weights.

The environmental factor (EF) which is associated with use case analysis is calculated by using the formula,

$$EF = 1.4 – (0.03 * ETOTAL)$$

Where, ETOTAL refers to the total of all the weights.

The adjusted use case points (UPC) which is associated use case analysis can be calculated by using the formula,

$$UPC = UUCP * TF * EF$$

Benefits of Use Case Analysis

Use case analysis help the system to capture requirements from the user point of view. Since they are written in natural languages, it is easy to understand and it provides a better communication between the client and the user. It helps to manage the complexity [8] of larger projects.

According to our trained data inputs, solution for an incoming bug has been found out and it is being stored in bug repositories [9]. The best method to solve this bug in terms of time, cost and performance is estimated using any one of the above mentioned analysis methods. Both the function point analysis and the use case analysis method is incorporated to find out the effort that is required to solve an incoming bug. Effort estimation in the sense refers to the numbers of human triagers required to solve a bug and the number of days required to solve that particular bug. According to our error data and the type of weights and factors which we used to implement our system we have come to a result that use case analysis method is the best method to solve an incoming in terms of count value and in terms of hours/persons that is required [10]. This comparison is done with the help of a Decision Table. Identification of the rules and actions is the first step in building a decision table since these factors have a great influence on the decision table. Specification of the entries plays a major role in the construction of the decision table [11]. Examines the DT with regard to the requirements of completeness, contradiction and correctness. To train the decision table, we have to add the input parameters.

RESULTS AND DISCUSSION

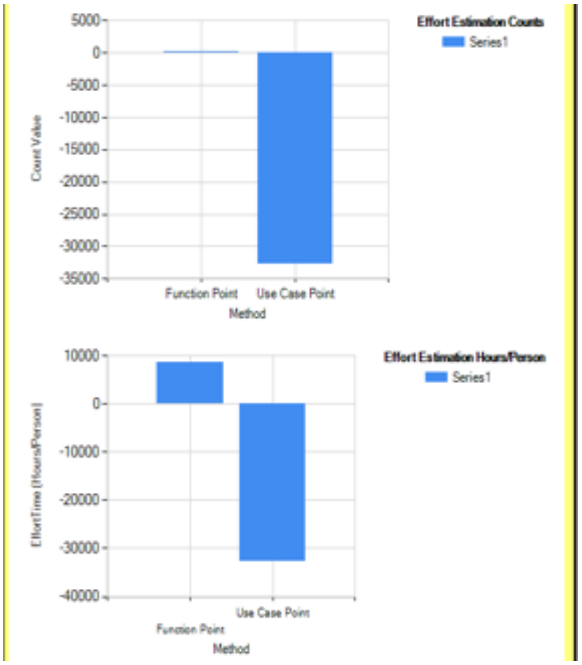

Figure.6. Comparison of both the approaches

ISSN: 0975-8585

According to our error data and the type of weights and factors which we used to implement our system we have come to a result that use case analysis method is the best method to solve an incoming in terms of count value and in terms of hours/persons that is required.

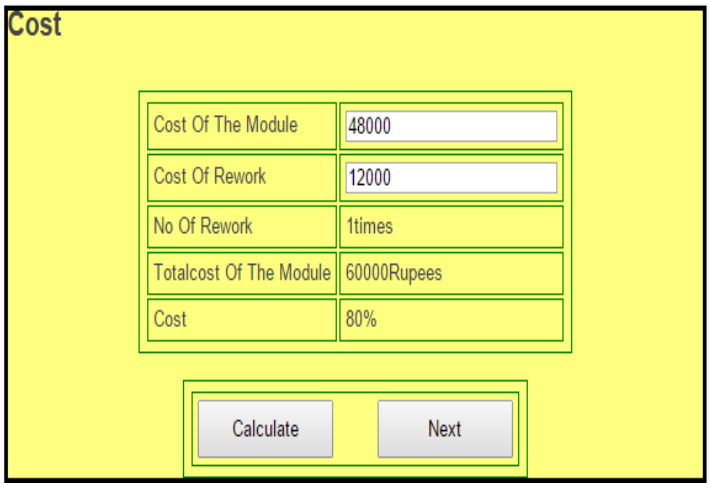

Figure.7. Cost Involved

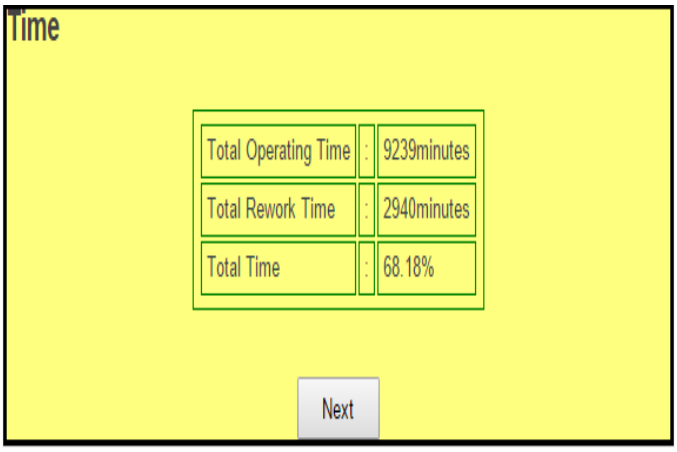

Figure.8. Time Involved

We have also implemented the system in such a way which would calculate the cost and time required to finish a particular module.

CONCLUSION AND FUTURE WORK

The entire system is divided into two phases. In the first phase a bug repository is developed and the incoming bugs are stored in it with the results for future purpose. In the second phase a tool for effort estimation is done and the result is compared with traditional effort calculated using functional point analysis. The calculated effort using the proposed method is found to obtain a more accurate result when compared with the traditional approaches. The estimated result can be used for further planning, tracking and calibrating the project. In future these functions, mining methodology and classification methods will be implemented in datasets of patients who have been affected with diabetes.

REFERENCES

[1] Abdukalykov R, Hussain I, Kassab M, Ormandjieva O Software Engineering Research, Management and Applications (SERA) 2011; 158-165.

[2] Swati M. Varade, Prof. M. D. Ingle The International Journal of Engineering and Science 2012; 1(2):239-242.

[3] Shay Artzi, Adam Kiezun, Julian Dolby, Frank Tip, Danny Dig, Amit Paradkar, Michael D. Ernst IEEE TRANSACTIONS ON SOFTWARE ENGINEERING 2010; 36(4):474- 494.
[4] Pamela Bhattacharya, Iulian Neamtiu IEEE Conference on Software Maintenance 2010; 85(2012):2275-2292.
[5] Tsunoda M, Kamei Y, Toda K, Nagappan M Mining Software Repositories (MSR), 2013 10th IEEE Working Conference on 2013; 4(13):429-438.
[6] Azath H, Wahidabanu R.S.D IET Software 2012; 6(4):335-341.
[7] Arpan Mukhopadhyay, Anwit Roy, Sourav Das, Swagatam Das. Techniques: Emerging Paradigms for Electronics and IT Industries, 2008, pp. 188-196.
[8] Katia Cristina A, Damaceno Borges 2012; 5:1-8.
[9] Katia Cristina A, Damaceno Borges, Iris Fabiana de Barcelos Tronto 2013;16(3):1-8.
[10] Oumout Chouseinolou, Ozlem Muge Aydm Turkish Journal of Fuzzy Systems 2013; 4(2):68-76.
[11] Patil L .V, Shivale N.M, Joshi S.D, Khanna.V Advance Computing Conference 2014; 8(14):1385-1391.