Vol.2, No.1 ISSN: 2349-0012

INNOVATIVE RESEARCH ORGANISATION
International Journal of Advance Research in Education, Technology & Management
(Scholarly Peer Review Publishing System)

Investigating Relationship between Software Effort Estimation and Team Parameters

Navjot Bagla[1] and Sumeet Kaur Sehra[2]

[1]GNDEC Ludhiana, Punjab, INDIA

[2]GNDEC Ludhiana, Punjab, INDIA

Abstract:

The relationship between effort estimation and team size is not yet fully analyzed. The relationship between effort estimation and team size must be understand so that we could be able to make reliable estimation and thus better project planning and management. There are many factors that affect software effort and team parameters like 'team experience', 'team skill set, interdependence, team incentives and team size. This study is made on the publicly available large datasets which involved detailed information on different projects. There is a need to formalize the relationship between effort estimation and team size for the better effort estimation purposes. Further, the algorithm is developed using the back propagation neural network with target MSE and optimum output depending upon the software under study. The system is trained using a no. of software modules and finally tested for different s/w modules other than the training modules. The results show a fair performance in computing the efforts estimate with respect to team parameters as discussed above.

Keywords: Efforts Estimation, Software Performance

1. Introduction

The competitive survival of many organizations depends on delivering projects on time and on budget. Software effort estimation has become a challenging and debatable research area. Software development effort is found to be one of the worst estimated attribute over the past few years. Software estimation means to predict the effort required to develop software. Software development efforts estimation is the process of predicting the most realistic use of effort required to develop or maintain software based on incomplete, uncertain and/or noisy input. a prediction that is equally likely to be above or below the actual result.

Team size is one of the base measures which are used for software effort estimation. The size of software project teams has been considered to be a driver of project productivity. People involvement in improvement activities is important because employees must adopt process innovation in their day to day activities. Communication and coordination is very important in different team size. The time spent in communication among team members increases with the size of the team. Project team size therefore affects schedule decisions. Team size is important when making decisions about efforts to be made in software development. It is very important factor in project success.

Among the important factors that affect productivity, project team size has been considered a key driver. The team size variable appears to impact software development effort in real-world empirical software engineering datasets, it can reduce software effort by significantly reducing the number of interactions between team members.

Data quality and its interactions with team size and FP were significant. Lower quality data on team size tended to increase software effort and lower quality data on FP tended to decrease the software effort. Thus, data quality appears to impact the results of study about effort estimation.

The team expertise information may allow us to better assess the impact of team size on software effort. it is important to consider how team size affects team performance . Increasing size offers a team multiple benefits. Labor can be subdivided across more team members, for example. This division of labor makes it possible to match workers with the tasks that are most interesting to them and for which they are best suited. It can also foster task specialization, which improves performance. Increasing team size is the increasing potential for conflict among members, which can again harm team performance strong teams have high productivity over all the assigned tasks.

The smallest team capable of getting the job done in the time available. The productivity per team member decreases as the team grows. In an Agile team, the project manager must define the relationships between the roles to enable the effective coordination and control of the project. Team efficiency is often dependent on the interaction between team members and the coordination of the team leader. Functional point of software, kilo lines of code, effort estimates, actual effort these are different parameters which plays very important role in effort estimation while developing a new software. Thus team size and its other important parameters affects both efficiency and productivity of teams directly about the efforts needed to develop the software .

2. Related Works

Bradley R. Staats et al. (2012) proposed in their paper that the total labor required completing team projects increase as the size of the project team increases. They carried out experiments to examine the causal impact of varied team size on estimation error, along with holding each team's task constant. The experiments also allowed to explicitly measuring estimators expectations concerning process gains and losses, which allowed documenting a possible mechanism for scaling the team. Mainly focus in this paper is on the gains rather than losses associated with increased team size.

D. Rodriguez et al. (2011) discussed in their paper to analyze the relationships between productivity, team size and other project variables using the International Software Benchmarking Standards Group (ISBSG) repository. They analyzed the ISBSG repository to study the relationship between staffing levels and productivity. They have applied statistical and machine learning approaches to a pre-processed subset of the ISBSG repository to facilitate the study. They have concluded that there are statistical

INNOVATIVE RESEARCH ORGANISATION
International Journal of Advance Research in Education, Technology & Management
(Scholarly Peer Review Publishing System)

correlations between team size, effort, productivity and project duration.

Sohaib S Bajwa (2009) has explored the relationship between effort and size and investigated what factors affect the nature of their relationship as well as increase the reliability of effort estimation. He has further investigated the size-effort relationship by applying Analysis of Co-variance (ANCOVA) to find significant factors. He has explored whether different representations of software size instead of using one total size figure would improve the estimation reliability. He performed a number of empirical studies utilizing the ISBSG dataset release 10 to investigate the effects of these factors on effort estimation. In this study has been conducted in two different phases' i.e. phase1 and phase2. In phase1 six analyses had been made to analyze the effects of Development Type, Architecture, Application Type, Language and Development Platform on Normalized Work Effort. While phase 2 shows that total COSMIC size along with other significant factors would better estimate the effort as compared to phase1.

Parag C. Pendharkar . James A. Rodger(2007) proposed in their paper the impact of team size on the software development effort using field data of over 200 software projects from various industries and analyzed the impact of team size and other variables such as software size in function points, ICASE tools, programming language on software development effort. They have investigated the factors impacting the software effort. Main focus of the paper is that ICASE tools have been known to have significant impact on software effort. They have proposed that the type of programming language did not have an impact on software effort.

Oddur Benediktsson et al. (2004) proposed in their paper the implications of working on projects with a fixed-effort level where increments may be utilized to obtain additional, or perhaps the only leverage, for project managers. The productivity gain from incremental development in fixed effort situations comes when relative productivity RLPR>1 as this is where the productivity is increased compared with a single delivery project. The key focus is that overall project effort is fixed and resources are determined for the duration of the project.

3. Methodology

The proposed work is aimed to study the affect of team parameters on software quality and performance. The team parameters may include team size, team mates skill set and experience, their interdependence, the platform they use for the software development and team members competency level to the platform they use for software development.

The algorithmic/model-based approaches for estimating the effort of software development use some sort of mathematical relationship between dependent and independent parameters to predict the estimates. The independent parameters normally characterise the software to be developed and the environmental conditions under which the development will be performed and the system will be operated whereas the dependent parameter corresponds to the effort estimated. The mathematical relationship is usually determined by the statistical analysis of data from past projects

The proposed technique includes following steps:

Objective of Software under analysis to be made clear to the team

Software Development Parameters compilation

Team Size

Team Skill Set

Platform used

Team Interdependence

Parameters Normalization

Parameters Metrics Development

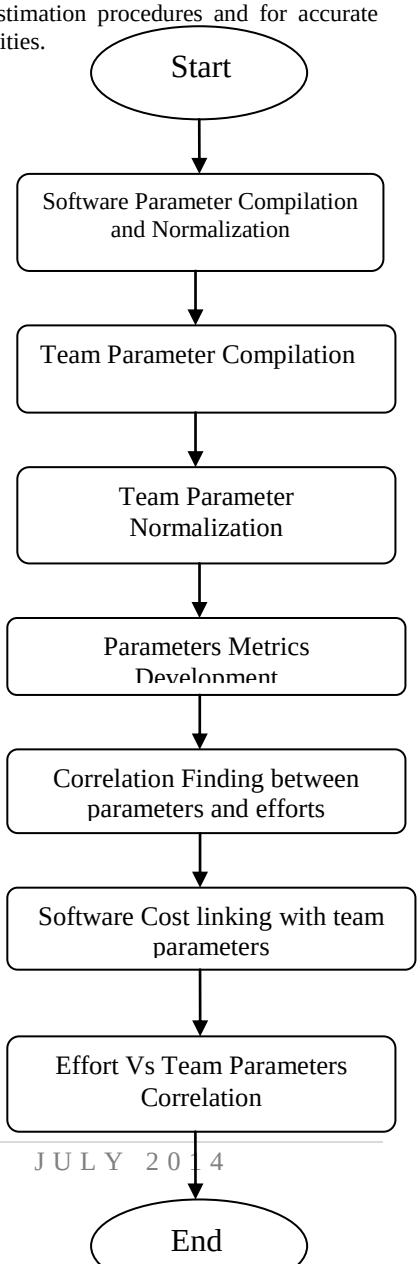
Correlation Finding between parameters and efforts

Software Cost linking with team parameters

Cost Vs Team Parameters (Efforts) Correlation

The success of a software development project is largely dependent on the effectiveness of the effort estimation process [1]. Correct effort estimation allows computing he required budget which, at the end, dictates the success of contract biding and constrains the project execution. This effort may be directly estimated based on the tasks to be performed or derived from the size of the product being developed (i.e., based on size estimation). The problem is that effort and size estimation is far from being a straightforward task, leading many projects to fail due to incorrect estimations. This way, companies frequently feel the need for improved estimation procedures and for accurate data to support those activities.

Flow Chart



INNOVATIVE RESEARCH ORGANISATION
International Journal of Advance Research in Education, Technology & Management
(Scholarly Peer Review Publishing System)

Finally, the efforts estimate can be formulated by integrating the above discussed team parameters as follows:
Efforts Estimate = F (TS, E, ID, IN,SK),
Where, TS → Team Size
E → Experience
ID → Interdependence
IN → Incentives
SK → Skills Proficiency
The efforts estimate is now made function of the team parameters based on results as follows:

$$\text{Effort Estimate} = K.\ \frac{E * IN * S}{TS * ID}$$

Where K is a constant and the value of constant depends upon the software under study. Presently as the K is a constant, the value may be taken as unity for evaluation purposes. In the presented work, the value of K is taken as unity.

4. Results

The proposed work is based on model basis. The validation of the work is required by using known case studies so that the parameters quantification could be justified. The software effort index largely depend upon the exact quantification of Team Size, Team Skill Set, Experince, Team Interdependence, Parameters Normalization, Parameters Metrics Development, Correlation Finding between Team parameters and Software effort made for the development of software .

Table 1. Variable Team size and efforts Estimate

Team size	3	5	7
Team Interdependence	0.5	0.5	0.5
Team experience	0.8	0.8	0.8
Team incentive	0.5	0.5	0.5
Team skills	0.6	0.6	0.6
Efforts Estimate	0.17	0.16	0.15

Table 2. Variable Team ID and efforts Estimate

Team size	3	3	3
Team interdependence	2	1	0.7
Team experience	0.8	0.8	0.8
Team incentive	0.5	0.5	0.5
Team skills	0.6	0.6	0.6
Efforts Estimate	0.05	0.07	0.06

Table 3. Variable Team Exp. and efforts Estimate

Team size	3	3	3
Team interdependence	0.5	0.5	0.5
Team experience	0.9	1	0.8
Team incentive	0.5	0.5	0.5
Team skills	0.6	0.6	0.6
Efforts Estimate	0.06	0.07	0.06

Table 4. Variable Team ICN. and efforts Estimate

Team size	3	3	3
Team interdependence	0.5	0.5	0.5
Team experience	0.8	0.8	0.8
Team incentive	0.5	0.9	1.8
Team skills	0.6	0.6	0.6
Efforts Estimate	0.04	0.06	0.08

Table 5. Variable Team Skill and efforts Estimate

Team size	3	3	3
Team interdependence	0.5	0.5	0.5
Team experience	0.8	0.8	0.8
Team incentive	0.5	0.5	0.5
Team skills	0.3	0.6	1.5
Efforts Estimate	0.04	0.06	0.08

5. Conclusion

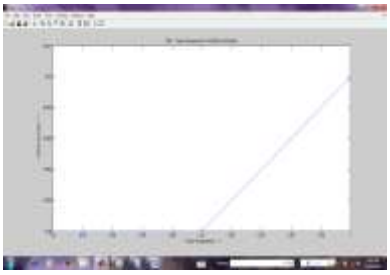
It is widely accepted that any software effort model becomes acceptable only if its practical usefulness has been proven via a thorough validation process. This is also true for estimating the effort of CBSD. The effort models surveyed have used different approaches and types of datasets to calibrate and validate their

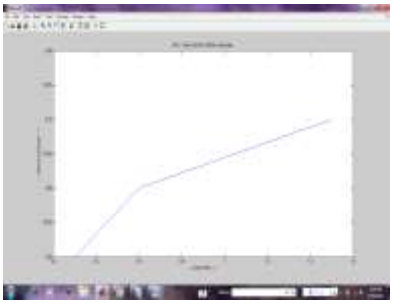
INNOVATIVE RESEARCH ORGANISATION
International Journal of Advance Research in Education, Technology & Management
(Scholarly Peer Review Publishing System)

modelling approach. most of the research attempts on the problem of predicting the effort of CBSD have mainly focused on dealing with issues related to the estimation of the effort of component integration. Although this is very much in keeping with the fact that component integration is the most determinant phase in the effort of developing software efforts estimation algorithm.

Future Scope

By using neural networks can be used in Fuzzy Analogy to integrate a supervised learning approach by comparing the estimated and the actual values. However, there are some shortcomings that prevent neural networks from being well accepted in cost estimation modeling. The most important is that they are considered as a _black boxes'. Consequently, it is not easy to understand and explain their process. To avoid this limitation, we have studied the interpretation of a cost estimation model based on a Backpropagation three-layer Perceptron network. This study has shown promising results by using fuzzy logic and the possibility theory in its estimation process, Fuzzy Analogy satisfies the two first criteria, i.e., the tolerance of imprecision when describing software project, and the uncertainty when estimating the development cost. The third criterion of an intelligent model that Fuzzy Analogy has not yet incorporated into its process is to learn from previous experiences. The learning criterion is required for any cost estimation model. Indeed, the software industry is continuously evolving: engineers use more and more high level programming languages, application generators, web-based technology, etc.





References

[1] Bradley R. Staats , Katherine L. Milkman , Craig R. Fox (2012), The team scaling fallacy: Underestimating the declining efficiency of larger teams , journal homepage: www.elsevier.com/ locate/obhdp 132–142.

[2] D. Rodrgueza, M.A. Siciliaa, E. Garcaa, R. Harrison (2011) , Empirical Findings on Team Size and Productivity in Software Development .

[3] Sohaib S Bajwa (2009) , Investigating the Nature of Relationship between Software Size and Development Effort.

[4] Parag C. Pendharkar Æ James A. Rodger (2007) , An empirical study of the impact of team size on software development effort, Information Technology and Management Volume 8, Issue 4 , pp 253-262.

[5] Oddur Benediktsson1 and Darren Dalcher2 (2004) , Project Effort Estimation: Or, when Size Makes a Difference , EuroSPI 2004 http://2004.eurospi.net/.

Author Profile

[1]Ms. Navjot Bagla is pursuing her M.Tech. in CSE from GNDEC, Ludhiana. Her field of interest is in software engineering and effort estimation.