International Journal of Engineering & Technology, 7 (4.12) (2018) 52-57

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

An efficient effort and cost estimation framework for Scrum Based Projects

Mohit Arora[1]*, Dr. Sahil Verma[2], Dr. Kavita[3]

[123]Lovely Professional University,Phagwara
**Corresponding author:sahil.21915@lpu.co.in*

Abstract

Software Process Models from its inception instill standardization and creates a generic culture of developing software for various IT industries. A great paradigm shift has been observed in terms of embracing Agile Development methodology as a viable development methodology in cross key business units. There is a buffet of agile methodologies comes under the umbrella of ASD, out of which Scrum got the highest popularity and acceptability index. Agile based software development is the need of immediate environment. There is an increasing demand for significant changes to software systems to meet ever-changing user requirements and specifications. As Agile is volatile, so effort estimation is challenging and still striving for perfection to decide size, effort, cost, duration and schedule of projects with minimum error. This cause sensitizes potential researchers all across the globe to start working on addressing the issue of inaccurate predication of efforts. The gap between estimated and actual effort is because of limited or no inclusion of various estimation factors like people and project related factors, inappropriate use of size metric and cost drivers, ignorance of testing effort, team member's inability to understand user story size and complexity etc. This paper attempts to bridge the gap of estimated and actual effort by the use of soft computing techniques thus taking the research to advance frontier area in terms of estimation.

Keywords: *Agile Software Development; Scrum; Effort estimation; Machine Learning.*

1. Introduction

Agile is a most common buzz word in the field of software engineering. With the advancements in the technology agile becomes the favorite choice of all the fortune 500 companies. The process can be viewed as Fig 1. Agile is neither a technique nor exactly a methodology. It is more of a concept that has an underlying foundation of four core values in the context of creating new era of software process models.

a) Agile focuses more on collaboration with its customers than negotiating contracts.
b) Agile believes in the principles on welcoming change over following a plan.
c) Agile focuses on working software over writing lot of documentation
d) Over the tools and process, agile focuses on individuals and their interactions.

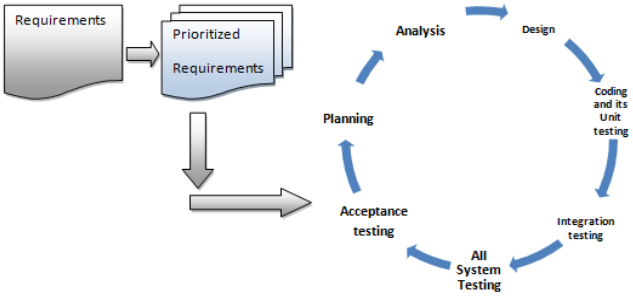

Fig 1: Essence of Agile

Scrum has been the most popular and widely acceptable agile methodology. Agile practitioners and researchers has carried out wide variety of research and found there are two main reasons behind software project failures are: Improper estimation in terms of project size, cost, and staff needed, and Uncertainty of software and system requirements. Mckinsey and the University of Oxford has conducted a study on 5,400 large scale IT projects, and found that on average large software projects run 66% over budget and 33% overtime. B. Flyvbjerg and A. Budzier also studied more than 1400 projects and revealed similar findings. According to ISPA and Standish group, Two-third of software projects fails to be delivered on time and within budget. These all facts and gaps give us a reason to pursue research in the field of agile estimation and in our specific case scrum estimation. A generic scrum estimation framework is shown in Fig 2.

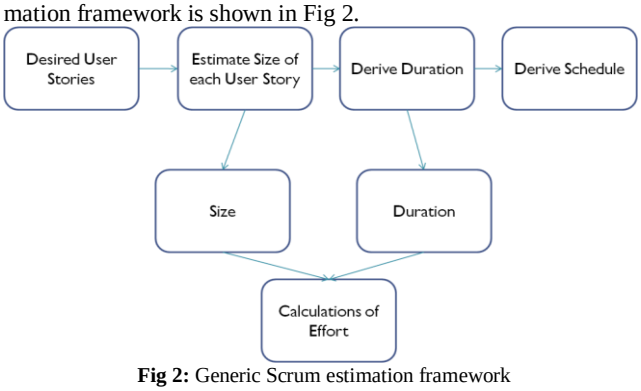

Fig 2: Generic Scrum estimation framework

A survey indicates out of Story Points (SP), Use Case Points (UCP), Function Points (FP), Object Points (OP), and Lines Of

Copyright © 2018Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Code (LOC), the most used approach by the IT industries is SP. The details are given in Table 1.

Table 1: Size metrics in agile based projects

Size estimation methods	Usage in industry
Story points	61.67%
Use case points	16.67%
Function points	28.33%
Object points	1.67%
Line of code	11.57%

Story point approach clearly wins in this context as per it's acceptability in the IT industry's working in agile based projects. The estimation techniques with their State of the Practice (SOTP) are given in Table 2.

Table 2: State of the practice for different estimation methods

Estimation techniques	Difficulties faced
Neural Networks	In Neural Networks, there is no recalibration support and thus has less performance as compared to the EJ techniques. The estimation depends on the nature of data and may change as the parameters adjusted. It does not have the explanation facility.
Expert Judgment	It is suffered from bias. An empirical technique that may work as the experts guess the effort based on their prior project experience. So, not a reliable estimation technique.
Planning Poker	Less literature available to comment on efficacy of planning poker but also lacks an analytical estimation.
Use Case Points	Product backlog in agile scrum does not satisfy some conditions in UCP docs.
Modified Use Case Point	Same problem as faced in UCP.
Linear regression	Need historical data which is missing in literature so may decrease in the accuracy.
Wideband Delphi	Accuracy affected by team's experience. An extension to EJ with a coordinator.
Bottom Up / Top Down	Not much evidence available in literature.

In the literature there are various effort estimation approaches used by researchers in estimation of different software process models. Their applicability varies from model to model. For heavy weight models, COCOMO has been extensively used and Story Point for Agile based estimation. Every estimation approach has its pros and cons. The detailed category, usage is tabulated in Table 3.

This literature survey clarifies the issues and the challenges faced by all estimation techniques. With the advancements in effort estimation techniques it is quite promising to opt for soft computing techniques for more accurate results. This research paper make use of Machine learning and Optimization techniques for an efficient effort and cost estimation. List of abbreviations used in this paper are given in Table 4.

Table 3: Comparison of effort estimation approaches

Estimation technique	Category	Usage	Pros	Cons
Estimation by Analogy	Formal estimation model / Non Algorithmic	Weighted micro function points	Estimation result is clear.	Gives unrealistic estimates, required previous existing projects
Planning poker	Expert estimation	Group estimation	Most popular	Less research done, so little empirical evidence available.
Expert Judgment	Empirical estimation	Educated guess based on past project experience	Fast result	Suffers from individual bias
Dephi estimation	Group estimation	Wideband Delphi	Collective opinion	No Analytic foundation
COCOMO and COCOMO-II	Heuristic approach	Parametric models	Clear results	Much data required for estimation.
Use Case points	Formal estimation model	Size based	Predicting initial estimates	Some conditions not met by product backlog.
Linear regression, Robust regression, Neural Nets (RBF)	Parametric	Wide spectrum	Suitable when existing data is available	Accuracy may decrease as less historical data available.
Neural nets (SVM)	Parametric	Wide spectrum	Do well when input data is distorted by high noise level	NN does not have any explanation facility.
Top Down (TD) estimation / Bottom Up (UP) estimation	Expert Estimation	Project management software	Good, if enough historical data available	Less empirical evidence, TD is better than BU.

Table 4: List of Abbreviations

ASD	Agile Software Development	ISPA	International Society of Parametric Analysis
NN	Neural Networks	SVM	Support Vector Machine
RBF	Radial Basis Function	MLP ANN	Multilayer Perceptron ANN

ANN	Artificial Neural Network	EJ	Expert Judgment
GLM	Generalized Linear Models	PRED	Percentage Relative Error Deviation
MMRE	Mean Magnitude of Relative Error	ISBSG	International Software Benchmarking Standards Group
FIS	Fuzzy Inference Systems	SP	Story Points
UCP	Use Case Points	MUCP	Modified Use Case Points
MDLEP	Multilayer Dilation-erosion-linear Perceptrons	BN	Bayesian Network
US	User Story	CART	Classification and Regression trees
RF	Random Forest	WD	Wideband Delphi

2. Related literature

Przemysław Pospieszny et. al. (2018) in their paper [1] make use of three approaches viz., SVM, MLPANN, GLM and their ensemble aggregation to estimate effort of agile based project using the ISBSG dataset and found that SVM outperform ANN and GLM in term of less MMRE and high PRED (25) and PRED (.30). Also, if log transformation applied to the dependent variable then GLM outperform ANN, but still SVM leads. Authors are saying result may vary because of the heterogeneous nature of ISBSG dataset and if applied on homogeneous set i.e., PROMISE or from source forge, then things may be different. Morakot Choetkiertikul, et. al (2017) in their paper [2] proposed the estimation of user stores instead of estimating the whole project effort using long short-term memory and recurrent highway network. They used data 16 open source projects data which contains 23313 user stories and as a result it concluded that random guessing, median and mean are not so efficient. Model is recurrent i.e., features will be same for all layers and avoids over-fitting. They made an end to end model in which words will be given as an input which passed to end to estimating the story points. With their approach they outperform random guessing, median and mean. Jasem M. Alostad et. al. (2017) in their paper [3] proposed an effort estimation model of Mamdani FIS type and took team's experience, story size, story complexity as an input to the Fuzzifier and estimation accuracy as to validate. They use the rules to predict story points required for the user stories or issues. MMRE is 0.28 and PRED is 50% which substantially increasing with the team's experience when applied to more sprints. Habibi Husain Arifin et. al., (2017) in their paper [4] proposed linear regression models for both effort – size and effort – time estimation with effort – size being a relative and effort – time an absolute estimate. They have collected data from Atlassian JIRA Repositories and provides an evidence based SE. As a future work, it can be applied for cognitive science. Saurabh Bilgaiyan et. al., (2017) in their paper [5] proposed a review of cost estimation in agile based software projects. They have worked on the various research questions to answer the most popular string i.e. what are most popular agile estimation approaches? In which environment it can be applied? What is the success and failure rate? Researchers found that Neural, Expert Judgement, Planning Poker, Linear regression, Wideband delphi, UCP and MUCP are widely searched. Murat Salmanoglu et.al., (2017) in their paper [6] compared the cosmic functional point with story point on the three industrial projects in agile context. As a result they have found that cosmic has better prediction as compare to story point with an underlying fact function points provide more objective estimates as compared to relative SP (which depends on team's experience). They claimed that regression models having cosmic FP as an independent variable outperforms SP. Ricardo de A. Araujo et. al., (2017) in their paper [7] proposed a hybrid MDELP model to deal with issues of software effort estimations. However, it is not applied in agile based projects, so can't say about effect. They use pessas ideas to decide dilation and erosion operators. They claimed the improvement in PRED for the hybrid approach as compared to the existing. Dragicevic Srdjana et. al., (2017) in their paper [8] discussed that the success of agile projects depends on the eliciting good user stories. They claimed that there proposed model can be applied in general to the agile projects regardless to the type. They took 160 projects data in their research. Their hybrid approach estimates the effort. They used RMSE as metric to check the deviation of actual effort to estimated effort. Vlad-Sebastian IONESCU et. al., (2017) in their paper [9] has done effort estimation for conventional methodologies using TF-IDF, SVR and GNB approaches The results seems to be promising when compared to the existing literature.

Maciej ŁABĘDZKI et. al., (2017) in their paper [10] presented a case study for agile estimation. They have considered a project OSW, TOPO system, FOODIE system and discussed the various issues associated with the same. They applied the agile estimation literature to all and found some significant inferences like Planning Poker has good results capability. Luigi Lavazza et. al., (2017) in their paper on [11] proposed a framework for constructing the indicators that can provide accurate accuracy of estimation models before they can be applied or used in practice. A discussion of various accuracy predication indicators has been made and used standardized accuracy measure for checking the model's accuracies. It is not yet for any agile estimation models. In the end, authors analyzed the different data set in the context of various models. Janeth López-Martínez et. al., (2017) in their paper [12] proposed a BN model in scrum context to determine the criteria for estimation with complexity and significance of user story as two ingredients. They used the correlation tests to validate the model. This model is going to assist all the fresher's moving to a scrum based projects. Furthermore, decomposing the complexity to experience, time and effort and important (US) to priority and value. So, created model that respond like experts. Mohd. Owais, R. Ramakishore. (2017) in their paper [13] proposed an approach for calculating the Effort, Duration and the Cost of agile based projects. This approach does not use any machine learning algorithm. The proposed algorithm is quite simple and no empirical evidence can be drawn that it will improve the present State-of-the-art. B. Prakash, V. Viswanathan (2017) in their paper [14] has done an extensive survey of all the agile estimation techniques with their popularity index i.e., what techniques have been used the literature eg, UCP, MUCP, Wideband Delphi to name a few. Seyyed Hamid Samareh Moosavi, Vahid Khatibi Bardsiri (2017) in their paper [15] proposed an optimization algorithm based on hybrid fuzzy interference system and claim to have more accurate results of effort predictions but not in agile context. The so called algorithm known as Satin bowerbird are used in generic with estimation models. The parameter that is going to input in the fuzzifier differs in context as compared to agile based softwares. Authors claim to provide optimized parameters with their bower algorithm to the fuzzy system. The data set used is ISBSG and the comparison is done with the existing techniques CART, etc and found 0.235 MMRE which is less as per group selected for comparison. Shashank Mouli Satapathy, Santanu Kumar Rath (2017) in their paper [16] discussed about improving the accuracy of effort with story point approach. Authors have used three machine learning algorithms viz.., Decision trees, Stochastic Gradient Boosting and Random forest and has done internal comparison and with the literature. They used 21 projects data from zia et. al paper for application. They then applied logarithmic transformations to the obtained dataset in order to normalize it. They provide story point count and velocity as an input to the ML model and Predicted effort will be the output. As a result SGB outperforms others. As a future work, BN can be used. Aditi Sharma, Ravi Ranjan (2017) in their paper [17] answered some of the alarming questions that what ANFIS used in effort estimation and what's there success rate? Explanations are not in the context of agile. Some hybrid approaches seems to give good results. As a future work it is rec-

ommended to use ANFIS with COCOMO, FP. Binish Tanveer (2017) in his paper [18] explained that before setting up for effort estimation it is must to have associated guidelines to have success chances. It is made for change impact analysis. A guideline framework is set after discussion with the experts. Sufyan Basri et. al., (2016) in their paper [19] suggested that non-algorithmic models are applicable for agile based projects. As in agile, requirements are volatile so a change is must to consider in the predicted effort and must be added to the final effort.. Saurabh Bilgaiyan et. al., (2016) in their paper [20] has designed a decade review of soft computing techniques used in agile effort estimation. As a result BN seems to be more promising with an accuracy rate of 62.8% as compared to other models like regression based models, composite models, expert judgment, planning poker but only in agile context. Anjali Sharma, Karambir (2016) in their paper [21] Empirical validation of random forest for agile software effort estimation based on story points has done comparison of Random forest algorithm with different types of neural networks and claimed that RF is giving better results than GRNN, PNN, CCNN, GMDH. They have taken story points as an input to the RF technique. Kayhan Moharreri et. al. (2016) in their paper [22] created an auto-estimate model for effort estimation in agile based projects. This model take input as dataset, do feature extraction, choose the features, and then performs cost estimate analysis. Different approaches have been applied like Random forest, PP, Naïve Bayes, LMT and their hybrids. Each results in a confusion matrix. As a result it has been found that hybrid outperforms PP. Binish Tanveer et al., (2016) in their paper [23] worked on effort estimation agile for an industrial case study. A survey conducted with three teams from SAP and found that impact change, team's experience and the complexity are major ingredients for affecting agile projects effort. Industry relies on PP and Story points heavily for estimation. Aditi Panda et. al., (2015) in their paper [24] compared various NN models for agile effort estimation based on S. Points. The models used are GRNN, PNN, GMDH, CCNN, and polynomial neural network. The dataset i.e. story point total, velocity and effort is taken from zia et.al paper. Then they portioned it into test and train set. They found CCNN outperformed all other with a PRED of 94%. As a future work, they propose SGB, RF with SP approach. Muhammad Usman et. al., (2015) in their paper [25] has done a SOTP survey to know the accuracy percentages from industry perspective. They collected data from sixty agile experts and found that PP is the most used estimation approach about 63%, followed by story point (62%). They also found the size metrics, cost drivers which are widely used and at which phase of SDLC. Hind Zahraoui, M Abdou (2015) in their paper [26] discussed scales, factors that influence user stories so they adjust it to make it more accurate. For this they decide the priority of story as multiplication of it urgency and value (in business). Create scales for it. Vachik S. Dave, Kamlesh Dutta (2015) in their review paper [27] has done analysis of decade of literature wherein NN are applied for effort estimation. They reviewed 21 papers and found some the major insights include comparison of NN with other techniques like PP etc., ANN models gives more accuracy than FP and SLIM. Size is an accuracy criterion for NN based models. No analysis done in agile context. Ali Bou Nassif et. al., (2015) in their paper [28] compared the effort for Radial basis function neural network (RBFNN), General regression neural network (GRNN), Cascade correlation neural network (CCNN) and Multilayer Perceptron (MLP) for non-agile based projects. Five datasets are used from ISBSG and four inputs were given to each system i.e., size, language, platform and source and found CCNN outperform others. Manga I, Blamah (2014) in their paper [29] suggested PSO framework which produces better results in terms of accuracy percentage. A comparison is done with adaptive learning, but facts are missing. Shashank Mouli Satapathy et. al., (2014) in their paper [30] proposed SVR techniques for enhancing the accuracy of effort estimation on the basis of SP approach. A comparison of all SVR kernel methods viz., SVR Linear Kernel, Polynomial Kernel, RBF kernel and sigmoid kernel carried out and found that RBF kernel outperforms others. Further enhancements can be performed using SGB, RF etc. Usman, M et. al., (2014) in their paper [31] found various estimation techniques like PP, UCP, MUCP, EJ, LR, NN(RBF) and there accuracy percentages. MUCP and NN seem good in terms of accuracy. In addition, various size metrics and cost drivers has also been identified which includes Team's experience, task size, test efficiency and risk factors as cost drivers and SP, UCP, class point as size metrics. As per the nature of Agile based projects, it has been found that estimations and plans also need to be done progressively. Studies reveal that group consensus estimates were less optimistic and more accurate than statistical combination of individual estimates. During investigation, following estimation techniques found viz., EJ, PP, UCP, MUCP, LR, RR, Neural Nets(RBF, SVM), Constructive Agile Estimation Algorithm, WD, wherein the estimation accuracy parameter increases in the following order MUCP, NN, EJ, LR, PP. Story Points are the most used size metric as compared to UCP, LOC, Function Point in Agile. Studies have shown that soft computing methods are becoming quite suitable for handling problems like cost estimation, optimization, machine learning, forecasting, etc. Many soft computing techniques like Mamdani FIS, ANNs (i.e. Regression Neural Networks, Radial Basis Functions (RBF), Counter Propagation Neural Network (CPNN), etc.), Bio-Inspired Techniques (i.e. PSO, etc.) are being applied successfully for estimating cost and effort in agile software development environment.

3. Proposed research methodology

The proposed research methodology helps to fill the gaps in the literature and promises a better estimation in Agile based projects. It has been broadly divided in three categories viz., Data Preparation, Data Set Partitioning and Model Selection and Testing Part. The above stated categories will make use of different CASE tools and techniques to complete the process of estimation. As per the literature context it is evident that a machine learning models yields more acceptable results as compared to traditional or non-machine learning models. The steps are given below:

3.1. Data preparation

There are online data repositories like ISBSG data sets and Atalassian JIRA repositories which contains Agile and traditional projects data. As the data is heterogeneous so it is required to be filter for Agile projects as per the steps below:

a) Data Understanding: Data understanding is a crucial phase and pertains to a better data extraction from the repositories.

b) ISBSG data sets/ Atalassian JIRA repositories: The data in heterogeneous form will be extracted from the online repositories. The data from these sources are real projects data and may not be uniform. It cannot be used as such for training purposes. As data plays a significant and indispensible role w.r.t. estimating so it must be standardized.

c) Quantify and process missing data: The next level of standardizing data is to quantify and process any missing data.

d) Determine Agile data using discriminate Analysis: Based on the attributes and unique characteristics of Agile based projects, scrum data will be extracted from the data pool.

e) Agile development sample set: The sample set is ready.

f) Calculate correlation and coefficient matrix

g) Calculate Eigen value of correlation coefficient matrix

h) Determine the number of principle components

3.2. Data set partitioning and model selection

Standardized data from the previous step will be partitioned into training and test data sets using K-fold cross validation. Create and train the ANFIS module using SVM as a training algorithm. The same can be viewed in Fig 3 as given below:

a) Create a base fuzzy system

b) Get parameters of base fuzzy system

c) Carry out the optimization of new parameters using Satin Bower Bird optimization algorithm

d) Insert new parameters in the base fuzzy system

e) Classify data and inference results.

f) Estimate the effort and cost

g) Evaluate using metrics MMRE and PRED.

h) Parameters are optimized.

3.3. Testing part

After the cross validation, the testing data will then be used and can be viewed in Fig 4 as given below:

a) Create a base fuzzy system for testing data

b) Set parameter of membership function using the optimized parameters obtained in the previous step.

c) Classify data and inference results.

d) Estimate the effort and cost

e) Evaluate using metrics MMRE and PRED.

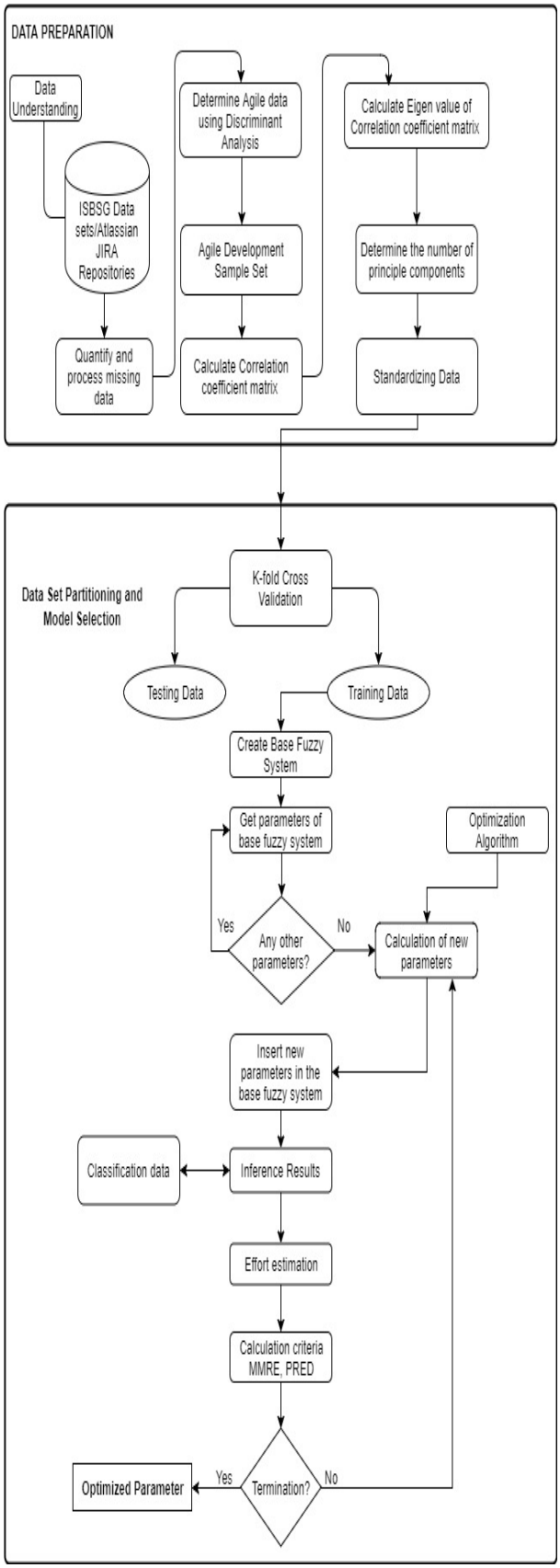

Fig 3: Proposed methodology-I

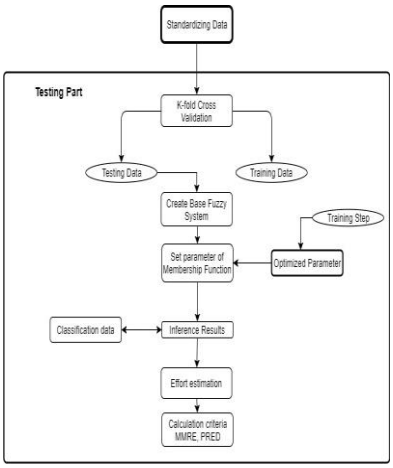

Fig 4: Proposed methodology-II**.**

4. Conclusion

In this paper, various research papers have been revisited, generalized and formalized and it has been found that proposed research work yields promising results for effort and cost estimation in agile software. As a future work, there are some factors like regression test efforts, software architecture erosion effort, people factors, project factors and its accelerating and decelerating factors are not exactly identified in literature and can be added to improve the results.

Acknowledgement

I would like to express my sincere gratitude to Dr. Sahil Verma for his continuous support throughout my literature work. His guidance helped me a lot to address all the issues and challenges that came in my way. My family has always been a support to me. It is because of them I have been able to get time for research.

References

[1] P. Pospieszny, B. Czarnacka-Chrobot, and A. Kobylinski, "An effective approach for software project effort and duration estimation with machine learning algorithms," *J. Syst. Softw.*, vol. 137, pp. 184–196, 2018.

[2] M. Choetkiertikul, H. K. Dam, T. Tran, T. T. M. Pham, A. Ghose, and T. Menzies, "A deep learning model for estimating story points," *IEEE Trans. Softw. Eng.*, vol. 14, no. 8, pp. 1–1, 2015.

[3] J. M. Alostad, L. R. A. Abdullah, and L. S. Aali, "A Fuzzy based Model for Effort Estimation in Scrum Projects," *IJACSA) Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 9, pp. 270–277, 2017.

[4] H. H. Arifin, J. Daengdej, and N. T. Khanh, "An empirical study of effort-size and effort-time in expert-based estimations," *Proc. - 8th IEEE Int. Work. Empir. Softw. Eng. Pract. IWESEP 2017*, pp. 35–40, 2017.

[5] S. Bilgaiyan, S. Sagnika, S. Mishra, and M. Das, "A systematic review on software cost estimation in Agile Software Development," *J. Eng. Sci. Technol. Rev.*, vol. 10, no. 4, pp. 51–64, 2017.

[6] M. Salmanuglu, T. Hacaloglu, O. Demirors "Effort Estimation for Agile Software Development : Comparative Case Studies Using COSMIC Functional Size Measurement and Story Points," 12th International Conference on Software process and product meansurement, pp. 9 pages, 2017.

[7] R. de A. Araújo, A. L. I. Oliveira, and S. Meira, "A class of hybrid multilayer perceptrons for software development effort estimation problems," *Expert Syst. Appl.*, vol. 90, pp. 1–12, 2017.

[8] S. Dragicevic, S. Celar, and M. Turic, "Bayesian network model for task effort estimation in agile software development," *J. Syst. Softw.*, vol. 127, pp. 109–119, 2017.

[9] V.-S. Ionescu, H. Demian, and I.-G. Czibula, "Natural Language Processing and Machine Learning Methods for Software Development Effort Estimation," *Stud. Informatics Control*, vol. 26, no. 2, pp. 219–228, 2017.

[10] M. Łabędzki, P. P. Romiński, A. Rybicki, and M. Wolski, "Agile effort estimation in software development projects – case study," *Cent. Eur. Rev. Econ. Manag.*, vol. 1, no. 3, pp. 135–152, 2017.

[11] L. Lavazza and S. Morasca, "On the Evaluation of Effort Estimation Models," *Proc. 21st Int. Conf. Eval. Assess. Softw. Eng. - EASE'17*, pp. 41–50, 2017.

[12] J. López-Martínez, A. Ramírez-Noriega, R. Juárez-Ramírez, G. Licea, and S. Jiménez, "User stories complexity estimation using Bayesian networks for inexperienced developers," *Cluster Comput.*, pp. 1–14, 2017.

[13] M. Owais and R. Ramakishore, "Effort, duration and cost estimation in agile software development," *2016 9th Int. Conf. Contemp. Comput. IC3 2016*, pp. 1–5, 2017.

[14] B. Prakash and V. Viswanathan, "A survey on software estimation techniques in traditional and agile development models," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 7, no. 3, pp. 867–876, 2017.

[15] S. H. Samareh Moosavi and V. Khatibi Bardsiri, "Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation," *Eng. Appl. Artif. Intell.*, vol. 60, pp. 1–15, 2017.

[16] S. M. Satapathy and S. K. Rath, "Empirical assessment of machine learning models for agile software development effort estimation using story points," *Innov. Syst. Softw. Eng.*, vol. 13, no. 2–3, pp. 191–200, 2017.

[17] A. Sharma and R. Ranjan, "Software Effort Estimation using Neuro Fuzzy Inference System : Past and Present,", pp. 78–83, 2017.

[18] B. Tanveer, "Guidelines for utilizing change impact analysis when estimating effort in agile software development," *ACM*, 2017.

[19] S. Basri, N. Kama, F. Haneem, and S. A. Ismail, "Predicting effort for requirement changes during software development," *Proc. Seventh Symp. Inf. Commun. Technol. - SoICT '16*, pp. 380–387, 2016.

[20] S. Bilgaiyan, S. Mishra, and M. Das, "A Review of Software Cost Estimation in Agile Software Development Using Soft Computing Techniques," *2nd Int. Conf. Comput. Intell. Networks*, pp. 112–117, 2016.

[21] S. Anjali, Karambir., "Empirical Validation of Random Forest for Agile Software," vol. 5, no. 7, pp. 1437–1446, 2016.

[22] K. Moharreri, A. V. Sapre, J. Ramanathan, and R. Ramnath, "Cost-Effective Supervised Learning Models for Software Effort Estimation in Agile Environments," *2016 IEEE 40th Annu. Comput. Softw. Appl. Conf.*, pp. 135–140, 2016.

[23] B. Tanveer, L. Guzmán, and U. M. Engel, "Understanding and improving effort estimation in Agile software development- an industrial case study," *Proc. Int. Work. Softw. Syst. Process - ICSSP '16*, pp. 41–50, 2016.

[24] A. Panda, S. M. Satapathy, and S. K. Rath, "Empirical Validation of Neural Network Models for Agile Software Effort Estimation based on Story Points," *Procedia Comput. Sci.*, vol. 57, pp. 772–781, 2015.

[25] M. Usman, E. Mendes, and J. Börstler, "Effort estimation in Agile software development: A survey on the state of the practice," *ACM Int. Conf. Proceeding Ser.*, vol. 27–29–Apri, 2015.

[26] H. Zahraoui and M. A. Janati Idrissi, "Adjusting story points calculation in scrum effort & time estimation," *2015 10th Int. Conf. Intell. Syst. Theor. Appl. SITA 2015*, 2015.

[27] V. S. Dave and K. Dutta, "Neural network based models for software effort estimation: A review," *Artif. Intell. Rev.*, vol. 42, no. 2, pp. 295–307, 2014.

[28] A. B. Nassif, M. Azzeh, L. F. Capretz, and D. Ho, "Neural network models for software development effort estimation: a comparative study," *Neural Comput. Appl.*, vol. 27, no. 8, pp. 2369–2381, 2016.

[29] I. Manga and N. V. Blamah, "A particle Swarm Optimization-based Framework for Agile Software Effort Estimation," *Int. J. Eng. Sci.*, vol. 3, no. 6, pp. 30–36, 2014.

[30] S. M. Satapathy, A. Panda, and S. K. Rath, "Story Point Approach based Agile Software Effort Estimation using Various SVR Kernel Methods," *26th Int. Conf. Softw. Eng. Knowl. Eng.*, pp. 304–307, 2014.

[31] M. Usman, E. Mendes, F. Weidt, and R. Britto, "Effort estimation in Agile Software Development: A systematic literature review," *ACM Int. Conf. Proceeding Ser.*, pp. 82–91, 2014.Lee JR, Kim SA, Yoo JW & Kang YK (2007), The present status of diabetes education and the role recognition as a diabetes educator of nurses in korea. Diabetes Research and Clinical Practice 77, 199–204.

.