Software Development Framework on Small Team using Agile Framework For Small Projects (AFSP) with Neural Network Estimation

Wuwanjie Septian
Master of Computer Science
STMIK Nusa Mandiri
Jakarta, Indonesia
wuwanjie_septian@yahoo.com

Windu Gata
Master of Computer Science
STMIK Nusa Mandiri
Jakarta, Indonesia
windugata@gmail.com

***Abstract*—Technology development in growing companies is important because it has great potential to reduce cost, time of work, efficiency, flexibility. In small developer team, finishing many software development projects with tight schedule is a challenge to work efficiently and effectively. Wrong estimation could delay the other project completion and lower the team's assessment point. This study focuses on developing framework that fits well with small-scale developers by combining Agile Framework For Small Projects (AFSP) and Neural Network estimation. The research method is to identify 5 risks based on the agility factor of each project. Then determine the appropriate agile practices. Next, do the selection of dataset agile characteristics, and calculate the Neural Network estimation. Implementation is done with estimation result guidance. The project that has been done is assessed to measure the level of agility of project result. Then, the result of project's succession and accuracy are analyzed. This research is implemented on six projects in fast moving consumer goods (FMCG) company. The results showed all projects are agile, four projects with small Mean Relative Error (MRE) values and two projects with big MRE values.**

Keywords— Software Engineering, Agile Development, Agile Framework, Software Process Improvement, Neural Network

I. INTRODUCTION

Technology development in growing companies is important because it has great potential to reduce cost, time of work, efficiency, flexibility and service level [4]. The Company will strive to capitalize the advantages of updated technology, with the aim to enchancing sales capability, efficiency and savings on all sides of the company. The existence of IT departments within a company becomes important to support the development & maintenance of technology in the long term.

Understanding user needs is a crucial factor in a small team of developers, failure to understand is a failure factor in project development [6]. In small developer team, finishing many software development projects with tight schedule is a challenge to work efficiently and effectively. Wrong estimation could delay the other project completion and lower the team's assessment point.

AFSP has been tested as an effective software development framework on very small teams, resulting a focus on managing multiple projects and can be used on many types of projects. Even though, the framework has a lack of project experience and observation on teams needs [10].

Estimation of software development are the solution for the problems of software development which become more complex day to day [21]. Each technique have advantages and disadvantages, therefore the use of several estimation approaches needs to be done to get near-reality estimation [20].

The use of machine learning can be positive or negative depending on the parameters used [19]. The Neural Network technique is effectively used as an in-depth learning model to estimate [1][12]. The open source dataset can be used to help development team estimation [18]. This option can be triggered by the incomplete documentation of the same type project, and the commercial dataset which often requires a purchase fee. The use of open source data projects can be good by filtering the suitability of data before processing [9].

After observation in one of multinational FMCG company, several issue regarding the development issue in small team: wrong project estimation, failure to adjust timeline when the project scope changes, the development schedule is too tight, incomplete project specifications, and unclear user needs.

Knowing the obstacle regarding the agile development, this research focuses on developing framework that fits well with small-scale developers by combining AFSP and Neural Network estimation with reseach question:

- How is the result of using AFSP framework combine with neural network on the estimation of project completion?
- Does the AFSP ftramework that combined Neural Network suitable for calculation of project estimation in small teams?

978-1-5386-2827-0/17/$31.00 ©2017 IEEE

II. BACKGROUND

A. Agile Development

Agility becomes the main concern when describing the modern software process [5]. Agile Software Engineering combines a philosophy and set of software development guides. The essence of the agile software development philosophy consists of several elements, including: performance that encourages increased user satisfaction, delivery of results on an ongoing basis, small team with high motivation, informal method, and the simplicity of overall project development [14].

The agile development model itself has four important values that need attention to meet the users need and developers; communication, simplicity, feedback, and courage. Knowing each characteristic can help developer while implementing agile process [7].

The training dataset used to calculate the estimation of AFSP project work is taken from previous research [22], 21 projects developed by 6 software development companies. The project built with agile development methodologies of each project has a level of difficulty with varying speed of work.

B. Agile Framework for Small Projects

AFSP is an effective software development framework for small teams, resulting in a focus on managing multiple projects and can be used on many types of projects [10]. This Framework uses SCRUM techniques as a center and produces a structured process that provides guidance to software developers in adopting agile practices.

The AFSP framework consists of two main components: The AFSP process, which helps to build the correct software or system; AFSP practice pool, helps to determine the techniques and tools to build the software or system correctly. In addition, AFSP combines four evaluations: Risk assessment model, which can be used as a starting point for agile evaluation and selection of agile projects; 4-Dimensional Analytical Tool (4-DAT) which can be used to measure agile agility; Agile Adoption and Improvement Model (AAIM), is used to determine the adoption of agile techniques or improvement rates, as well as to assess a particular level of agile adoption and to advance to the next level; Critical Success Factors (CSF), is used to measure the success rate of a project based on the evaluation calculation of CSF attributes with the equation:

$$DAPS(Project) = \sum_{i=1}^{m} W_i \times \frac{Sum_i}{N_i} \quad (1)$$

M is CSF number and CSF = {DS, AT, TC, PM, TE, CI} components. N is the number of attributes in each CSF, T is the total attribute value and the constant W represents the weight of each CSF. Assessment of 25 attributes is evaluated by a value of 0 or 1 (failed or passed) on each project that has been done.

C. Neural Network Estimation

A neural network is a massively parallel distributed processor made up of simple processing units that has a natural propensity for storing experiential knowledge. Neural network is a great estimator tools, knowledge of this model is acquired through a learning process from its environment [23]

Neural network that used for estimation in this research was Multilayer Perceptrons (MLP). MLP neural network is a feed-forward neural network that has 1 input layer, 1 or more hidden layer, and 1 output layer [24]. The main reason MLP chosen because this model was known as a model which can handle complex learning and have capability to learn how to do job according to training dataset in real-time and support non linear models [23].

Backpropagation algorithm is most known training algorithms in MLP algorithm. This algorithm focus on measuring the output error margin and calculate the weights on to decrease the error. The advantages of this function is providing a computationally efficient method for the training steps [24]. Sigmoid function is part of backpropagation algorithm that help determine appropriate weights each hidden layer connection based on the each contribution node to the output layer, here is the sigmoid function equation:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

x is nodes weight and e was exponent factors.

D. State of The Art Research

This study uses previous research references that have links to agile software development, understanding the determinants of a project's success, and the use of data mining as a reference for early developers who are new with agile development. The study review can be seen in Table I.

Here is summary of background research point. The lack of agile approaches that work well as single or independent systems [10], Least practitioners who can use agile approaches quickly and effectively when the transition of software development methods [16]. The difficulty of assessing the success of agile software development projects objectively [2]. Few research that can define preconceptions of agility and how the evaluation of agile methods should be done [15]. The inaccuracies of the traditional business estimation approaches used to estimate agile software projects [17]. The limitations of accuracy of algorithmic estimation techniques due to the loss of factors to account for differences between cost drivers [8]. Authors combines the advantages of each methods to answer this research questions after evaluate background point, methodology, and the result from these previous research.

TABLE I. LITERATURE SUMMARY

Author	*Methodology*	*Result*
[12]	Development of AFSP framework	AFSP framework is effective in agile project completion
[17]	Evaluation of success degree and agile adoption, AAIM	AAIM is successful in assisting the understanding of agile method exploration
[2]	A survey study regarding key success factors of agile software development projects	Surveys result three success factors and three additional factors to ensure project success
[16]	Development of an analytical framework to assess the degree of agility of agile practices	Agility level useful for constructing methodologies and comparing the agile-traditional

Author	*Methodology*	*Result*
	carried out. 4-DAT	methods.
[22]	Development of a suitable effort estimation model for agile projects.	The developed model has good estimation accuracy.
[9]	Comparison of algorithmic estimation with data mining estimation technique	Data mining techniques improve the accuracy of estimates in many cases.

E. Research Methodology

The method in Fig. 1 is combination AFSP framework with neural network estimation. The flow described as follows:

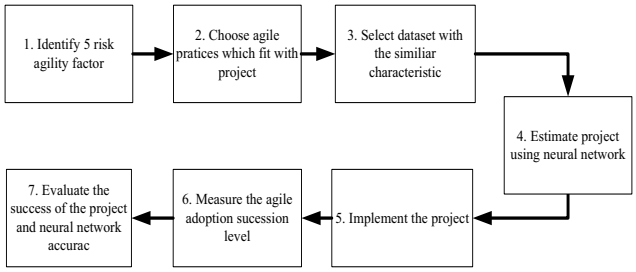

Fig. 1. Method Flow

1) Identify 5 risk agility factor: The team identifies five risk based on agility factors that influence the selection of agile methods. Size, Criticality, Dynamism, Personnel, Culture factor need to be identified to improve the ability of project team members to understand the development environment, organizational capability, identification and collaborating with the person initiating the project.

2) Choose agile pratices which fit with project: The project team determines agile steps in which their strength can be applied maximally and with consideration of the risk of minimal project failure. The AFSP framework has agile steps that include data, critical success factors of the project, agile steps contained in SCRUM, XP, FDD, DSDM, Crystal Clear and 4-DAT can be used to measure the agility of each stage if necessary.

3) Select dataset with the similiar characteristic: The project team selects the open source dataset or project history that has been done according to the agile characteristics. The result of dataset quality depends on how many characteristics are used for selection.

4) Estimate the project using neural network: The project team estimate the project completion schedule using the MLP neural network on the selected dataset and specifies the target for each of the project's sprint stages.

5) Implement the project: The project team focuses on project development and implementation by following the AFSP process from start to finish project. The specific results of each process will depend on the developer ability to organize and the experience in the field of application. At this stage, the management team must constantly monitor the performance of the selected agile steps while keeping the project completion accurately in each sprint based on the neural network estimation.

6) Measure the agile adoption sucession level: The project team assesses AAIML in any software development project. It is to assess how effective the agility process and the steps taken within an organization. The adoption model can also help to build agile software that requires systematic development.

7) Evaluate the success of the project and neural network accuracy: The project team quantitatively measures the Degree Agile Project Success (DAPS) based on CSF studies in software projects. CSF can be defined as a factor that must exist agile projects for the project succession. DAPS shows how well the performance of project organization to achieve success. This step helps to identify and measure organizational performance and review the accuracy of neural network used for future project improvement.

III. IMPLEMENTATION AND RESULT

A. Identify 5 risk agility factor Agile Development

Risk identification was perform on 6 projects in one of multinational FMCG company. The project spesification detail displayed in Table II. The Identification result a similar trend pattern, imperfect small pentagon as shown in Figure 2. A narrow tendency of the pentagons indicates agile development is a good option on the six projects [3].

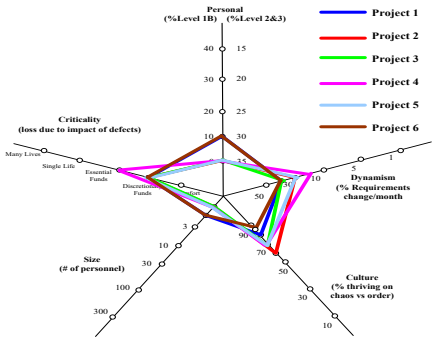

Fig. 2. The Five Risk-based Agility Factors of The Six Projects

TABLE II. PROJECT SPESIFICATION

Context Factor	*P1*	*P2*	*P3*	*P4*	*P5*	*P6*
Team Size	2	1	1	1	1	2
Platform	Desk top	Desk top	Web	Desk top	Desk top	Desk top
User Stories	30	15	12	20	7	13
Project Type	New	New	New	Extend	Extend	Extend
Complexity	High	Mid	Low	High	Low	High
Communication	Face to face, email, telephone					
Time Completion (week)	12	3	1	4	1	5

The team personnel consists of 2 person, 1 developer with 1B understanding level and 1 as developer and system analyst level 2. The more complex project is done by the two persons and for the simple project one person.

The dynamics factor, the changes of user requirement on the six projects has interval around 20-40%. Several factors are causing changes of the user requirement: the explanation of the business process is not clear or incomplete, complex workflow and inefficient calculation formula, user do not understand the standard operating procedure (SOP), the project culture interval is 60-85%. This means the development team is more likely to have tendency thrive on chaos than thrive on order.

The size of the development team consist 2 person who can work together as a team or individual. 4 projects are handle as individual projects and 2 project are handle as team project. Even though some project are individual projects, the business process still communicates within the team to make sure the user needs well captured.

The criticality factor has been analysed. 1 project impact essential funds if the project failed. The other projects impact only discretionary funds. Overall, the criticality factor from these project suitable with agile approach which no need high attention of risk.

B. Choose agile pratices which fit with project

Initial review is held with the users. The objectives to determine the agile practices which fit with project. The result after discussion and develepor team discussion, all projects need 10-14 agile practices. Table III describes the agile practices used in project completion.

TABLE III. AGILE PRACTICE EACH PROJECT

Type	*Agile Practice*	*P1*	*P2*	*P3*	*P4*	*P5*	*P6*
Plan	Planning game	X	X	X	X	X	X
Plan	Product backlog	X	X	X	X	X	X
Plan	Sprint backlog	X	X	X	X	X	X
Design	Simple Design	X	X	X	X	X	X
Develop	Coding Standard	X	X		X	X	X
Develop	Regular builds	X	X	X	X	X	X
Develop	Configuration Management						
Develop	Reversible changes	X	X	X	X	X	X
Develop	Pair Programming	X					X
Develop	Collective Ownership		X				
Develop	Refactoring		X		X		
Develop	Continuos Integration	X	X	X	X		X
Test	Test first programming	X	X	X	X	X	X
Test	Integrated testing	X	X	X	X	X	X
Project Management	Sprint review	X	X	X	X	X	X
Organizational Development	Empowered SCRUM						
Project	SCRUM Master						
Management							
Project Management	Sprint planning meeting	X	X	X	X	X	X
Project Management	Daily SCRUM meeting	X		X	X	X	X
Project Management	Burn-down chart	X			X		
Total		15	14	12	15	12	14

Each project has different levels of difficulty and priority. The developer team choose to prioritize project 1 and project 4 with consideration the project has a great efficiency of working time & reduced human power for the company. Therefore, the development team use more agile practices on the two projects.

C. Select dataset with the similiar characteristic

The training dataset used to calculate the estimation of AFSP project. The training dataset is taken from the previous research [22]. The attributes have been filtered from 13 and remaining 6 attributes. Attributes which related about salary cost are not used for calculation because the development team came from internal company developer. The attributes are effort, Initial velocity, deceleration. sprint size, work days, actual time, estimate time, time MRE. The estimate time attribute will be calculated at a later stage, the actual time and MRE time will be calculated at the evaluation stage.

D. Estimate the project using neural network

Estimation are performed on the testing dataset based on the training data. MLP neural network trained by a back propagation algorithm with configuration training cycles value 500, learning rate value 0.3, momentum value 0.2 and error optimisation below 0.00001.

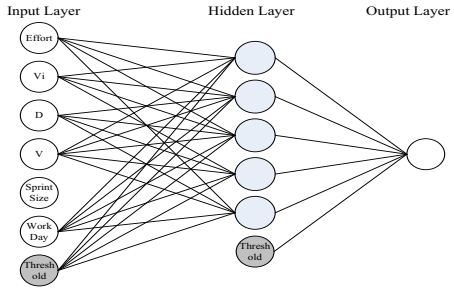

Fig. 3. Neural Network Estimation Layer

The input layer consists of 6 nodes that selected with similiar characteristics. The hidden layer consist of 1 hidden layer with 5 sigmoid nodes, each of nodes have different bias value. Table IV describes the result of MLP neural network estimation from testing dataset.

TABLE IV. RESULT OF NEURAL NETWORK ESTIMATION

P No	*Effort*	*Vi*	*D*	*V*	*Sprint Size*	*Work Day*	*Est Time*
1	168	2.625	0.742	2.0	8	20	69.39478
2	84	3.8	0.773	2.8	4	20	26.82393

P No	Effort	Vi	D	V	Sprint Size	Work Day	Est Time
3	67	3.9	0.859	3.2	4	20	8.302631
4	112	4.6	0.903	4.0	4	20	4.535079
5	39	4	0.903	3.5	4	20	6.70476
6	72	3.7	0.742	2.6	4	20	28.43741

After the calculation is done, the estimation results of each project obtained. The estimation on project 1 is 69 days, project 2 is 26 days, project 3 is 8 days, project 4 is 4 days, project 5 is 6 days and project 6 is 28 days. The estimation results are used as target of completion and discussed at each SCRUM meeting.

E. Implement the project

The six projects are implemented based on management priorities. Projects 1 and 6 became the focus of management, thus using more agile practices than other projects.

In this implementation stage, the work procedure of each project is same. Project specification that collected after meeting with users recorded as planning game. Planning games are detailed into the several target of achievement, called product backlog. The developer team selects one of the priorities features and set it as the sprint backlog. The first sprint target is to completing the first sprint backlog and keep iteratively in each sprint to completing another sprint backlog. In each sprint iteration, the team always put attention to the remaining development time that calculated from the previous neural network estimation.

F. Measure the agile adoption sucession level

The team assess each project as an agile evaluation with objective to know how much the agile adoption level using AAIML framework. Project 4 and 5 are examples of projects that have readiness applied in AFSP. In experience, the development team has a deeper experience of how the two projects work. The criticality and complexity of the project is relatively moderate. Although the project 5 sprint meeting is done by online meeting, there are no significant obstacles in the evaluation and sprint backlog planning.

The results of agile adoption assessment and project progress rate can be seen in Table V. Team readiness & lack of experience in agile method is the weakness point in the team. As a result, projects 1, 2, 3 and 6 only focus on level 1, while projects 4 and 5 focus on level 2.

TABLE V. PROJECT AGILE ADOPTION AND IMPROVEMENT LEVEL

Apex	AAIML 6 : Agile Progress	AAIML 6: Lean Production, Keep Agile	
		• Quality production	
		• Minimal possible resources	
		• Keep the process agile	
	AAIML 5 : Agile Smart	AAIML 5: Learning	
		• Research and lesson learning	
		• Learning management	
		• Application of the learnt lessons	
Crux	AAIML 4 : Agile	AAIML 4: People-Oriented	
		• Value people	
		• Do not ignore processes and tools	
	Value	• Keep intact the valued-people	
	AAIML 3 : Agile Realization	AAIML 3: Executable Artifacts	
		• Executable software versions	
		• Minimal documentation	
		• Encourage Minimal documentation	
	AAIML 2 : Agile Initial	AAIML 2: Communication-oriented	P 5 P 4
		• Communication -focused	
		• Cooperative and Collaborative	
		• Face to face communication	
Prompt	AAIML 1 : Agile Infancy	AAIML 1: Speed, Flexibility, Responsive	P 1 P 2 P 3 P 6
		• Iterative and incremental	
		• Encourage and accommodate change	
		• Reflect changes	

G. Evaluate the success of the project and neural network

This stage is evaluated by calculating the project success degree. The project success degree is calculated based on the CSF component, DAPS. The degree of project success of each project displayed on table VI.

TABLE VI. CSF SIX PROJECT ATTRIBUTE

Succes Factor	Value	Attibute	P 1	P 2	P 3	P 4	P 5	P 6
Delivery Strategy	0.3	DS1	0	1	1	1	1	0
		DS2	1	1	1	1	1	1
		sum	1	2	2	2	2	1
Agile software engineering techniques	0.2	AT1	1	0	1	1	1	1
		AT2	1	1	1	1	1	1
		AT3	1	0	0	1	0	1
		AT4	1	1	1	1	1	1
		AT5	1	0	0	0	1	1
		sum	5	2	3	4	4	5
Team capability	0.2	TC1	0	1	1	1	1	1
		TC2	1	1	0	1	1	1
		TC3	1	1	1	1	1	1
		TC4	1	1	1	1	1	1
		TC5	0	0	0	0	0	0
		sum	3	4	3	4	4	4
Project management process	0.1	PM1	1	1	1	1	1	1
	0.1	PM1	1	1	1	1	1	1
		PM2	1	1	1	1	1	1
		PM3	1	1	1	1	1	1
		PM4	0	1	1	1	1	1
		PM5	1	1	1	1	1	1
		PM6	1	0	0	0	0	0
		sum	5	5	5	5	5	5
Team environment	0.1	TE1	1	0	1	0	0	0
		TE2	1	1	1	1	1	1
		TE3	1	1	1	1	1	1
		TE4	1	1	1	1	1	1
		sum	4	3	4	3	3	3
Customer involvement	0.1	CI1	1	1	1	1	1	1
		CI2	1	1	1	1	1	1
		CI3	0	0	0	1	0	0
		sum	2	2	2	3	2	2
DAPS (Project) Score			0.72	0.76	0.79	0.87	0.84	0.73

The DAPS of projects 1, 2, 3 and 6 have intervals above 0.7 and below 0.8, whereas only projects 4 and 5 have DAPS values above 0.8. Based on the quantitative evaluation of Lee and Yong's research, the interval above 0.7 can be identify as a successful agile project [11]. In reality, projects built and implemented on target time, with support from neural network

estimation. The results of actual project duration and estimation are shown in Table VII.

TABLE VII. ACTUAL AND ESTIMATION PROJECT RESULT

P No	*Actual Time*	*Estimated Time*	*Estimated Month*	*Time MRE*
1	82.09474	69.39478	3.469739	15%
2	29.92999	26.82393	1.3411965	10%
3	20.81375	8.302631	0.4151316	60%
4	28.23239	4.535079	0.226754	84%
5	11.15333	6.70476	0.335238	40%
6	27.27262	28.43741	1.4218705	4%

Projects 1, 2, 5, and 6 have ≤ 40% MRE values (15%, 10%, 40% and 4%). While on projects 3 and 4 have ≥ 60% MRE values (60% and 80%) .The error rate of the entire project has an average 34%. This result show the time and effort estimation using neural network still not very accurate, requires an evaluation of the agile practices and method improvement as a future work.

IV. CONCLUSION

The suitable software development framework with company's condition becomes important. The suitable methods have great potential to reduce cost & work time, quality improvement, efficiency, flexibility and others intangible benefits such as user trust, team trustworthiness.

After this novel framework implementated,. The authors summarize the following conclusion:

- The results of using the developed framework successfully assess all projects as agile with the DAPS above 0.7. 4 projects have a MRE value of ≤ 40%, 2 projects have a large MRE value, ≥ 60%.
- The AFSP framework with neural network is not yet fit well with all project due to high of MRE in 2 projects. Evaluation needs to be done in terms of agile approach understanding, technical issues, and estimation method.

Improvement is needed to to make the model works well. The development of agile method estimation with deeper data mining neural network analysis needs to be done such as General Regression Neural Network (GRNN) [1], long short-term memory (LSTM) and recurrent Highway network [13]. Compile the dataset using local developer agile projects. The working culture of agile developers in each country certainly has unique and different characteristics, that be more easier for development team to adapt.

REFERENCES

[1] M. Choetkiertikul et al. “A deep learning model for estimating story points”. arXiv:1609.00489v2 [cs.SE] 6 Sep 2016.

[2] T. Chow, D..B. Cao. “A survey study of critical success factors in agile software projects”. Elsevier The Journal of Systems and Software 81, 2008, pp. 961–971.

[3] A. Cockburn. “Agile Software Development - Software Development as a Cooperative Game 2nd edition”. Boston:Pearson Education. 2007. pp. 19-20.

[4] T. Davenport. “Process Innovation : Reengineering Work Through Information Technology”. Harvard Business School Press. (2011). pp. 1.

[5] I. Jacobson. “A Resounding ‘Yes’ to Agile Processes—But Also More”. Cutter Business Technology Journal, vol. 15, no. 1, January 2002, pp. 18–24.

[6] R. Kaur, J. Sengupta. “Software Process Models and Analysis on Failure of Software Development Projects”. International Journal of Scientific & Engineering Research Volume 2, Issue 2, February-2011 ISSN 2229-5518 pp 1-4.

[7] K.E. Kendall, J.E. Kendall. “Systems Analysis And Design Eight Edition”. New Jersey: Pearson. 2011. Pp 163.

[8] Z.A. Khalifelu, F.S. Gharehchopogh. “Comparison And Evaluation Of Data Mining Techniques With Algorithmic Models In Software Cost Estimation”. Elsevier Information and Software Technology 54, 2012. pp. 41–59.

[9] E. Kocaguneli, G. Gay, T. Menzies, Y. Yang, J.W. Keung. “When To Use Data From Other Projects For Effort Estimation”. Proceedings of the IEEE/ACM international conference on Automated software engineering, 2010 September 20 - 24, pp 321-324 .

[10] S. Lee, H.S. Yong. “Agile software development framework in a small project environment”. Journal Information Process System Vol.9, No.1, March 2013, pp 69-88.

[11] S. Lee, H.S. Yong. “Distributed Agile: Project Management in a Global Environment”. Empirical Software Engineering, Vol.15, No.2, 2010, pp.204-217.

[12] A.B. Nassifi, M. Azzeh, L.F. Capretz, D. Ho. “Neural Network Models For Software Development Effort Estimation: A Comparative Study”. Springer Neural Computing and Applications November 2016, Volume 27, Issue 8, pp 2369-2381.

[13] A. Panda, S.M. Satapathy, S.K. Rath. “Empirical Validation of Neural Network Models for Agile Software Effort Estimation based on Story Points”. Procedia 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015) Vol 57 pp. 772–781.

[14] R.S. Pressman. “Software Enginnering A Practitioner Aprroach Seventh Edition”. New York: Mc.Graw-Hill. 2010. pp 13, 14, 65-72.

[15] A. Qumer, B.H. Sellers, “An Evaluation of Degree of Agility in Six Agile Methods and its Applicability for Method Engineering”. Elsevier The Journal of Systems and Software 50 (2008) ,pp 280-295.

[16] A. Qumer, B.H. Sellers. “A Framework To Support The Evaluation, Adoption And Improvement Of Agile Methods In Practice”. Elsevier The Journal of Systems and Software 81, 2008, pp 1899–1919.

[17] I. Richardson, C.G.v. Wangenheim. “Why Are Small Software Organizations Different?”. IEEE Software, Vol.24,No.1, 2007, pp.18-22.

[18] F. Sarro, A, Petrozzielloy, M. Harman. “Multi-Objective Software Effort Estimation”. 2016 IEEE/ACM 38th IEEE International Conference on Software Engineering, pp 619-630

[19] L. Song, L.L. Minku. “The Impact Of Parameter Tuning On Software Effort Estimation Using Learning Machines”. ACM PROMISE '13 Proceedings of the 9th International Conference on Predictive Models in Software Engineering Article No. 9, 2013.

[20] P.k. Suri, P. Ranjan. “Comparative Analysis Of Software Effort Estimation Techniques”. International Journal of Computer Applications, 2012, vol 48 no 21 pp 12-19.

[21] A. Trendowicz. “Software Cost Estimation, Benchmarking, And Risk Assessment”. Berlin : Springer-Verlag. 2013, Pp 3.

[22] Ziauddin, S.K. Tipu, S. Zia. “An Effort Estimation Model for Agile Software Development”. Advances in Computer Science and its Applications,(ACSA) 314 Vol. 2, No. 1, 2012, ISSN 2166-2.

[23] S. Haykin. “Neural Networks and Learning Machines Third Edition”. Ontario: Pearson-Prentice Hall. 2009. pp 123-153, 217.

[24] F. Akhtar C. Hahne. “RapidMiner 5 Operator Reference”. Dortmund : Rapid I GmbH. 2012. pp 697-701.