

Towards a Machine Learning Model for Predicting Failure of Agile Software Projects

Ahmed Abdelaziz
Mohamed

Department of Information
Systems and Technology,
Institute of Statistical Studies
and Research,
Cairo University, Egypt

Nagy Ramadan Darwish
Department of Information
Systems and Technology,
Institute of Statistical Studies
and Research,
Cairo University, Egypt

Hesham Ahmed Hefny
Department of Computer and
Information Sciences, Institute
of Statistical Studies and
Research,
Cairo University, Egypt

ABSTRACT

Agile software development plays a very significant role in software projects. Agile software project is a refined approach to design and direct project processes. An agile project is finished in short sections called iterations. This paper introduces a survey of machine learning approaches for predicting failure of agile software projects. It reviews the uses of machine learning techniques such as fuzzy logic, multiple linear regression, neural network, logistic regression and etc., for predicting success and failure of agile software projects. This paper also proposes machine learning model for predicting failure of agile software projects. Many researches in this topic were reviewed, analyzed, summarized, and compared according to the used machine learning techniques in agile software projects.

General Terms

Agile Software Projects, Machine Learning

Keywords

Agile Software Projects, Machine Learning, Fuzzy Logic, Multiple Linear Regression.

1. INTRODUCTION

According to the Agile Manifesto (2011), Agile is a framework that concentrates on client value, refined delivery, small teams, self-enterprise and continuous enhancements. Many organizations use agile methodology in software projects to solve traditional methodology problems that lead to lost time, cost and effort [1]. As shown below in figure 1[2]:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan
That is, while there is value in the items on the right, we value the items on the left more.

Fig 1: Manifesto for Agile Software Development

Agile is a set of software development operations that are refined, incremental, self-enterprise [3]. Also, agile is a concept of flexibility, agility, action, cleverness in motion. Agile methodologies are effective, low-risk, predictable, and efficient way to develop software. In agile, the time and requirements

are obtainable which is constant. In traditional ways, the time and requirements are elastic while the functionality is constant in comparison with the agile methodology [4].

The features of agile software operation include: modularity on expansion operation standard, refined with small cycles, adaptive, incremental, reduce the risks, stakeholders oriented, concurrent and communicative [3-6]. The advantages of agile software include: income, performance, vision, risk management, lightness, client satisfaction and assist to produce the fully product [7].

The agile lifecycle based on experiment operation, which cause the good performance. The visibility indicates to support the product development. This provides efficient vision for key users. Risk management, a short incremental releases are visual to the product holder and product team through its development which help us to select any problems early and make it easier to reply to alteration. Agility indicates to admit the alterations, client satisfaction, creating the suitable product [6, 8]. There are agile methodologies: feature driven development, Extreme programming, rational unified process, animated systems development, scrum and Adaptive software development [9].

Many reasons cause the failure of agile software projects such as weak project planning, scanty communication between teams and customers, lack of alteration, monetary management, and quality management, ineffectual participation of implement management, lack of qualified team members, ability to adapt, and using wrong methodology and tools [10].

According to CHAOS report, there are three types of software projects as follow:

- Large size software projects
- Medium size software projects
- Small size software projects

Using agile methodology in large size software projects led to success in these projects approximately to 18% compared to success projects in traditional methodology such as waterfall model that approximately to 3%. Fail of software projects in agile methodology approximately to 23% while in waterfall model approximately to 42% [11]. In medium size software projects, success of software projects approximately to 27% while waterfall model approximately to 7%, fail of software projects approximately to 11% while waterfall model approximately to 25%. In small size software projects, success of software projects approximately to 58% while waterfall model approximately to 44%, fail of software projects

approximately to 4% while waterfall model approximately to 11%. As shown below in table 1.

Table1. CHAOS Report by Agile Versus Waterfall

Size	Method	Successful	Challenged	Failed
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	20%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

There is a lack of study on critical failure factors in the Agile software development project, based on recent searches in peer reviewed academic literature or practitioner literature related to this topic. There are historical data and research theories on successes or failures/problems in agile implementation and agile software development projects.

Failure prediction for agile software projects is one of the most challenging activities in software projects. Most researches use statistical techniques to predict failure of agile software projects that lead to poor results. Many organizations need intelligent techniques for failure prediction of agile software projects to obtain accurate results. Accurate prediction helps organizations to save cost, time and effort in projects.

This paper proposed a new model to predict failure of agile software projects. The proposed model is based on intelligent techniques such as multiple linear regression and fuzzy logic. Multiple linear regressions are used to determine critical failure factors. Fuzzy logic is used to predict failure of agile software projects.

This paper is arranged as follows: section 2 introduces problem definition; section 3 presents related work, section 4 presents proposed model, section 5 challenges and future work, and finally, section 6 presents conclusion.

2. PROBLEM DEFINITION

CHAOS report showed that 23% of large size of software projects failed, 11% of medium size of software projects failed and finally, 4% of small size of software projects failed (refer to table 1). Although many companies use agile methodology in software projects to solve traditional methodology problems such as waterfall model, many software projects are still failed. Companies that use agile methodology need intelligent techniques to predict failure of agile software projects.

Failure prediction for the agile software projects are based on accurate estimation in planning phase. The nature of agile methodologies are different compared with the traditional methods, so the traditional techniques can give us inaccurate estimation in failure prediction for the agile software projects that lead to lose in time and cost. Failure prediction for the agile software projects needs intelligent techniques to get better results in failure prediction. It will be useful to overcome these problems by introducing a proposed model based on multiple linear regression and fuzzy logic.

3. RELATED WORK

Through related work, many researches were reviewed and analyzed, these researches use traditional techniques such as (Discriminate analysis, Canonical Correlation and etc.) and machine learning techniques such as (neural network, fuzzy logic and etc.) to estimate failure and success of agile software projects. This paper can classify the reviewed researches into two categories. Category one contains researches which presented traditional statistical techniques of agile software projects. Category two contains researches which presented machine learning techniques of agile software projects. This paper interpreted two categories of reviewed researches as shown below:

3.1 Statistical Techniques of Agile Software Projects

Category one presented important researches that use statistical techniques of agile software projects. **V.Lalsing and et al**, presented new method to evaluate people factors in agile software projects based on time delivery, rework level, defect rate, accountability, communication channels. This paper implemented experiments on small and medium companies and using scrum method [1]. Researcher opinion, people factors may effect on agile teams in software projects.

H. Taherdoost and et al, presented a survey to show failure and success factors of information technology software projects through organizational, people, process and technical factors. This paper focuses on organizational factors that use in companies interested to agile methodologies [12]. Researcher opinion, this study can determine failure and success factors that effected on information technology software projects.

M. Shepperd and et al, presented a new framework to predict success of agile software projects based on unbiased statistic and Standardized Accuracy. This paper focuses on different factors such as organizational, people, process and technical factors. This study tries to find the optimal prediction of agile software projects [13]. Researcher opinion, proposed framework can't detect the optimal prediction of agile software projects.

S. Lee and et al, presented a new framework to evaluate agile methodology in small projects based on agility factors are composed of flexibility, speed, leanness, learning and responsiveness. This paper implemented experiments on small projects and using scrum method [14]. Researcher opinion, agility factors may find the best evaluate small projects in agile methodology.

A. Elzamy and et al, presented a new framework to estimate risk management to evaluate success software projects based on risk factors such as (identification risk, risk analysis and evaluation, risk treatment, risk controlling, risk communication and documentation) and mining techniques. This paper uses questionnaire and historical data from different companies to evaluate software projects [10]. Researcher opinion, risk management may effect on agile software projects.

M. Tanner and et al, introduced a new method to find critical success factors to improve successful agile adoption based on customer involvement, stakeholder involvement, team structure, project type, and skill level of team members. This paper tries to collect historical data through questionnaires and meeting to experts to find optimal critical success factors [15]. Researcher opinion, critical success factors may effect of agile software projects.

Feras A. Batarseh and et al, introduced a new method to predict failure of agile software projects based on Mean Time between Failures and modeling regression. The proposed method uses modeling regression to determine types of software systems failures in different enterprises interested to agile methodology [16]. Researcher opinion, the proposed method can't find the best evaluate failure of agile software projects.

D. S. Nguyen introduced a new method to determine success factors that effect of agile software projects based on collect historical data from questionnaire and brain storming. this paper can determine critical success factors that effect of agile software projects such as customer involvement, good planning and continuous delivery products [17]. Researcher opinion, critical success factors may effect of agile software projects.

3.2 Intelligent Techniques of Agile Software Projects

Category two presented important researches that using machine learning techniques of agile software projects. **T. Chow and et al**, presented a new framework to determine critical success factors that effect of agile software projects based on multiple linear regression. This paper shows three critical success factors are Delivery Strategy, Agile Software Engineering Techniques and Team Capability in order to Quality, Scope, Time, and Cost [18]. Researcher opinion, multiple linear regressions may find critical success factors that effect of agile software projects.

N. Cerpa and et al, presented a new model to evaluate success factors to predict successful of agile software projects based on logistic regression. This paper focuses on different success factors such as customer involvement, good scheduling, project manager, and development process and development team. This paper shows that customer involvement one of the most important factors that effect of agile software projects [19]. Researcher opinion, logistic regression can predict successful of agile software projects.

R. P. Mohanty and et al, presented a new approach to estimate critical risk factors that affect information systems software projects based on logistic regression and neural network. This paper uses logistic regression and neural network to evaluate performance of information systems projects and determine critical risk factors [20]. Researcher opinion, the proposed approach may find critical risk factors and evaluate performance of information systems projects.

D. Stankovic and et al, introduced a new method to determine critical success factors in information technology projects based on multiple linear regression. This paper shows critical success factors such as team environment, Project management process and agile software engineering [21]. Researcher opinion, multiple linear regressions can detect critical success factors that effect of information technology projects in companies interested of agile methodology.

Pushpavathi T.P and et al, introduced a new model for predicting successful of agile software projects based on genetic algorithm based fuzzy c-means clustering and random forest classifier. This paper focuses on different success factors such as Total project time, Experience level of inspectors, testing time scheduled and Defect count estimation. This paper shows that the accuracy result of the model is 93.05% [22]. Researcher opinion, the proposed model may find the optimal prediction for successful of agile software projects.

H. B. Yadav and et al, introduced a fuzzy logic model for predicting successful of agile software projects based on software metrics such as requirements phase, design phase, coding phase and testing phase. This paper implemented experiments on twenty projects in different sizes [23]. Researcher opinion, fuzzy logic model may detect the best prediction of agile software projects.

T. Hovorushchenko and et al, introduced a neural network model for predicting successful of agile software projects based on software project cost, software project duration, software project complexity, software project quality and software project usability. This paper shows that software project duration and software project quality that effect of successful of agile software projects [24]. Researcher opinion, neural network model may find the optimal prediction of successful of agile software projects.

S.A. Rizvi and et al, introduced a fuzzy logic model for predicting early stage of successful of agile software projects based on requirement stage and design stage. This paper shows that the accuracy result of the model is 98.4% [25]. Researcher opinion, the proposed model may find the best prediction for successful of agile software projects.

V. Vashisht and et al, presented a neuro- fuzzy model for predicting successful of agile software projects based on requirement stage, analysis stage, design stage and test phase. This paper shows that the accuracy result of the model is 93.4% [26]. Researcher opinion, the neuro-fuzzy model can find the optimal prediction for successful of agile software projects.

Table 2 introduced contributions of statistical techniques of agile software projects that were covered in 8 papers.

Table2. Contribution of Statistical Techniques of Agile Software Projects

No	Authors	Contribution
1	V.Lalsing and et al	Using quantitative approach to evaluate people factors that effect of agile software projects.
2	H. Taherdoost and et al	Using standard deviation to evaluate failure and success factors that effect of information technology software projects.

Table2. Contribution of Statistical Techniques of Agile Software Projects Con't

No	Authors	Contribution
3	M. Shepperd and et al	Using unbiased statistic to predict success of agile software projects.
4	S. Lee and et al	The proposed agility equations to evaluate small projects in agile methodology.
5	A. Elzamy and et al	Using risk management equations to estimate agile software projects.
6	M. Tanner and et al	Using mean analysis and standard deviation to determine critical success factors in agile software projects.
7	Feras A. Batarseh and et al	Using mean time and modeling regression to evaluate failure of agile software projects.
8	D. S. Nguyen	Using time series to determine critical success factors in agile software projects.

Table 3 introduced contributions of machine learning techniques of agile software projects that were covered in 9 papers.

Table3. Contribution of Intelligent Techniques of Agile Software Projects

No	Authors	Contribution
1	T. Chow and et al	Using multiple linear regressions to determine critical success factors in agile software projects.
2	N. Cerpa and et al	Using logistic regression to predict successful of agile software projects.
3	R. P. Mohanty and et al	Using logistic regression and neural network to evaluate performance of information technology software projects.
4	D. Stankovica and et al	Using multiple linear regressions to determine critical success factors of information technology software projects.
5	Pushpavathi T.P and et al	Using genetic algorithm based fuzzy c-means clustering and random forest classifier to predict successful of agile software projects.
6	H. B. Yadav and et al	Using fuzzy logic to predict successful of agile software projects.
7	T. Hovorushchenko and et al	Using neural network to predict successful of agile software projects.
8	S.A. Rizvi and et al	fuzzy logic model for predicting early stage of successful of agile software projects
9	V. Vashisht and et al	Using neuro-fuzzy to predict successful of agile software projects.

Table 4 introduces summary of two categories (statistical techniques of agile software projects and machine learning techniques of agile software projects) as shown below:

Table4. Related Work Summary

No	Category	Some Approach
1	Statistical Techniques of Agile Software Projects	<ul style="list-style-type: none"> ➤ quantitative approach ➤ standard deviation ➤ unbiased statistic ➤ mean analysis ➤ mean time and modeling regression ➤ time series
2	Machine Learning Techniques of Agile Software Projects	<ul style="list-style-type: none"> ➤ multiple linear regressions ➤ logistic regression ➤ neural network ➤ genetic algorithm ➤ fuzzy c-means clustering ➤ random forest classifier ➤ fuzzy logic ➤ neuro-fuzzy

Through related work, neural network and fuzzy logic uses more than once in agile software projects. So, this paper introduces comparative study between neural network and fuzzy logic, show below in table 5.

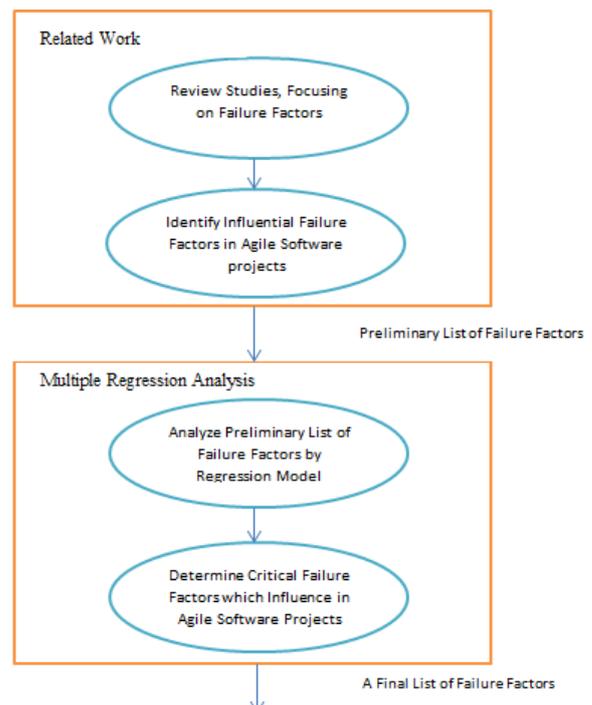
Table5. Comparison between Neural Network and Fuzzy Logic

Criteria	Neural Network	Fuzzy Logic
Knowledge unit	Cases	Rules and membership
Complexity	Complex to understand	easy to understand
Knowledge acquisition unit	Precedent cases, hierarchies	Membership, hierarchies
Accuracy	Not Provide with accurate responses to ambiguous data	Provide accurate responses to ambiguous data
Learning	Learning from scratch.	Apriority of knowledge is essential
Knowledge containers	Learn from past cases	Set of rules(rule based expert system)
Reliability	Based on past examples	Based on expert or knowledge source
Adaptability to new knowledge	Automatically adapt to training with new examples	Need to add new knowledge explicitly
Maintenance of knowledge	Easier because of adaptability	Modifying knowledge base is a difficult task

4. PROPOSED MODEL

This section introduced the proposed model for predicting failure of agile software projects. The proposed model is composed of three parts: as shown below in figure 2.

1. Review recent papers to extract important failure factors related to agile software projects.
2. Determine critical failure factors in agile software projects by using multiple linear regressions.
3. Predict failure of agile software projects by using fuzzy logic.



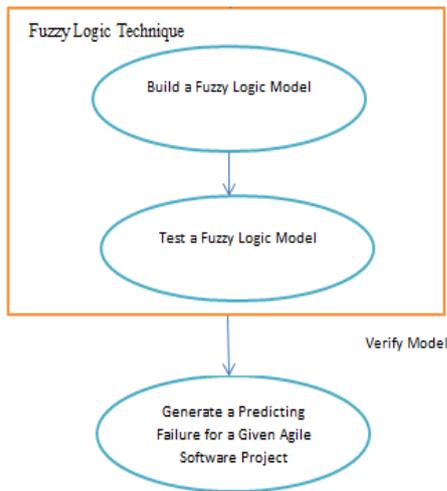


Fig 2: The Proposed Model for Predicting Failure of Agile Software Projects

Each component of the proposed model will be described briefly in the following sub sections:

4.1 Review Studies

The first step is to review recent studies for predicting failure of agile software projects. This step aims to extract important failure factors in agile software projects. Extract important failure factors are based on some evaluation criteria (Ambiguity, Usability and Agility) as shown below in figure 3.

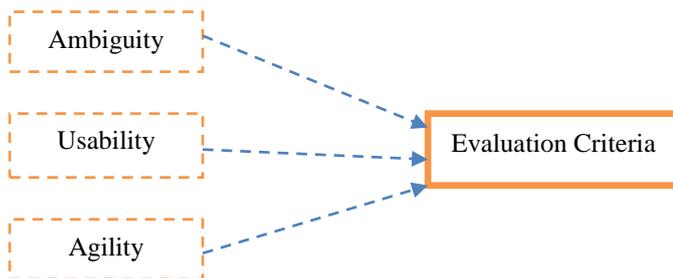


Fig 3: Evaluation Criteria to Extract Important Failure Factors

- **Ambiguity of failure factors:** clarity of failure factors to users in agile software projects that can't be understood in two or more possible ways.
- **Usability of failure factors:** enhancing the quality of failure factors in agile software projects and Stay away from complexity in using failure factors.
- **Agility of failure factors:** gauge the efficiently of failure factors in agile software projects in order to adaptability and flexibility.

4.2 Multiple Regression Analysis

Multiple linear regressions are used to determine critical failure factors in agile software projects. Simple regression analysis is composed of one dependent variable and one independent variable. Multiple regression analysis is composed of one dependent variable and more than independent variable. It is formulated as follows [27]:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \varepsilon \quad (1)$$

Where:

- y represents the dependent variable
- x_1, x_2, \dots, x_n are the independent variables
- β_i is the regression coefficient
- ε is the random error component.
- β_0 is the y intercept

Y represents the dependent variable (degree of effect in failure factors of agile software projects) and x_1, x_2, \dots, x_n are the independent variables (failure factors in agile software projects).

Multiple linear regression introduced three main tables are regression statistics, ANOVA analysis and coefficients analysis as shown below in figure 4.

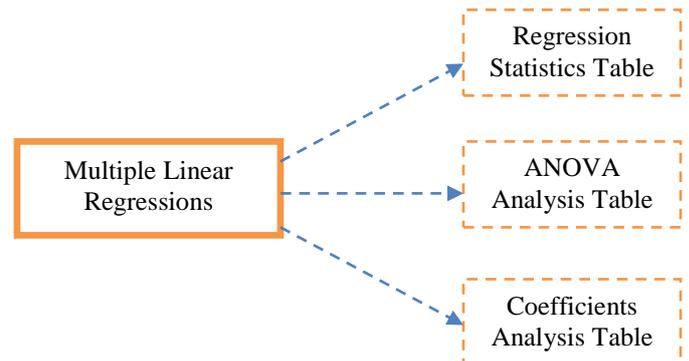


Fig 4: Three Main Tables in Multiple Linear Regressions

- **Regression Statistics** are consists of multiple correlation coefficient (R), determination coefficient, adjusted R square, and finally standard error. Adjusted R square represents the percentage of variance in the dependent variable (degree of effect in failure factors of agile software projects) can be interpreted by the independent variables (failure factors in agile software projects).
- **ANOVA Analysis** consists of regression, residual and significance F . whenever, significance F is lower than 0.05 , this means that the regression model helps to determine the most significant attributes that impact failure factors in agile software projects.
- **Coefficients Analysis** consists of coefficients, standard error, t stat and p -value. If p -value in failure factors < 0.05 , then failure factors are included in the critical list is used.

4.3 Fuzzy Logic Technique

Fuzzy logic technique is used to predict failure of agile software projects. Fuzzy logic is an approach to computing based on "degrees of truth" rather than the usual "true or false" (1 or 0) Boolean logic on which the modern computer is based. The idea of fuzzy logic was first advanced by Dr. Lotfi Zadeh of the University of California at Berkeley in the 1960s. Some fuzzy logic concepts as follows:

- **Fuzzy Sets** are groups whose elements have degrees of membership. Fuzzy sets can be defined mathematically:

$$A = \{(x, \mu_A(x)) / x \in X\} \quad (2)$$

$$\mu_A(x) \in [0, 1]$$

Where, The (X) is called the universe discourse (representing all the possible values of variable (x)). the possible values of variable x between 0 and 1.

- **Linguistic Variables** are the variables that carry values in the form of Words or sentences of language such as "low", "medium", "high".
- **Membership Function** used to determine how the affiliation of any element of elements to fuzzy sets. The basic condition of this function is to have a range between Zero and one. The most common forms are: Triangular, Trapezoidal and Gaussian as shown below in figure 5.

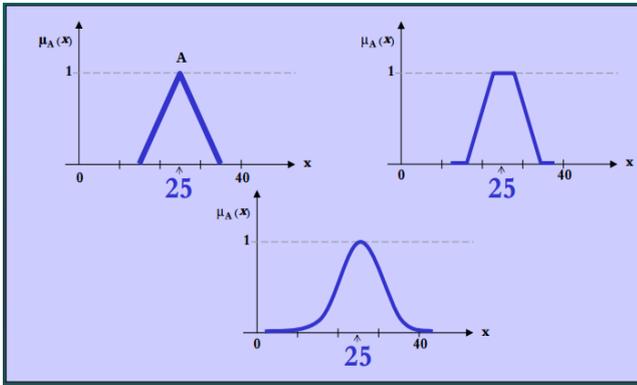


Fig 5: The Most Common Forms in Membership Function

- **Basic operations in fuzzy sets as follows:**
- **Intersection** of two sets A and B ($C = A \cap B$) can be defined in membership function as follows:

$$\mu_{A \cap B}(x) = \mu(A \text{ AND } B) = \min \{ \mu_A(x), \mu_B(x) \} \quad (3)$$

- **Union** of two sets A and B ($C = A \cup B$) can be defined in membership function as follows:

$$\mu_{A \cup B}(x) = \mu(A \text{ OR } B) = \max \{ \mu_A(x), \mu_B(x) \} \quad (4)$$

- **Complement** is the part that stays out of fuzzy sets N although it belongs to the universal group X can be defined in membership function as follows:

$$\mu_{\bar{N}}(x) = \mu(\text{NOT } N) = 1 - \mu_N(x) \quad (5)$$

This paper proposed fuzzy logic model to predict failure of agile software projects through inference engine that contains three main steps are Fuzzification, knowledge base and Defuzzification as shown below in figure 6.

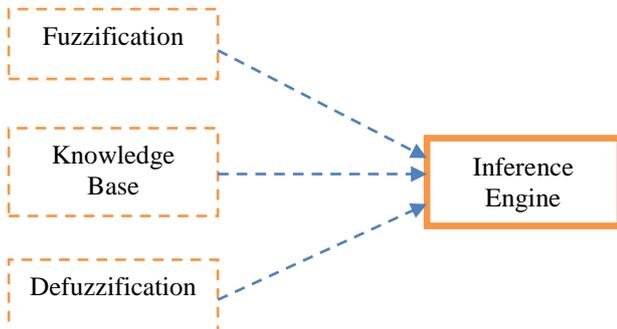


Fig 6: Three Main Steps in Inference Engine

Fuzzification is the operation of changing an actual number value (critical failure factors in agile software projects) into a fuzzy value. Fuzzification uses linguistic variables such as (low - normal and high) and illustrates it in membership function. Fuzzification process converts critical failure factors to

linguistic variables in membership function as shown below in figure 7.

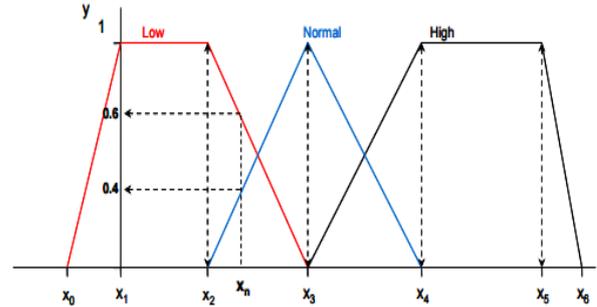


Fig 7: Sample of Fuzzification of Failure Factors in Agile Software Projects

Knowledge base converts inputs (critical failure factors in agile software projects) and output (failure decision in agile software projects) into if-then rules, sample of knowledge base in critical failure factors of agile software projects as follows:

If factor 1 is low and factor 2 is medium **then** failure is low

Defuzzification process, the linguistic values are converted to numerical values to facilitate on computers and machines to deal with it. Defuzzification process can be defined mathematically as follows [28]:

$$y_o = (y_1 \mu_Y(y_1) + y_2 \mu_Y(y_2)) \div \mu_Y(y_1) + \mu_Y(y_2) \quad (6)$$

Where:

- Y: The fuzzy sets to which the decision belongs.
- y₁: The first decision
- y₂: The second decision
- μ: Degree of membership
- y_o: The final decision

Defuzzification process converts Fuzzification process (linguistic variable of failure factors in agile software projects) into numerical value to facilitate on stakeholders that interested in agile software to deal with it.

5. CHALLENGES AND FUTURE WORK

Future work aims to improve the accuracy of failure prediction of agile software projects in early phase in software project phases by applying the multiple linear regressions to determine the critical factors of failure and fuzzy logic to predict failure of agile software projects. Therefore, this paper will propose a model based on hybrid machine learning to predict failure of agile software projects as shown below in figure 8.

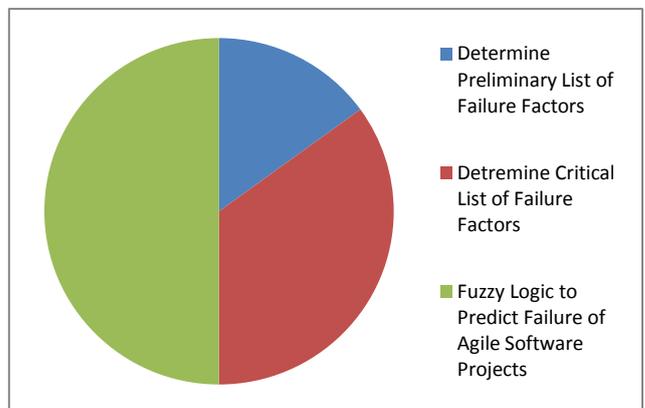


Fig 8: Future Challenges in Agile Software Projects

Future challenges work aims to achieve three main objectives as shown below:

- Determine preliminary list of failure factors in agile software projects.
- Determine critical list of failure factors in agile software projects by using multiple linear regression.
- Using fuzzy logic to predict failure of agile software projects.

6. CONCLUSION

This paper introduced seventeen studies in agile software projects. The reviewed studies are classified into two categories, the first category is statistical techniques in agile software projects and the second category is intelligent techniques in agile software projects. This paper presented contributions of the reviewed studies in agile software projects and related work summary. This paper introduced the proposed model for predicting failure of agile software projects. This paper also introduced the future challenges work, using multiple linear regressions to determine critical failure factors in agile software projects and using fuzzy logic to predict failure of agile software projects.

7. REFERENCES

- [1] V. Lalsing, S. Kishnah and S. Pudaruth “People Factors in Agile Software Development and Project Management”, *IJSEA*, Vol.3, No.1, 2012, pp. 117-137.
- [2] J. Li “Agile Software Development”, Technics University Berlin, Berlin, Germany, 2012.
- [3] John Hunt, “Agile software construction”, Springer, 2006.
- [4] Ann L. Fruhling and Alvin E. Tarrell, “Best Practices for Implementing Agile Methods”, IBM, 2008.
- [5] Laurie Williams, Robert R. Kessler, Ward Cunningham, and Ron Jeffries, “Strengthening the case for pair Programming”, *IEEE Software*, 2000.
- [6] Mike Cohn, “Software Development using Scrum”, Addison-Wesley, 2010.
- [7] K. N. Rao, G. K. Naidu and P. Chakka “A Study of the Agile Software Development Methods, Applicability and Implications in Industry”, *IJSEA*, Vol.5, No.2, 2011, pp. 35-46.
- [8] Ioannis G. Stamelos and P. Sfetos “Agile Software Development Quality Assurance”, information science references, Idea Group Inc, 2007.
- [9] V. E. Jyothi and K. N. Rao “Effective Implementation of Agile Practices”, *IJACSA*, Vol.2, No.3, 2011, pp. 41-48.
- [10] ELZAMLY and B. HUSSIN “An Enhancement of Framework Software Risk Management Methodology for Successful Software Development”, *JATIT*, Vol.62, No.2, 2014, pp. 410-423.
- [11] K. Jammalamadaka and V. R. Krishna “Agile Software Development and Challenges”, *IJRET*, Vol.2, No.8, 2013, pp. 125-129.
- [12] TAHERDOOST and A. KESHAVARZSALEH “A Theoretical Review on IT Project Success/Failure Factors and Evaluating the Associated Risks”, *Mathematical and Computational Methods in Electrical Engineering*, vol.1, 2012, pp. 80-88.
- [13] M. Shepperd and S. MacDonell “Evaluating prediction systems in software project estimation”, *Information and Software Technology*, ELSEVIER, vol.54, 2012, pp. 820-827.
- [14] S. Lee and H. Yong “Agile Software Development Framework in a Small Project Environment”, *JIPS*, Vol.9, No.1, 2013, pp. 69-88.
- [15] Tanner and U. Willingh “Factors Leading to the Success and Failure of Agile Projects Implemented in Traditionally Waterfall Environments”, *MKL*, 2014, pp. 693-701.
- [16] Feras A. Batarseh and Avelino J. Gonzalez “Predicting failures in agile software development through data analytics”, Springer, 2015.
- [17] D. S. Nguyen “Success Factors That Influence Agile Software Development Project Success”, *ASRJETS*, vol.17, No 1, 2016, pp. 172-222.
- [18] T. Chow, D. Cao “A survey study of critical success factors in agile software projects”, *JSS*, Elsevier, vol.81, 2008, pp. 961-971.
- [19] N. Cerpa, M. Bardeen, B. Kitchenham and J. Verner “Evaluating logistic regression models to estimate software project outcomes”, *IST*, Elsevier, vol.52, 2010, pp. 934-944.
- [20] R. P. Mohanty, G. Sahoo and J. Dasgupta “Identification of Risk Factors in Globally Outsourced Software Projects using Logistic Regression and ANN”, *Int. J. Sup. Chain. Mgt*, vol.1, 2012, pp. 2-11.
- [21] D. Stankovic, V. Nikolicb, M. Djordjevic, D. Caod “A survey study of critical success factors in agile software projects in former Yugoslavia IT companies”, *JSS*, Elsevier, vol.86, 2013, pp. 1663– 1678.
- [22] Pushpavathi T.P, Suma V, and Ramaswamy V “Defect Prediction in Software Projects-Using Genetic Algorithm based Fuzzy C-Means Clustering and Random Forest Classifier”, *IJSER*, Vol.5, 2014, pp. 888-898.
- [23] H. Yadav, D. Yadav “A fuzzy logic based approach for phase-wise software defects Prediction using software metrics”, *INFOSOF*, 2015, vol.1, pp. 1-19.
- [24] T. Hovorushchenko and A. Krasiy “Realization of the Neural Network Model of Prediction of the Software Project Characteristics for Evaluating the Success of its Implementation”, *ICIDAACS*, IEEE, 2015, pp. 348-353.
- [25] S. A. Rizvi, R. A. Khan and V. K. Singh “Software Reliability Prediction using Fuzzy Inference System: Early Stage Perspective”, *IJCA*, vol.145, No 10, 2016, pp. 16-23.
- [26] V. Vashisht, M. Lal and G. S. Sureshchandar “Defect Prediction Framework Using Adaptive Neuro-Fuzzy Inference System (ANFIS) for Software Enhancement Projects”, *British Journal of Mathematics & Computer Science*, vol.19, No 2, 2016, pp. 1-12.
- [27] N. R. Darwish, A. A. Mohamed and A. S. Abdelghany “A Hybrid Machine Learning Model for Selecting Suitable Requirements Elicitation Techniques”, *IJCSIS*, vol.14, No 6, 2016, pp. 1-12.
- [28] A. A. Mohamed and A. S. Salama “A Fuzzy Logic based Model for Predicting Commercial Banks Financial Failure”, *IJCA*, vol.79, No 11, 2013, pp. 16-21.