



Intelligent System for Forecasting Failure of Agile Projects

Ahmed Abdelaziz and Alia N Mahmoud

Nova Information Management School, Universidade Nova de Lisboa, 1070-312, Lisboa, Portugal
D20190535@novaims.unl.pt, M20190508@novaims.unl.pt

Abstract:

Revealing the failure of agile software projects is a great challenge faced by software companies. This paper focuses on the using of intelligent techniques such as fuzzy logic, multiple linear regressions, support vector machine, neural network to address this challenge. This paper also presents a review of some works related to this area of interest. In this paper, the researchers propose an approach for revealing the failure of agile software projects based on two intelligent techniques: fuzzy logic and multiple linear regressions (MLR). MLR is used to determine crucial failure factors of agile software projects. Fuzzy logic is used for revealing failure of agile software projects.

Keywords: Agile Projects, Intelligent Techniques, Fuzzy Logic, Multiple Linear Regressions.

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, 2]. Agile is a set of software development operations that are refined, incremental, self-enterprise [3]. The term Agile was appeared in February 2001 where the Agile Manifesto is formulated. Agile development is a framework for managing the development of software projects, which are based on obtaining early customer feedback with many frequent releases. In Agile development, the work is executed in small time-boxed iterations. Each iteration is viewed as full software development cycle (i.e. planning, requirements analysis, design, coding, unit testing, and acceptance testing). This helps in reducing the overall risk and allows the project to quickly adapt to changes [4-6].

The most used agile methodologies include feature driven development (FDD), Extreme programming (EP), rational unified process (RUP), Adaptive software development (ASD) and scrum [7-10]. In agile projects, the team is usually cross-functional and self-organizing. Most of the agile methodologies have a daily face-to-face communication between team members. The agile lifecycle focuses on test process, customer satisfaction and timely delivery. The agile software development lifecycle is controlled by the incremental process.

Software companies are facing a big challenge for early revealing failure of agile software projects. Most of the studies used traditional methods to reveal failure of agile software projects that lead to poor results. There is a need for new methods to reveal failure of agile software projects to get more accurate results. This will also help software companies to save cost and time. Furthermore, the factors that affect the success or failure of software projects need to be identified.

The failure of agile software project has many reasons such as weak experience with agile method, insufficient connection between teams and clients, lack of modification, fiscal management, fineness management, inactive sharing of execute management, lack of competent team members and using weak tools [11].

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

- Big size software projects
- Moderate size software projects
- Little size software projects

CHAOS report concentrates on three main axes in software projects, as follows:

- Software projects failure
- The major features that cause software projects to fail
- The key ingredients that can minimize project failures

The Standish Group's CHAOS report shows a comparison between agile and traditional waterfall projects. According to this report, agile approaches have more successful projects and less outright failures for every project size. Table 1 shows the results of the most recent report [11].

Table1. Standish Group's CHAOS Report

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%

This paper proposes an approach to reveal failure of agile software projects. For this purpose, this paper tries to utilize the advantages of intelligent techniques to address this problem. Linear regression analysis is used to identify crucial failure factors. Fuzzy logic is used for revealing the failure of agile software projects.

The rest of the paper is organized as follows: section two presents a background overview; section three introduces the previous work in this research filed; section four introduces the proposed approach and its components; section five introduces the challenges and future work; and finally, section six gives the conclusion.

2. Background and Overview

This section presents an overview of agile methodologies such as Scrum, Extreme programming (XP), Adaptive Software Development (ASD) and Feature-Driven Development (FDD). It also presents an overview of intelligent techniques such as fuzzy logic and multiple linear regressions.

2.1 Agile Methodologies

XP was proposed by Kent Beck in 2000. XP has a set of values and practices. XP includes five basic values: communication, simplicity, feedback, courage, and respect. XP supports the following practices: small iterations, code standards, spike solutions, user stories, unit testing, pair programming, and Continuous Integration [12]. Scrum is an

agile method to manage a software project. Scrum focuses on the members of staff should function in order to create the system flexibly in constantly changing environment [13]. It is composed of five phases as follows:

Product backlog creation
 Sprint Planning and Sprint Backlog Creation
 Scrum Meetings
 Testing and Product Demonstration
 Retrospective and Next Sprint Planning

FDD is a reiterated software development process. Its main objective is to provide tangible, working software iteratively in a timely manner [12]. It is composed of five activities as follows:

Develop overall model
 Build feature list
 Plan by feature
 Design by feature
 Build by feature

ASD is an agile method for managing a large software project. It concentrates on the problems of improving complex and huge systems. This methodology highly supports incremental, iterative development, with fixed prototyping. ASD is composed of six basic characteristics: mission focused, feature based, iterative, time-boxed, risk driven and change tolerant [13].

2.2 Intelligent Techniques:

This section introduces overview of linear regression and fuzzy logic as follows:

2.2.1 Linear Regression Analysis

Linear regression (LR) analysis is composed of two types which are simple linear regression and multiple linear regressions. The general linear regression formula can be formulated as follows [14]:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_N x_N + \epsilon \quad (1)$$

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

In linear regression, there are three main outputs that include regression details, ANOVA table and coefficients table as shown below in figure 1.

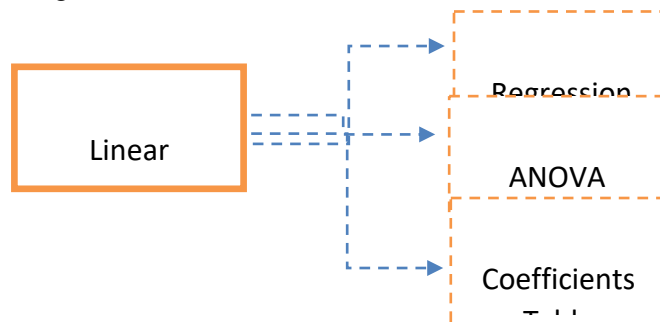


Figure.1 Three Main Results in Linear Regression

- Regression Detail shows the precise of the LR model. It is composed of three indicators which are correlation coefficient, determination coefficient, adjusted R square, and finally standard error.

- ANOVA table contains one of the most significant features which are 'significance F'. Whenever, significance F is < 0.05 , this means that the proposed approach supports to define the most important factors that influence failure of agile software projects.
- Coefficients table contains one of the most significant features which are 'p-value'. If p-value in failure factors is less than 0.05, then failure factors are included in the crucial list.

2.2.2. Fuzzy Logic

Fuzzy logic is used for revealing failure of agile software projects. It is based on probabilities between 0 and 1. Fuzzy logic also contains many concepts such as linguistic variables, membership function, knowledge rules, Fuzzification and Defuzzification. These concepts will be described as follows:

- Linguistic Variables are the variables that hold items in a form of statements or sentences such as "hot", "cold" and "very cold".
- Membership functions are used to map the non-fuzzy input values to fuzzy linguistic terms and vice versa. As shown in figure 2, a membership function can have many forms such as triangular, trapezoidal, Gaussian or generalized bell.

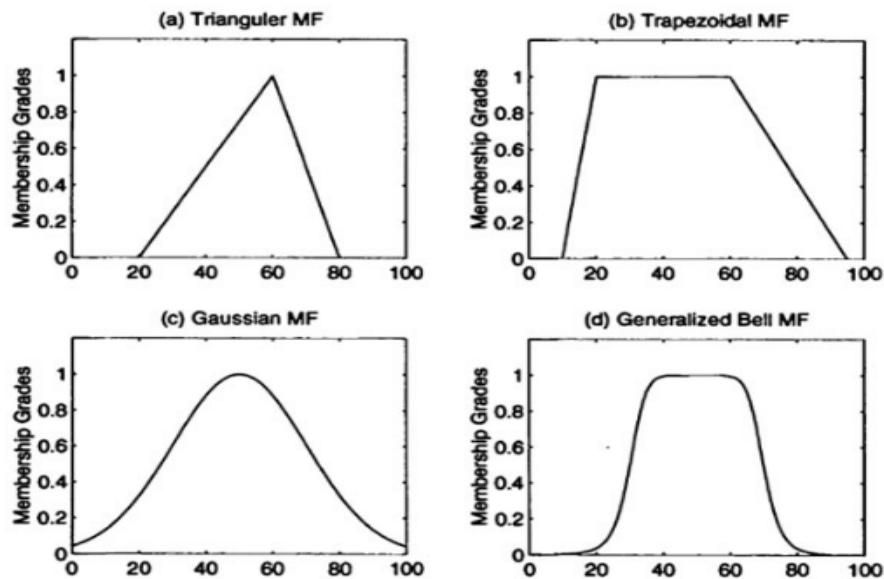


Figure.2 Sample of Different Membership Functions

Knowledge base converts input values and output values into if-then rules for example:

If cost is low and time is high then price is Low

Fuzzification is the process of replacing a current digit value into a fuzzy value, as shown below in figure 3.

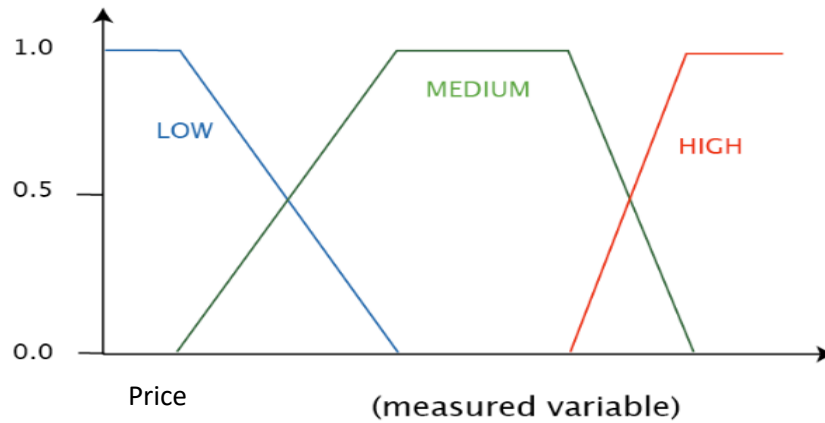


Figure.3 Example of Fuzzification process

Defuzzification is the process in which the linguistic values are transformed into numerical values that computers can deal with it. Defuzzification (center of gravity) can be formulated as follows [15,16]:

$$Y_o = (y_1 \mu_Y(y_1) + y_2 \mu_Y(y_2)) \div \mu_Y(y_1) + \mu_Y(y_2) \quad (2)$$

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

3. Related Work

Through previous work, many studies were reviewed and construed, these studies utilize classical techniques such as (statistical analysis and etc.) and intelligent techniques such as (support vector machine, fuzzy logic and etc.) to detect failure and success of agile software projects, as follows:

V.Lalsing and et al, introduced a new process to assess people attributes in agile software projects based on time submission, rework standard, fault rate, and connection channels. This paper executed tests on little and moderate enterprises and using scrum method [1]. The importance of this research is to assess people attributes that influence of agile software projects by using a quantitative approach.

H. Taherdoost and et al, introduced a survey to display failure and success features of information software projects through organizational, people, process and technical features. This paper congrats on organizational features that utilize in enterprises concerned to agile methodologies [17]. The importance of this research is to use standard deviation to assess failure and success features that influence of information software projects.

M. Shepperd and et al, introduced a new approach based on unbiased statistic to detect success of agile software projects. This paper concentrates on various features such as organizational, people, process and technical features. This study seeks to detect the best method prediction of agile software projects [18]. The importance of this research is to use unbiased statistic for predicting success of agile software projects.

S. Lee and et al, introduced a new approach to assess agile method in little projects based on agility features that are consisted of elasticity, velocity, learning and responsiveness. This paper executed tests on little projects and using scrum method [19]. The importance of this research is to assess little projects in agile methodology by using agility features.

A. Elzamy and et al, introduced a new model based on risk features such as (determination risk, risk evaluation, risk handling, risk dominant and risk documentation) to evaluate success of agile software projects. This paper utilizes historical information from various enterprises to assess software projects [10]. The importance of this research is to use risk management equations to evaluate success of agile software projects.

M. Tanner and et al, presented a new approach to reveal crucial success features to enhance successful agile adoption based on customer sharing, stakeholder sharing, team construction, project kind, and skill level of team members. This paper seeks to gather historical information through questionnaires and meeting to experts to detect the crucial success features [20]. The importance of this research is to use mean analysis and standard deviation for determining crucial success features in agile software projects.

Feras A. Batarseh and et al, presented a modeling regression to reveal failure of agile software projects. The new method seeks to define kinds of software systems failures in various companies concerned to agile methodology [21]. The importance of this research is to use modeling regression to reveal failure of agile software projects.

D. S. Nguyen introduced a new approach to define success features that influence of agile software projects based on gathering historical information from a questionnaire. This paper can define crucial success features that influence of agile software projects such as customer sharing, perfect planning and continuous delivery products [22]. The importance of this research is to use time series for determining crucial success features in agile software projects.

T. Chow and et al, introduced a new approach based on multiple linear regression to determine crucial success features that influence of agile software projects. This paper displays three crucial success features: (1) Delivery Strategy, (2) Agile Software Engineering Techniques and (3) Team Capability in order to Quality, Scope, Time, and Cost [23]. The importance of this research is to use multiple linear regressions to determine crucial success features in agile software projects.

N. Cerpa and et al, introduced a new framework based on logistic regression to assess success features for predicting successful of agile software projects. This paper congrats on various success features such as customer sharing, perfect planning, project manager, and expansion process and development team. This paper displays that customer participation one of the most significant features that influences of agile software projects [24]. The importance of this research is to use logistic regression to predict successful of agile software projects.

R. P. Mohanty and et al, introduced a new model based on genetic algorithm (GA) to detect failure of agile software projects. This paper utilizes genetic algorithm to assess accuracy of agile software projects and detect crucial risk features [25]. The importance of this research is to use genetic algorithm to predict failure of agile software projects.

D. Stankovic and et al, presented a new approach based on multiple linear regression to detect crucial success features in information technology projects. This paper displays crucial success features such as good team, Project management process and agile software engineering [26]. The importance of this research is to use multiple linear regressions for determining crucial success features of information technology projects.

Pushpavathi T.P and et al, presented a new approach based on GA based fuzzy c-means clustering and random forest classifier to predict successful of agile software projects. This paper congrats on various success features such as Total project time and Defect counts estimation. This paper displays that the precise of the proposed approach is 93.05% [27]. The importance of this research is to use genetic algorithm based fuzzy c-means clustering and random forest classifier for predicting successful of agile software projects.

H. B. Yadav and et al, presented a new method to predict successful of agile software project by using fuzzy logic. This paper executed tests on 20 projects in various sizes in small companies [28]. The importance of this research is to use fuzzy logic for predicting successful of agile software projects.

T. Hovorushchenko and et al, presented a new model to predict successful of agile software project by using neural network. This paper displays the software project period and software project performance that influence of successful of agile software projects [29]. The importance of this research is to use neural network for predicting successful of agile software projects.

S.A. Rizvi and et al, presented a new method to predict early stage of successful of agile software project by using fuzzy logic. This paper displays that the precise of the proposed model is 98.4% [30]. The importance of this research is to use fuzzy logic for predicting early stage of successful of agile software projects.

V. Vashisht and et al, introduced a new model to predict successful of agile software project by using neuro - fuzzy. This paper displays that the precise of the proposed model is 93.4% [31]. The importance of this research is to use neuro - fuzzy model for predicting successful of agile software projects.

Through previous work, there are not official researches to define crucial failure features that effect on agile software projects. Crucial failure features of agile software projects are very significant for enterprises that combining agile method with software projects. This paper tries to detect crucial failure features of agile software projects for avoiding software projects to fail. It also aims to find the optimal intelligent techniques to reveal failure of agile software projects.

4. The Proposed Approach

This section presents an approach to reveal failure of agile software projects. An approach consists of three parts: as shown below in figure 4.

1. Survey recent researches to elicit significant failure features related to agile software projects.
2. Linear regression analysis is used to define crucial failure features in agile software projects.
3. Fuzzy logic is used to reveal failure of agile software projects.

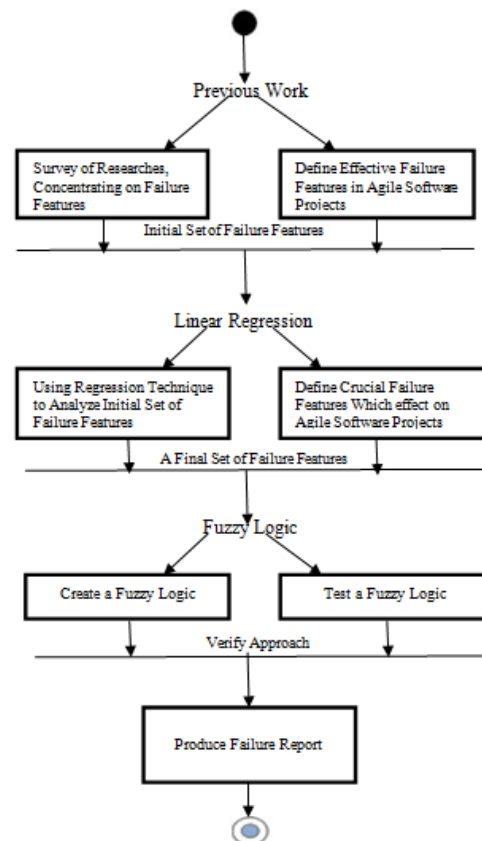


Figure.4 Proposed Approach for Revealing Failure of Agile Software Projects

Each component of the approach will be shown in the following subsections:

4.1 Survey of Manuscripts

The first part is to survey recent manuscripts for revealing failure of agile software projects. This part aims to elicit significant failure features in agile software projects. Elicit significant failure features are based on some estimation criteria (Transparency, Clarity and Easy) as shown below in figure 5.

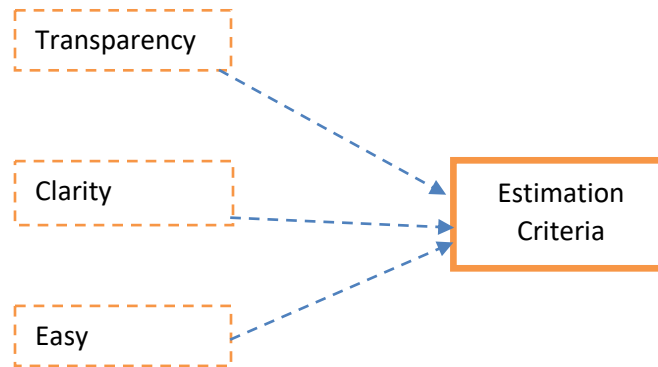


Figure.5 Estimation Criteria to Elicit Significant Failure Features

4.2 Linear Regression

Linear regression is used to define the crucial failure factors (CFF) in agile software projects. It consists of one dependent variable and independent variables. It is formulated as follows:

$$Y = \beta_0 + \beta_1 CFF_1 + \beta_2 CFF_2 + \dots + \beta_N CFF_N + \epsilon \quad (3)$$

Where:

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

This section presents a general linear regression algorithm to define crucial failure factors in agile software projects.

Algorithm 1. The general Algorithm of the Linear Regression Analysis to Define Crucial Failure Features that Effect on Agile Software Projects.

1. **Input :** α \leftarrow (dependent variable grade of influence the failure features of agile software projects)
2. σ \leftarrow (independent variables failure features that effect on agile software projects)
3. **Output :** ξ \leftarrow (a final list of crucial failure features of agile software projects)

4. **MLR = Multiple Linear Regression**
5. **PV = P-value**
6. **Build the MLR model based on the set of α and σ**
7. **Estimate the MLR model**
8. **Check the value of Standard Error.**
9. **Check the value of R- Squared.**
10. **Check the value of Adjusted R- Squared.**
11. **Check PV for each variable to define ξ**
12. **If (PV < 0.05)**
13. **Approve ξ**
14. **Else**
15. **Reject the other failure features**
16. **End If**

17. **Return ξ**

Linear regression is used to obtain the final list of crucial failure factors of agile software projects.

4.3 Fuzzy Logic

In this paper, the linguistic variables (light, medium-heavy and heavy) are used as input variables. The output variables are low, medium, high and very high. The triangular membership function is selected because it gives accurate results. As shown in figure 6, the main steps of the inference engine include: Fuzzification, knowledge base and Defuzzification.

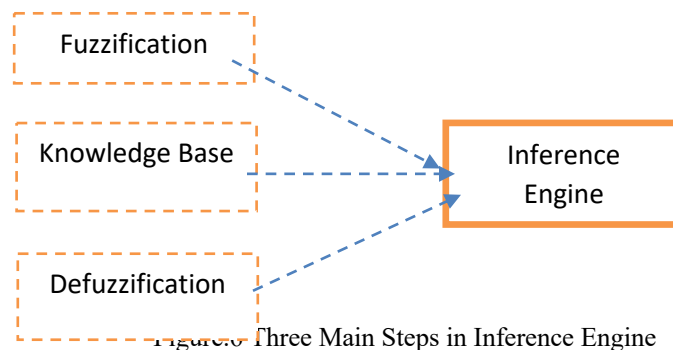


Figure 6 Three Main Steps in Inference Engine

Fuzzification is the process of replacing a current digit value (crucial failure factors in agile software projects) into a fuzzy value. Fuzzification uses linguistic values such as (light – medium-heavy and heavy) and clarifies it in membership function. Fuzzification process transforms crucial failure factors to linguistic values in membership function.

Knowledge base transforms input values (crucial failure factors in agile software projects) and output values (failure ruling in agile software projects) into if-then rules. An example of these rules is shown below:

If CFF1 is light and CFF2 is heavy then failure is Low

Defuzzification process transforms Fuzzification process (linguistic value of failure factors in agile software projects) into numerical values that the users can understand it.

This section also presents a general fuzzy logic algorithm to reveal failure of agile software projects.

Algorithm 2. The general Algorithm of the Fuzzy Logic to Reveal Failure of Agile Software Projects.

1. Input : α \leftarrow (a final list of crucial failure features of agile software projects)
2. Output : ξ \leftarrow (revealing failure of agile software projects)
3. LV = Linguistic Variables
4. MF = Membership Functions
5. F = Fuzzification
6. D = Defuzzification
7. IE = Inference Engine
8. Define LV in α and ξ
9. Create MF to α and ξ
10. Create the Rule Base
11. Transform Crisp Input Data to Fuzzy Values Using the MF (F)
12. Evaluate the Rules in IE
13. Combine the results of each rule in IE
14. Compute D
15. Check the D value for ξ
16. Return ξ

5. Challenges and Future Work

Future work seeks to enhance the precise to reveal failure of agile software projects by the linear regression to define the crucial features of failure and fuzzy model for revealing failure of agile software projects.

Thus, this paper will suggest a model based on hybrid intelligent techniques to reveal failure of agile software projects as shown below in figure 7.

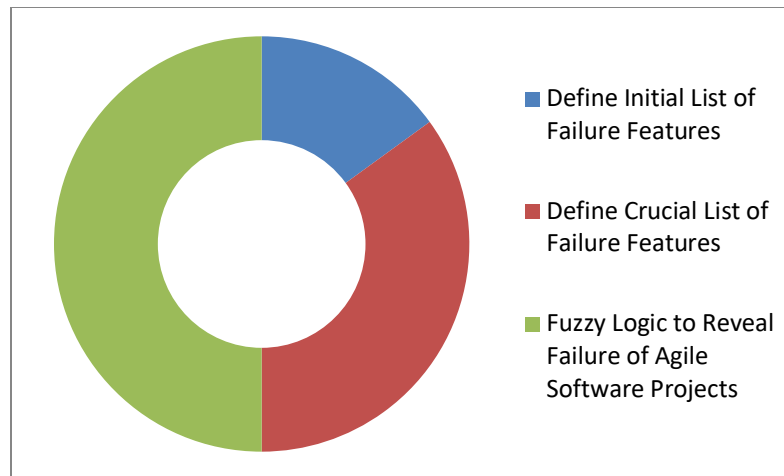


Figure.7 Future Work in Agile Software Projects

Future work seeks to realize three major targets as follows:

- Define initial list of failure features in agile software projects
- Define crucial list of failure features in agile software projects by linear regression
- Build fuzzy logic model to reveal failure of agile software projects.

6. Conclusion

Crucial failure factors of agile software projects are significant for software companies that are seeking for adopting agile method in their software projects. In this paper 17 works were reviewed related to revealing the failure of agile software projects and shows the advantages and disadvantage of each of them. This paper proposed a new approach for revealing the failure of agile software projects. This paper also presented intelligent techniques: linear regression and fuzzy logic. Linear regression was used to define crucial factors features in agile software projects. Fuzzy logic was used for revealing failure of agile software projects.

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] N. R. Darwish "Enhancements in Scrum Framework using Extreme Programming Practices", IJICIS, Vol.14, No.2, 2014, pp. 53-67.
- [9] Ioannis G. Stamelos and P. Sfetsos "Agile Software Development Quality Assurance", information science references, Idea Group Inc, 2007.
- [10] V. E. Jyothi and K. N. Rao "Effective Implementation of Agile Practices", IJACSA, Vol.2, No.3, 2011, pp. 41-48.

- [11] 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.
- [12] S. MERZOUK, S. ELHADI, H. ENNAJI, A. MARZAK and N. SAEL “A Comparative Study of Agile Methods: Towards a New Model-based Method”, IJWA, Vol.9, No.4, 2017, pp. 121-128.
- [13] M. Al-Zawahiri, M. Biltawi, W. Etaiwi, A. Shaout “Agile Software Development Methodologies: Survey of Surveys”, Journal of Computer and Communications, Vol.5, 2017, pp. 74-97.
- [14] 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.
- [15] 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.
- [16] K. Jammalamadaka and V. R. Krishna “Agile Software Development and Challenges”, IJRET, Vol.2, No.8, 2013, pp. 125-129.
- [17] 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.
- [18] M. Shepperd and S. MacDonell “Evaluating prediction systems in software project estimation”, Information and Software Technology, ELSEVIER, vol.54, 2012, pp. 820-827.
- [19] S. Lee and H. Yong “Agile Software Development Framework in a Small Project Environment”, JIPS, Vol.9, No.1, 2013, pp. 69-88.
- [20] Tanner and U. Willingh “Factors Leading to the Success and Failure of Agile Projects Implemented in Traditionally Waterfall Environments”, MKL, 2014, pp. 693-701.
- [21] Feras A. Batarseh and Avelino J. Gonzalez “Predicting failures in agile software development through data analytics”, Springer, 2015.
- [22] D. S. Nguyen “Success Factors That Influence Agile Software Development Project Success”, ASRJETS, vol.17, No 1, 2016, pp. 172-222.
- [23] T. Chow, D. Cao “A survey study of critical success factors in agile software projects”, JSS, Elsevier, vol.81, 2008, pp. 961-971.
- [24] 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.
- [25] M. Mayo, and S. Spacey “Predicting Regression Test Failures Using Genetic Algorithm-Selected Dynamic Performance Analysis Metrics”, Proceedings of the 5th International Symposium on Search Based Software Engineering, Springer, vol.8084, 2013, pp. 158-171.
- [26] D. Stankovic, V. Nikolicb, M. Djordjevicc, 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.
- [27] 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.
- [28] H. Yadav, D. Yadav “A fuzzy logic based approach for phase-wise software defects Prediction using software metrics”, INFISO, 2015, vol.1, pp. 1-19.
- [29] 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.
- [30] 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.
- [31] 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