**International Journal of Research in IT and Management (IJRIM)**
Available online at: http://euroasiapub.org
Vol. 13 Issue 03, March- 2023
**ISSN(o): 2231-4334 | ISSN(p): 2349-6517 | Impact Factor: 8.106**

# AN ANLAYSIS ON THE ESTIMATION OF MAINTAINABILITY FOR AGILE SOFTWARE PRODUCT

**Deepika**
Research Scholar, Department of Computer Science Engineering, OPJS University

**Dr Vijaypal Singh**
Associate Professor, Department of Computer Science Engineering, OPJS University

**Dr Suman Rani**
Associate Professor, Department of Electronics and communication Engineering, OPJS University

## ABSTARCT
When it comes to the creation of a programme or software product, quality is an extremely important factor. The quality of the software process that is used for building programming frameworks has an effect on the level of customer loyalty overall. The primary objective of this study is to carry out in-depth research on the evaluation of maintainability for agile software solutions. The focus of this study is to present a model that integrates the most salient features of fuzzy logic systems (FLS) and neural networks (NN) into an agile maintainability estimation model. The findings indicate a high degree of precision, which provides reassurance that the proposed model could be utilised appropriately for the purpose of evaluating the maintainability of software programming products generated utilising a lightweight technique.
***Keywords:*** Agile software product; Software development; Fuzzy Logic Systems (FLS); Neural Networks (NN).

## INTRODUCTION
Today, selecting a method that is appropriate for creating software that complies with established standards and releases a high-quality result is necessary. As a programming product is being developed, quality is important. The level of customer loyalty is influenced by the quality of the software used to design the programming frameworks. The quality of the programme depends on a number of distinct factors. Maintainability is a critical component of the quality factors for agile processes, which are based on the ISO 9126-1 quality model. Maintenance is seen as a key step in the software development process for expanding and improving software programming capabilities (Franch et al., 2018).
According to the definition provided by IEEE, the formal definition of software maintainability is "the modification of a software product after delivery in order to correct faults, to improve the performance or other attributes, or to adapt to a modified environment (Alsolai, Roper, and Nassaras, 2018)." Maintainability, in most cases, refers to the manner in which these actions done satisfactorily. According to IEEE, the activity of software maintenance is the one that demands the most effort and takes the most time to properly carry out, making it the costliest activity in the software life cycle for development (Jha et al., 2019). This is something that is always seen when developing software. Examining the upkeep of the programming product is one way to bring down the expense. The evaluation of such attributes, on the other hand, is not as straightforward and can never produce an output that is precisely one hundred percent.
In this study, the estimation of maintainability for agile software products is the primary focus. The previous literatures on this subject are elaborated upon in the next section.

**International Journal of Research in IT and Management (IJRIM)**
Available online at: http://euroasiapub.org
Vol. 13 Issue 03, March- 2023
**ISSN(o): 2231-4334 | ISSN(p): 2349-6517 | Impact Factor: 8.106**

## LITERATURE REVIEW

**Jain, Sharma & Ahuja (2018)** found that Agile methods are widely used to produce high-quality products quickly and efficiently. Agile process quality has not yet been quantified. Software quality requires measurement. Estimating quality factors yields a good product. Agile maintainability is estimated using an adaptive neuro-fuzzy inference system (ANFIS) model. Agile development values maintainability.

The authors **Jain, Sharma & Ahuja (2019)** stated that product quality is crucial today. This lets buyers customise the product fast. As software product quality depends on maintainability, the paper proposes a model for measuring maintenance value for agile projects. Step-Wise Weight Assessment Ration Analysis (SWARA) uses decision-making to determine weight values for each given maintainability criterion. The results allow estimating maintenance value for agile software development products.

The Researchers **Charles & Caesar (2021)** have used many quality criteria to quantify software quality due to the effort and difficulty involved. Software quality is determined by each development process's standards and quality attributes and software engineering principles. Twelve open-source software projects were examined for six quality aspects. Each result is quantified and reported. SW2's maintainability, availability, and reliability metrics are 6 minutes, 50%, and 0.62, respectively. It implies that high maintainability does not equal high reliability. These values define the relationship between qualities and improve developers' and consumers' knowledge of software quality and attributes.

Although agile methodologies have been extensively used by software development organizations for the development of small- and large-scale software products, it is evident from the literature that corporations are still not convinced of the agile system's overall superiority.

## METHODOLOGY

For the purpose of evaluating the evolution of software maintainability in lightweight environments based on software impediment, an ANFIS model has been presented. Fuzzy logic frameworks have been used in a variety of disciplines of designing for evaluations in the past, and they have seen an exceptional amount of success in doing so. The key challenge in developing FIS is finding fuzzy sets and fuzzy inference rules that can be used by the system to generate output. As a result, it is beneficial to use FIS in conjunction with the learning capabilities of NNs as a mix, which is termed as ANFIS.

*STEP 1 - Identification of factors:* The agile technique differs from established methodologies in the software development life cycle. In the agile methodology, the product owner first compiles a list of user stories that need to be built into a product backlog. It contains all the information needed to create the whole software product. In the agile software development process, a user story refers to a specific functionality. sprint refers to the many iterations used to produce these stated user stories.

*STEP 2 - Collection of Data sets:* The data collection was compiled from a variety of industry sources that have been active in the agile software development field for many years. The data set contains information on 93 different software projects in total.

*STEP 3 - Model Evaluation:* For the purpose of assessing the suggested model, a fuzzy inference system of the Sugeno type is created. It is an ANFIS framework that makes use of both fuzzy and neural network theory. The framework first undergoes training in order to comprehend the network and create rules accordingly, after which it provides output while incorporating part of the inputs utilised during training. The framework is then tested with data that is distinct from the training data set to produce the desired results. For the proposed framework, 93 data points from active software projects are gathered, of which 70 are used to train the framework and the remaining 23 are used to test the framework as it has been created. The four attributes listed above—three inputs and one output—are used to evaluate the framework.

**International Journal of Research in IT and Management (IJRIM)**
Available online at: http://euroasiapub.org
Vol. 13 Issue 03, March- 2023
**ISSN(o): 2231-4334 | ISSN(p): 2349-6517 | Impact Factor: 8.106**

## RESULTS AND DISCUSSIONS

The software maintenance created using lightweight methods is evaluated using the ANFIS model. The ANFIS framework is tested on the available data set of agile created projects after being trained in order to produce the findings. The model may be validated as shown in the following figure thanks to the testing mistake that the framework predicted. The framework's effectiveness is tested by calculating the average inaccuracy for the entire set of collected data. The result specifies a precision of about 85%, enabling the engineers to effectively use the suggested model and anticipate post-defects before delivering the product to the customer. One can forecast the upkeep of the developed product while it is still in development by examining the post-defects.

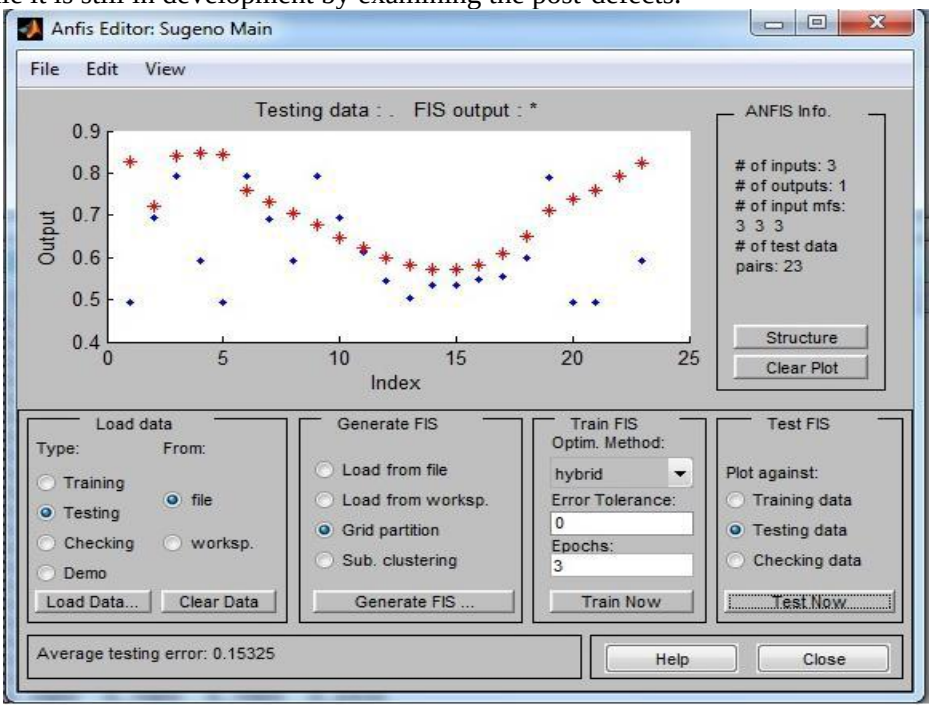


Figure 1: Testing of proposed model.

All of the members of the development team are given access to an agile environment in order to ensure that the projects are developed in accordance with the agile principles. Approximately one-half of a month is required to complete any job from start to finish. The following table provides an outline of all the projects that must be completed in order to verify the proposed framework:

Table 1: Project Details

| PROJECTS | USER STORIES | STORY POINTS | PRE – DEFECTS | ACTUAL POST - DEFECTS |
|---|---|---|---|---|
| Attendance Monitoring System | 18 | 31 | 35 | 2 |
| End Semester Result Preparation | 12 | 28 | 18 | 4 |
| Fee Module System | 10 | 21 | 25 | 6 |
| Online Complaint Management System | 19 | 33 | 15 | 3 |

**International Journal of Research in IT and Management (IJRIM)**
Available online at: http://euroasiapub.org
Vol. 13 Issue 03, March- 2023
**ISSN(o): 2231-4334 | ISSN(p): 2349-6517 | Impact Factor: 8.106**

## CONCLUSION

The proposed model is applied to the specified four classroom-based projects, and the input and output parameter values are gathered once the projects are fully finished. The outcomes are contrasted with the real outcomes, which are the quantity of post-defects resulting from the projects and the value for the same originating from the ANFIS framework. Since the post-defect values obtained from the framework are normalised values, they must first be de-normalized and rounded off in order to obtain the precise number of post-defects. Using the framework, the outcome indicates that there were 3, 3, 7, and 5 post-defects for each project, correspondingly. A high degree of agreement between the post-defect values obtained using the proposed framework and the real ones indicates acceptance of the proposed technique.

## REFERENCES

1. Alsolai, H., Roper, M., & Nassar, D. (2018, September). Predicting software maintainability in object-oriented systems using ensemble techniques. In *2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (pp. 716-721). IEEE.
2. Charles, I., & Caesar, N. I. (2021). Quantifying software quality in agile development environment. *Software Engineering*, *9*(2), 36-44.
3. Franch, X., Gómez, C., Jedlitschka, A., López, L., Martínez-Fernández, S., Oriol, M., & Partanen, J. (2018). Data-driven elicitation, assessment and documentation of quality requirements in agile software development. In *Advanced Information Systems Engineering: 30th International Conference, CAiSE 2018, Tallinn, Estonia, June 11-15, 2018, Proceedings 30* (pp. 587-602). Springer International Publishing.
4. Jain, P., Sharma, A., & Ahuja, L. (2018). Software Maintainability Estimation in Agile Software Development. *International Journal of Open Source Software and Processes (IJOSSP)*, *9*(4), 65-78.
5. Jain, P., Sharma, A., & Ahuja, L. (2019, April). The Model for Determining Weight Coefficients of Maintainability Criteria in Agile Software Development Process. In *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)* (pp. 1-4). IEEE.
6. Jha, S., Kumar, R., Abdel-Basset, M., Priyadarshini, I., Sharma, R., & Long, H. V. (2019). Deep learning approach for software maintainability metrics prediction. *Ieee Access*, *7*, 61840-61855.