ALKU Journal of Science 2021, Sayı 3(3): 9-24
e-ISSN: 2667-7814

Original Article

Data Mining based Inferences and Fuzzy based Rules about Software Development

Mustafa Batar[1*], Kökten Ulaş Birant[2]
[1]*Burdur Mehmet Akif Ersoy University, Faculty of Engineering and Architecture, Department of Computer Engineering, Burdur, Turkey*
[2]*Dokuz Eylül University, Faculty of Engineering, Department of Computer Engineering, İzmir, Turkey*
**mbatar@mehmetakif.edu.tr*

Abstract

Up to now, several criteria (software parameters) have been determined in order to measure and evaluate software development projects: Productivity, engagement, attention to quality, code base knowledge and management, adherence to coding guidelines and techniques, learning and skills, personal responsibility and etc. However, there isn't any universally accepted criteria or a methodology to measure and evaluate software development projects. In this context, the common criteria set about measurement and evaluation of software development projects has been tried to be created in this study. In addition, some information has been taken from 105 software experts with 55 different software companies so as to evaluate the use of the common criteria in real work life. Accordingly, a measurement and evaluation criteria set (software parameters) about the software development projects has been formed based on the data mining algorithm – *"Association Rule Mining Apriori Algorithm"* – with its 12 inferences. By the contribution of these data, the designed and developed software parameters have had more than 75 percent accuracy rate in the data mining algorithm. Finally, 8 important fuzzy-based rules about these 12 inferences have been formed and generated in this work.

Keywords: Software engineering, software development, software parameters, software rules, data mining, Apriori algorithm, fuzzy logic.

Yazılım Geliştirme ile İlgili Veri Madenciliğine Dayalı Çıkarımlar ve Bulanık Mantığa Dayalı Kurallar

Özet

Yazılım projelerinin ölçülmesi ve değerlendirilmesine yönelik bugüne kadar çeşitli kriterler ve parametreler belirlenmiştir: Verimlilik, katılım, kaliteye dikkat, kod tabanlı bilgi ve yönetim, kodlama ile ilgili yönerge ve tekniklerine uyum, öğrenme ve beceriler, kişisel sorumluluk vb. Bununla birlikte, yazılım geliştirme projelerini ölçmek ve değerlendirmek için evrensel olarak kabul görmüş herhangi bir yöntem ya da kriter maalesef ki yoktur. Bu bağlamda, bu çalışma kapsamında, yazılım geliştirme projelerinin ölçülmesi ve değerlendirilmesi ile ilgili ortak kriterler ortaya konmaya çalışılmıştır. Ayrıca, bu ortak kriterlerin gerçek iş hayatında kullanımını değerlendirmek amacıyla da 55 farklı yazılım şirketinde çalışan 105 yazılım uzmanından) konu ile ilgili çeşitli bilgiler elde edilmiştir. Bu doğrultuda, veri madenciliği algoritması – "Association Rule Mining Apriori Algorithm" – kullanılarak 12 tane çıkarım yapılmış ve bunun neticesinde, yazılım geliştirme projeleri ile ilgili bir ölçme ve değerlendirme yazılım parametre seti oluşturulmuştur. Bu elde edilen verilerin doğrultusunda tasarlanan ve geliştirilen yazılım parametreleri, ortaya konan veri madenciliği algoritması çerçevesinde yüzde 75'in üzerinde yüksek bir doğruluk oranına sahip olmuştur. Bununla birlikte, yazılım geliştirme üzerine elde edilen 12 çıkarımla ilgili bulanık mantığa dayalı 8 tane hayati öneme sahip kural oluşturulup çalışmada ortaya konmuştur.

Geliş/Received: 22 Ağu 2021 / 22 Aug 2021
Kabul Ediliş/Accepted: 6 Aral 2021 / 6 Dec 2021

Anahtar Kelimeler: Yazılım mühendisliği, yazılım geliştirme, yazılım parametreleri, yazılım kuralları, veri madenciliği, Apriori algoritması, bulanık mantık.

1. INTRODUCTION

Measurement is the task of numerals to questions or occasions as indicated by some standards. Numerals are the marks which have no inalienable significance, such as postal districts and car tags. Numbers are the numeric which have quantitative importance and could be investigated, such as weight and height. The guidelines for allotting marks to properties of variables are the most critical segments of estimation, as the result of the fact that the consequence of poor principles is futile results. Ideas frequently could not be estimated straightforwardly, so what are generally estimated are markers of builds; for example, speed, rationale, verbal aptitude and so on [1].

Evaluation is the deliberate (methodical) gathering of data about the attributes, exercises and results of administrations or projects to evaluate the degree to which targets have been attained to, recognize required changes and/or settle on choices about future programming [2].

Measurement and evaluation of software development projects is a seriously critical task. Software projects often have objectives which are about maintainability, reliability, efficiency, security, flexibility, testability and etc. Unfortunately, it is often unclear which software developer deals with which part of a software project and how their change affects achievement of those goals. There are three main reasons of this unclear situation: Firstly, each part of software creation is unique. There is no compelling reason to assemble two times the same parts of software as it might be duplicated by copying it. This makes it truly difficult to make a formal and thorough correlation between two parts of software. Secondly, the current technology is something that changes at a truly fast phase. So, each time a methodology in respect to a certain wave of technology is dependable enough, it is for the most part as of recently old. Thirdly, there is a gigantic zone for innovativeness in discovering the diverse answers for a unique issue. Thus, measuring the effect in terms of "lines of code", for instance, is measuring the size of the solution, not the issue. With this work, some relationships between measurable development criteria and objectives are tried to be defined and some evaluation techniques tried to be defined for better estimation in very early times of software development.

2. RELATED WORKS

Gallivan (1998) has showed the relationship between job and profession about measurement and evaluation of software development process in his study. He has indicated several special points in this relationship, such as job satisfaction and difficulty, actual performance, technical knowledge about job, analytical thinking skill, verbal skill, working habits, being open to new ideas and creativity, and etc. [3].

Sawyer and Guinan (1998) have showed several points about working as a team in software development for measurement and evaluation of software development process in their study. Those points are about team support, team loyalty, team vision, team personality, team meeting, team members and team leader. In addition, they have tried to find the answers to some questions about software development. Those questions are about software development method, code storage, code library, working time and documentation about development [4].

Hall, Wilson, Rainer and Jagielska (2007) have tried to find the answers to several questions about several topics for measurement and evaluation of software development process in their study. Those topics are about software team, software project, job life, working and personality [5].

Baggelaar (2008) has showed importance of several properties in software development process. Those properties are abstraction, testability, coupling, modularity, layouts, test coverage, error handling and exception handling. In addition, he has tried to find the effect of code and comment lines to measurement and evaluation of software development [6].

Lee, Joshi and Kim (2008) have analyzed measurement and evaluation of software development process with regards to personality and working habits in the work [7].

Thing (2008) has dwelt on several topics about personality, working style, work load and software development process in measurement and evaluation of software development [8].

Zhang, Wang and Xiao (2008) have asked several questions and have got some answers to them about measurement and evaluation of software development process in their study. Those questions are about code lines, comment lines, class number, instance number, class relationship, method number, inheritance depth degree and difficulty of software development [9].

Calikli and Bener (2010) have indicated general information about measurement and evaluation of software development process in their study. They have showed the effect of education level of software developers and some points in software development area, such as satisfaction degree, trust degree, working experience and etc. to software development [10].

Chilton, Hardgrave and Armstrong (2010) have showed several points in measurement and evaluation of software development process in their workday. Those points are about working life, working habits, personality, age and gender [11].

Ramler, Klammer and Natschläger (2010) have tried to find the answers to some questions about software quality in measurement and evaluation of software development process in their study [12].

Wang and Zhang (2010) have showed importance of several topics in measurement and evaluation of software development process in their work. Those topics are about working life, working experience, work load, education level and gender [13].

Balijepally, Nerur and Mahapatra (2012) have tried to find the answers to several questions about pair programming, difficulty of software development, software quality and personality for measurement and evaluation of software development in their study [14].

Duarte, Faria and Raza (2012) have tried to find the effect of some issues to measurement and evaluation of software development process in their study. Those issues are about timing error, size error, segmentation error, missing parts, irrelevant parts, error number and unit test number [15].

Ehrlich and Cataldo (2012) have dwelt on some points in software project development process for measurement and evaluation of software development in their work. Those points are about team leader, team coordination, company management, company employees and private life [16].

Kelly and Haddad (2012) have tried to find the effect of "error" to measurement and evaluation of software development process in their study [17].

Schröter, Aranda, Damian and Kwan (2012) have tried to find the answers to several questions for measurement and evaluation of software development process in their study. Those questions are about build number, code changes, method number, fixed code parts, working life, working quality, team leader, software project documentation and software development tool [18].

Westermann (2012) has indicated importance of some points in software development process in his study. He has searched the effect of reliable code writing, software project outcomes and working style to measurement and evaluation of software development [19].

Calikli and Bener (2013) have showed some important points in measurement and evaluation of software development process in their study. Those points are about software project development schedule and software development team psychology [20].

Kumar, Santhosh, Kalaikumaran, and Karthik (2013) have examined data mining and customer relationship management models and tried to propose a new data mining model for customer relationship management.

Customer loyalty and profit increase have been observed in the companies where the method has been applied [21].

In their study, Keleş and Kaya (2014) have applied the Apriori algorithm to find out how factors such as the number of workers in the group, age and experience levels of masonry workers affect the daily productivity value. They have aimed to find rules aimed at ensuring that productivity could be increased [22].

Akhondzadeh-Noughabi, Amin-Naseri, Albadvi, and Saeedi (2016) have proposed a new approach for personnel evaluation using data mining techniques. Complaints and satisfactions of citizens about personnel and jobs in different units of a company have been collected. Performance evaluations of the units have ben made using the Apriori algorithm. In this way, it has been aimed to increase satisfaction by making necessary improvements [23].

Xi, J., Zhao, Z., Li, W., and Wang, Q. (2016) have listed the causes of traffic accidents according to their relative importance and determined the most effective ones with the AHP method. They then have used the Apriori algorithm to analyze the extent or impact level of the accident [24].

Chiang (2018) have suggested a data mining application to be used in marketing strategy for companies selling online. In the study, a priori and fuzzy clustering algorithms have been used to analyze the customers [25].

Taş (2018) and Gedleç (2019) have examined association analysis rules and association relations in sales data. In the study of Ayberkin and Özen in 2019, a software that analyzes data has been designed using the Apriori algorithm [26,27].

A similar study has been conducted by Rokhayati, Rusdi, Kurniawan, Janah, and Irawan (2019). In the study, the factors affecting the absenteeism of students have been investigated by using AHP and data mining. With the AHP method, the five criteria that most affect absenteeism have been found. Afterwards, their relations with each other were examined using the Apriori algorithm [28].

In 2019, data mining has been carried out on energy consumption data with different algorithms by Güler. Also, Gürcanok (2020) have examined the trends and habits of consumers through e-commerce sales data, with data mining [29,30].

3. ASSOCIATION RULE MINING APRIORI ALGORITHM

“Association Rule Mining Apriori Algorithm” [31] is a calculation for incessant thing set mining and affiliation principle learning over value-based databases. It continues by distinguishing the continuous individual things in the database and stretching out them to larger and larger item sets in the length which those thing sets show up adequately frequently in the database. The successive thing sets dictated by Apriori [32] can be utilized to focus affiliation rules which highlight general patterns in the database: this has applications in spaces, for example, market basket analysis.

Data mining methods that analyze the co-occurrence of events are called association rule methods. The association rule is an approach that supports the analysis of past data and the determination of association behaviors in these data and future studies. Market basket analysis can be given as an example of association rule application. The purpose of the association rule is to find the association relationship between the products purchased by the customers during shopping, and to determine the purchasing habits of the customers in line with this relationship data. Sellers have effective and profitable marketing and sales opportunities with the contribution of these discovered association relations and habits. For example, in 70% of customers’ purchases of milk and cheese from a grocery store, yoghurt was purchased along with these products. In order to detect this kind of association pattern, the products included in the pattern has to be included in more than one purchasing activity [33].

In addition to these, Apriori (meaning "prior") obtains the information from the previous step. Based on this algorithm, it has an iterative (repetitive) nature and is used to discover frequent item sets in databases containing motion information. Association rule mining has two stages: finding all the frequent items and generating strong association rules from these frequent items. The Apriori Algorithm, which is used for the first stage of the association rule, is the most popular and classical algorithm used in frequent items mining. In this algorithm, features and data are evaluated with "boolean" association rules [34].

The Apriori algorithm was developed by Agrawal and Srikant in 1994 for association rules. Association rules are rules that check whether an element exists. This algorithm has been designed for databases containing transactions and timestamps. Every transaction is a set of items [35]. In the Apriori algorithm, the relationship between items is calculated for certain support and confidence values. As these two values increase, the importance of the combination of the two elements also increases. Confidence is the value that indicates how likely one item will be with another item. The support value is the value that shows how often two items are found together [36]. The support value is very important for the interpretation of the resulting rules. Rules with very low support values were most likely created by chance. In addition, infrequent rules will not matter to the decision maker. If the support value is too high, it is possible to miss the rules that may be important for the decision maker [37].

The purpose of the algorithm is to find the over-repeated associations by scanning the database of transactions with different associations. The algorithm consists of two operations, pruning and merging. During the process, the database is scanned many times. The K item set is used to find the frequently repeated K+1 item set in the algorithm. The first scan finds single-element item sets and their support values. How often each item is used determines the support value of that item. Item sets with more elements are created with items that provide the specified support value. This process continues until the item set that provides the desired support value is not found [38].

Furthermore, to understand better this method, the pseudocode of Apriori algorithm has been shown in Figure 1 and its working mentality has been demonstrated in Figure 2 in the following.

```
procedure Apriori (T, minSupport) { //T is the database and minSupport is
                                    //the minimum support
     L1 = {frequent items};
     for (k=2; Lk-1!=∅; k++) {
           Ck = candidates generated from Lk-1
           //that is cartesian product Lk-1 x Lk-1 and eliminating any k-1
           //size itemset that is not frequent
           for each transaction t in database do{
             #increment the count of all candidates in Ck that are
             #contained in t
             Lk = candidates in Ck with minSupport
           }//end for each
     }//end for
     Return UkLk;
}
```

Figure 1. Pseudocode of Apriori algorithm [39].

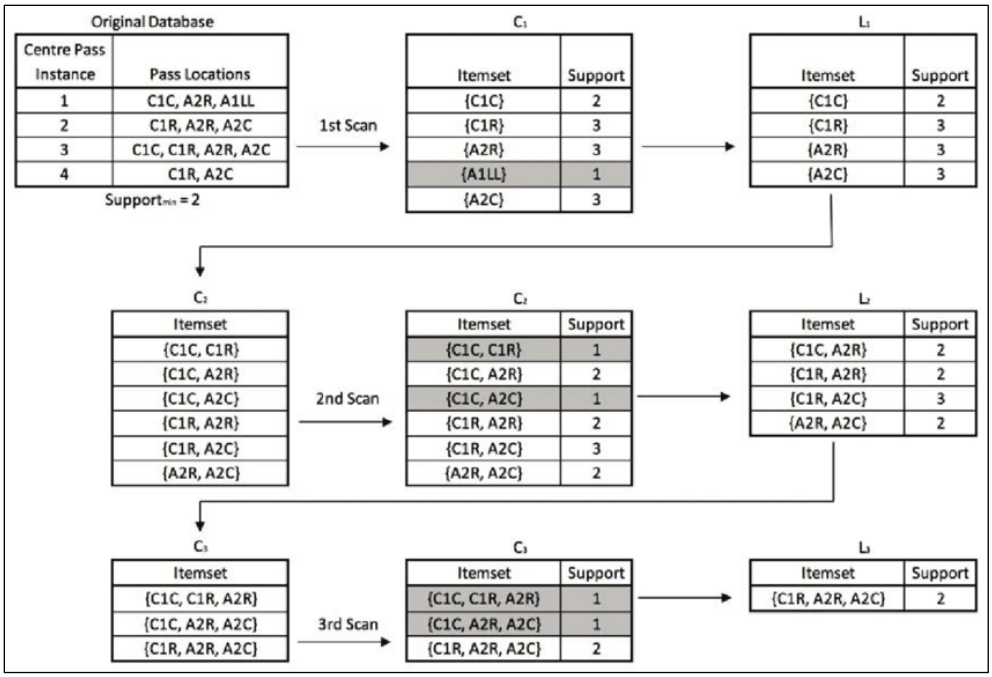

Figure 2. Process of Apriori algorithm [40].

4. DATA MINING BASED DISCOVERED INFERENCES

Apriori Algorithm is one of the best-known association rule mining algorithms. It uses prior knowledge of frequent itemset properties and runs an iterative approach called level-wise search. That is, k−itemsets are used to explore (k + 1)−itemsets (they are called candidate itemsets before testing them against the database) by eliminating the candidates that do not satisfy the minimum support. This process terminates when no frequent or candidate set can be generated. The efficiency of the level-wise generation of frequent itemsets is improved by the Apriori Property: "All nonempty subsets of a frequent itemset must be frequent". By means of this property, many unnecessary candidate generation and support counting are eliminated [41].

Based on the "Association Rule Mining Apriori Algorithm", 12 inferences with 5 parameters in each have been extracted and leapt to the eye about measurement and evaluation of software development while designing and implementing software, by the help of the real software experts, who have given information about the study and research. Also, the number of these developers and experts has been 105, and they have worked at 55 different countries about software development. Among these companies, whereas there have been big institutive global companies such as Ericsson, Huawei Technologies, Siemens, Alcatel Lucent and OBSS, on the other hand, there have been startup companies where about 5 people work such as Metadata, Semafor and Mavikent Bilisim. Therefore, it can be said that the software experts, who have given information about the software parameters, have been from wide range of the companies in software industry. In addition, those extracted twelve inferences have been determined and identified based on the software experts' opinions, and these opinions have been given in the same direction (in parallel) by at least 75 percent accuracy rate with the contribution of the implementation and application of "Association Rule Mining Apriori Algorithm".

In the following, the file format of the taken software experts' thoughts have been given in Figure 3, and the operation of these in the application of "Apriori Algorithm" have been shown in Figure 4.

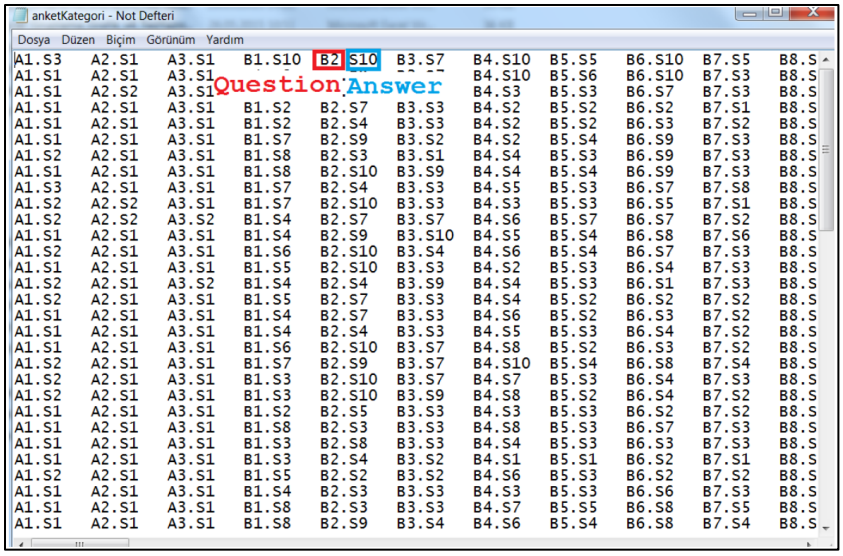

Figure 3. The file format of the taken software experts' thoughts.

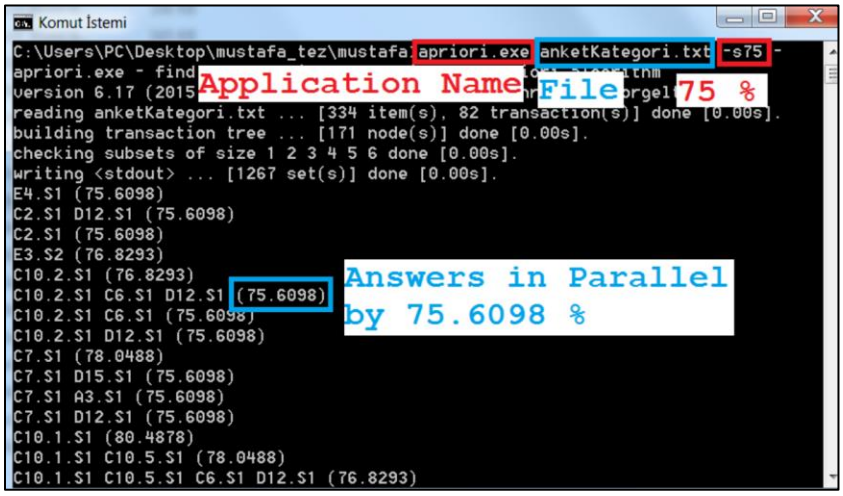

Figure 4. Apriori algorithm with at least 75 percent accuracy rate.

The discovered inferences, which have been obtained based on data mining, have been listed and explained in the following subtitles – 4.1 to 4.12.

4.1 Inference-1

- *Do you reuse any code parts while writing (developing) a program?*
- *Do you use any software development method while writing (developing) a program?*
- *Do you use any methods or models which have been determined before while writing (developing) a program?*
- *Do you use the property of Error Handling while writing (developing) a program?*
- *Individually, do you pay attention to the quality in the program which you have written (developed)?*

About 80 of the software experts (105), who have given their opinions about the software parameters, have answered these questions in the same direction – *YES.*

4.2 Inference-2

- *Do you reuse any code parts while writing (developing) a program?*
- *Do you use any software development method while writing (developing) a program?*
- *Do you use any methods or models which have been determined before while writing (developing) a program?*
- *Do you use the property of Modularity while writing (developing) a program?*
- *Individually, do you pay attention to the quality in the program which you have written (developed)?*

About 82 of the software experts (105), who have given their opinions about the software parameters, have replied these questions in the same direction – *YES.*

4.3 Inference-3

- *Do you use any software development method while writing (developing) a program?*
- *Do you use any methods or models which have been determined before while writing (developing) a program?*
- *Do you use the property of Modularity while writing (developing) a program?*
- *Do you use the property of Error Handling while writing (developing) a program?*
- *Individually, do you pay attention to the quality in the program which you have written (developed)?*

About 83 of the software experts (105), who have given their opinions about the software parameters, have answered these questions in the same direction – *YES.*

4.4 Inference-4

- *Do you use any methods or models which have been determined before while writing (developing) a program?*
- *Do you use the property of Reusability while writing (developing) a program?*
- *Do you think that you have written (developed) a practical (usable&understandable) program?*
- *As company, do you guarantee the quality of your software?*
- *Individually, do you pay attention to the quality in the program which you have written (developed)?*

About 81 of the software experts (105), who have given their opinions about the software parameters, have replied these questions in the same direction – *YES.*

4.5 Inference-5

- *Do you use any methods or models which have been determined before while writing (developing) a program?*
- *Do you use the property of Error Handling while writing (developing) a program?*
- *Do you think that you have written (developed) a practical (usable&understandable) program?*
- *As company, do you guarantee the quality of your software?*

- *Individually, do you pay attention to the quality in the program which you have written (developed)?*

About 81 of the software experts (105), who have given their opinions about the software parameters, have answered these questions in the same direction – *YES.*

4.6 Inference-6

- *Do you use any methods or models which have been determined before while writing (developing) a program?*
- *Do you use the property of Modularity while writing (developing) a program?*
- *Do you think that you have written (developed) a practical (usable&understandable) program?*
- *As company, do you guarantee the quality of your software?*
- *Individually, do you pay attention to the quality in the program which you have written (developed)?*

About 82 of the software experts (105), who have given their opinions about the software parameters, have replied these questions in the same direction – *YES.*

4.7 Inference-7

- *Do you use any methods or models which have been determined before while writing (developing) a program?*
- *Do you use the property of Modularity while writing (developing) a program?*
- *Do you use the property of Reusability while writing (developing) a program?*
- *Individually, do you pay attention to the quality in the program which you have written (developed)?*
- *Do you depend on the mechanism in the software development process?*

About 84 of the software experts (105), who have given their opinions about the software parameters, have answered these questions in the same direction – *YES.*

4.8 Inference-8

- *Do you use any methods or models which have been determined before while writing (developing) a program?*
- *Do you use the property of Error Handling while writing (developing) a program?*
- *Do you use the property of Reusability while writing (developing) a program?*
- *Do you think that you have written (developed) a practical (usable&understandable) program?*
- *Individually, do you pay attention to the quality in the program which you have written (developed)?*

About 83 of the software experts (105), who have given their opinions about the software parameters, have replied these questions in the same direction – *YES.*

4.9 Inference-9

- *Do you use any methods or models which have been determined before while writing (developing) a program?*
- *Do you use the property of Modularity while writing (developing) a program?*

- *Do you use the property of Reusability while writing (developing) a program?*
- *Do you think that you have written (developed) a practical (usable&understandable) program?*
- *Individually, do you pay attention to the quality in the program which you have written (developed)?*

About 85 of the software experts (105), who have given their opinions about the software parameters, have answered these questions in the same direction – *YES.*

4.10 Inference-10

- *Do you reuse any code parts while writing (developing) a program?*
- *Do you use any methods or models which have been determined before while writing (developing) a program?*
- *Do you use the property of Modularity while writing (developing) a program?*
- *Do you use the property of Reusability while writing (developing) a program?*
- *Individually, do you pay attention to the quality in the program which you have written (developed)?*

About 86 of the software experts (105), who have given their opinions about the software parameters, have replied these questions in the same direction – *YES.*

4.11 Inference-11

- *Do you use any methods or models which have been determined before while writing (developing) a program?*
- *Do you use the property of Modularity while writing (developing) a program?*
- *Do you use the property of Error Handling while writing (developing) a program?*
- *Do you use the property of Reusability while writing (developing) a program?*
- *Individually, do you pay attention to the quality in the program which you have written (developed)?*

About 82 of the software experts (105), who have given their opinions about the software parameters, have answered these questions in the same direction – *YES.*

4.12 Inference-12

- *Do you use any methods or models which have been determined before while writing (developing) a program?*
- *Do you use the property of Modularity while writing (developing) a program?*
- *Do you use the property of Error Handling while writing (developing) a program?*
- *Do you think that you have written (developed) a practical (usable&understandable) program?*
- *Individually, do you pay attention to the quality in the program which you have written (developed)?*

About 89 of the software experts (105), who have given their opinions about the software parameters, have replied these questions in the same direction – *YES.*

5. FUZZY LOGIC

Fuzzy logic is communicated as a methodology dependent on "levels of exactness" instead of the "valid or bogus" state which is the Boolean methodology. During the 1960s, Dr. Lotfi Zadeh applied the fuzzy logic mentality firstly in his classes in the University of California at Berkeley. Fluffy hypothesis can be utilized as a method for speaking to dubiousness in building nonlinear associations with heuristic data. The hypothesis essentially works with the rationale that rather than an articulation being 0 or 1, its worth may have an esteem that can differ in this range [42].

Fuzzy approach aims to display the overall working rationale of the PC such that individuals can comprehend inside the system of rationale. A PC's rationale block gets outright contribution from the client and gives the yields TRUE or FALSE, which is equal to YES or NO outcomes. As per the fluffy rationale approach, the client's choice expresses that there are various conceivable outcomes between YES and NO. Utilizing fluffy rationale strategy, it is meant to demonstrate unsure circumstances and inappropriately characterized or complex frameworks [43].

The engineering of fuzzy logic framework comprises three principle parts as shown in the accompanying figure. Initially, it changes over framework contributions to the fluffy sets gave in the fuzzification module. The standards area depicts the circumstances that decide the yields of the framework's fluffy rationale approach. These circumstances show which articulation should be yielded against the changing info articulations of the framework [44]. At long last, the defuzzification module changes over the fluffy set produced by the surmising motor to net worth. Along these lines, the framework yields give diverse yield esteems as indicated by the guidelines in Figure 5 [45].

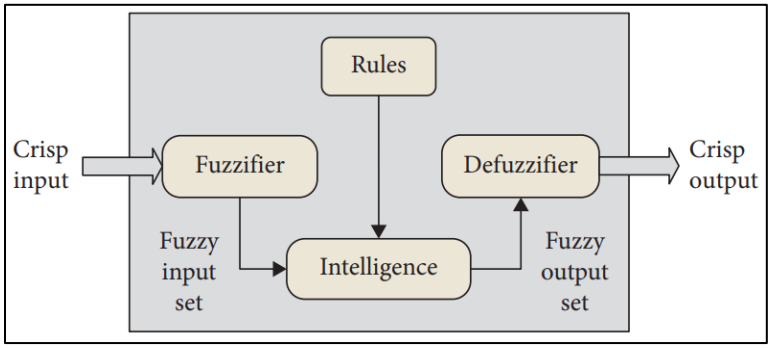

Figure 5. Fuzzy logic process [45].

6. FUZZY-BASED RULES

Based on the explanations in Section 4 (Data Mining Based Discovered Inferences), eight criteria – fuzzy-based rules – have been found to be "key value" and have been listed in the following.

Using any pre-determined model or method in the software development process: While developing a software project, it means benefiting from the models and methods that have been previously used, tested and operated by some employees.

Dependent on any software development method in the software development process: When developing the software project, the software development process under the main headings such as "heavy-weight" or "lightweight" is put into a specific plan. The use of one or more of the methods means benefiting from these.

Whether developed software makes it easier for customers to do their job or not: A software is developed based on the customer's wishes and requirements is expected to facilitate the solution of the existing problem. If it makes the work done earlier more complicated, the effectiveness of the software produced has to be questioned.

In the software development process, to do the process of erroneous, exceptional situation: software development process in the coding part of the program when writing, "error handling" and "exception handling" are expected to be used; otherwise, the developed program will produce erroneous results in the test section, meaning that it will collapse, and as a result, the desired product will not emerge.

Whether the developed software has good quality or not: The essential aim of the software development process is producing and developing qualified software, which is the sine qua non of the "Software Engineering".

In the software development process, pay attention to the concept of "re-usability": As in other projects rather than software, it is inevitable to use some modules, some functions or some structures previously created in developing software projects, and this is in fact a facilitating feature. The higher the concept of re-usability, the lower the cost of the software development process.

To develop reliable/secure code while developing software: In order to ensure reliable operation of the product that will be produced during the software development process and to give the desired results, it requires the reliable writing of the program of the software, which means that the codes of the program are developed according to the reliable structure.

Whether the developed software is understandable and usable or not: The developed software must be able to be used in a practical way by customers to produce solutions to existing problems; for this to be effective, the product in hand is understandable and therefore should be usable.

7. CONCLUSION

Based on the explanations in Section 4 (Data Mining Based Discovered Inferences), software development methodology brings with software quality, reusability, usability and understandability and modularity together. The attribute of "software quality" is the main aim of a software development method in software quality standards. In addition, the attribute of "reusability" provides software developers to use some methods and models or some classes and packages or some code parts which have been designed and developed before while developing software, shortly, it provides to re-use. Moreover, the attributes of "usability" and "understandability" go in the same direction together since if one uses software, this means s/he can understand that or if one understands software, this means s/he can use that. Also, these attributes show up in the result that a software development method requires systematic work based on a schedule. As a result of this, systematic work creates more usable and more understandable software by software developers. Furthermore, the attribute of "modularity" helps software developers to design and develop software class by class or package by package or module by module or part by part.

Based on the explanations in section 6 (fuzzy-based rules), the criteria for the measurement and evaluation of the software development, which are the result of our research and which are taken into consideration as a result of the analyses: software development process by relying on a specific method, taking care of reliable code writing, using tried and tested models, developing a product that is understandable and error-free by focusing on problem solving and, of course, as a result of which is an important point for software engineering. So, it shows that the gold of the concept has to be filled with concrete data.

In this context, the first goal in this study is to find the relationships between the software parameters, and software engineering terms and topics which affect objective criteria that are given value much more by solving these relationships. Afterwards, the ex-post aim is to minimize some bad things in software crisis in software engineering, such as, over-budget, over-time, low quality, not meeting requirements,

inefficiency according to the results of the survey. If the designed and developed software parameters, which have been formed and specified based on the software experts' thoughts, achieve the goal, maybe, the term software crisis in software engineering will disappear in the near future.

"You can't manage the process which you don't measure." This statement which is claimed to be said by Peter DRUCKER shows that the software development process has to be measured by clear and objective variables. Thus, some reliable data are specified and determined so that this software development process can be managed by benefiting from these trustworthy results. According to the results of these designed and developed software parameters, manpower, what is the main resource of software development process will have been used more effectively. And then, the benefits of the "Software Engineering" may be seen more tangible. As a natural result of this, software crisis may die out.

ACKNOWLEGEMENT

The ethics committee approval form and document is not required for this study, and the authors declare that there is no conflict of interest.

REFERENCES

[1] Ellis, B. (1968). *Basic concepts of measurement.* Cambridge, United Kingdom: Cambridge University Press.

[2] Pawson, R., & Tilley, N. (1994). What works in evaluation research? *The British Journal of Criminology, 34*(3), 291-306.

[3] Gallivan, M. J. (1998). The influence of system developers' creative style on their attitudes toward and assimilation of a software process innovation. Thirty-First Hawaii International Conference on System Sciences, 6-9 January, Kohala Coast, 435-444.

[4] Sawyer, S., & Guinan, P. J. (1998). Software development: Processes and performance. *IBM Systems Journal, 37*(4), 552-569.

[5] Hall, T., Wilson, D., Rainer, A., & Jagielska, D. (2007). The neglected technical skill? ACM SIGMIS CPR Conference on Computer Personnel Research: The Global Information Technology Workforce, 19-21 April, St. Louis Missouri, 196-202.

[6] Baggelaar, H. (2008). *Evaluating programmer performance visualizing the impact of programmers on project goals.* M.Sc Thesis, University of Amsterdam, Amsterdam.

[7] Lee, K., Joshi, K., & Kim, Y. (2008). Person-job fit as a moderator of the relationship between emotional intelligence and job performance. ACM SIGMIS CPR Conference on Computer Personnel Doctoral Consortium and Research, 3-5 April, Charlottesville VA, 70-75.

[8] Thing, C. (2008). The application of the function point analysis in software developers' performance evaluation. 4th International Conference on Wireless Communications, Networking and Mobile Computing, 12-17 October, China, 1-4.

[9] Zhang, S., Wang, Y., & Xiao, J. (2008). Mining individual performance indicators in collaborative development using software repositories. 15th Asia-Pacific Software Engineering Conference, 3-5 December, China, 247-254.

[10] Calikli, G., & Bener, A. (2010). Empirical analyses of the factors affecting confirmation bias and the effects of confirmation bias on software developer/tester performance. 6th International Conference on Predictive Models in Software Engineering, 12-13 September, Romania, no. 10.

[11] Chilton, M. A., Hardgrave, B. C., & Armstrong, D. J. (2010). Performance and strain levels of it workers engaged in rapidly changing environments: A person-job fit perspective. *ACM SIGMIS Database*, *41*(1), 8-35.

[12] Ramler, R., Klammer, C., & Natschläger, T. (2010). The usual suspects: A case study on delivered defects per developer. ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, 16-17 September, Italy, no. 48.

[13] Wang, Y., & Zhang, M. (2010). Penalty policies in professional software development practice: A multi-method field study. 32nd ACM/IEEE International Conference on Software Engineering, 1 May, Cape Town South Africa, 39-47.

[14] Balijepally, V., Nerur, S., & Mahapatra, R. (2012). Effect of task mental models on software developer's performance: An experimental investigation. 45th Hawaii International Conference on System Science, 4-7 January, Hawaii, 5442-5451.

[15] Duarte, C. B., Faria, J. P., & Raza, M. (2012). PSP PAIR: Automated personal software process performance analysis and improvement recommendation. Eighth International Conference on the Quality of Information and Communications Technology, 3-6 September, Portugal, 131-136.

[16] Ehrlich, K., & Cataldo, M. (2012). All-for-one and one-for-all?: A multi-level analysis of communication patterns and individual performance in geographically distributed software development. ACM 2012 Conference on Computer Supported Cooperative Work, 11-15 February, Washington, 945-954.

[17] Kelly, B., & Haddad, H. M. (2012). Metric techniques for maintenance programmers in a maintenance ticket environment. *Journal of Computing Sciences in Colleges*, *28*(2), 170-178.

[18] Schröter, A., Aranda, J., Damian, D., & Kwan, I. (2012). To talk or not to talk: Factors that influence communication around changesets. ACM 2012 Conference on Computer Supported Cooperative Work, 11-15 February, Washington, 1317-1326.

[19] Westermann, D. (2012). A generic methodology to derive domain-specific performance feedback for developers. 34th International Conference on Software Engineering, 2-9 June, Zurich, 1527-1530.

[20] Calikli, G., & Bener, A. (2013). An algorithmic approach to missing data problem in modeling human aspects in software development. 9th International Conference on Predictive Models in Software Engineering, 9 October, Baltimore Maryland, no. 10.

[21] Kumar, S. B., Sivaparthipan , C. B., Kalaikumaran, T., & Karthik, S. (2013). A case study of customer relationship management using data mining techniques. *International Journal of Technological Exploration and Learning*, *2*(6).

[22] Keleş A. E., & Kaya M. (2014). Analysis of factors affecting efficiency in wall construction using Apriori data mining method. XVI. Academic Informatics Conference, Mersin.

[23] Noughabi, E. A., Amin-Naseri, M. R., Albadvi, A., & Saeedi, M. (2016). Human resource performance evaluation from CRM perspective: a two-step association rule analysis. *International Journal of Business Performance Management, 17*(1), 89-102. Doi: 10.1504/IJBPM.2016.073340

[24] Xi, J., Zhao, Z., Li, W. ve Wang, Q. (2016). A traffic accident causation analysis method based on AHP-Apriori. *Procedia Engineering, 137*, 680-687. Doi: 10.1016/j.proeng.2016.01.305

[25] Chiang, W.-Y. (2018). Applying data mining for online CRM marketing strategy: An empirical case of coffee shop industry in Taiwan. British Food Journal, 120(3), 665-675. Doi:10.1108/BFJ-02-2017-0075

[26] Taş Y. (2018). *Association rules mining and an application.* M.Sc Thesis, Cumhuriyet University, Sivas.

[27] Gedleç Ş. (2019). *Multidimensional association rules analysis and business application.* M.Sc Thesis, Marmara University, Istanbul.

[28] Rokhayati, Y., Rusdi, U., Kurniawan, D., Janah, N., & Irawan, S. (2019). Analysis of SP students using AHP-Apriori combination. *Advances in Social Science, Education and Humanities Research, 354*, 186-191.

[29] Güler E. R. (2019). *Data mining application on energy consumption data.* M.Sc Thesis, Gazi University, Ankara.

[30] Gürcanok O. T. (2020). *A data science case study on e-commerce sales data.* M.Sc Thesis, Maltepe University, Istanbul.

[31] Mining frequent itemsets - apriori algorithm. (n.d.). Retrieved August 21, 2021, from http://software.ucv.ro/~cmihaescu/ro/teaching/AIR/docs/Lab8-Apriori.pdf

[32] Hegland, M. (2007). The Apriori algorithm – a tutorial. *Mathematics and Computation in Imaging Science and Information Processing*, 209-262.

[33] Al-Maolegi, M., & Arkok, B. (2014). An improved Apriori algorithm for association rules. *International Journal on Natural Language Computing (IJNLC), 3*(1), 21-29.

[34] Borgelt C., & Kruse R. (2002) *Induction of association rules: Apriori implementation.* In: Härdle W., Rönz B. (eds) Compstat. Physica, Heidelberg. https://doi.org/10.1007/978-3-642-57489-4_59

[35] Liu, H., & Wang, B. (2007). An association rule mining algorithm based on boolean matrix. *Data Science Journal, 6*(9), 559-565. Doi:10.2481/dsj.6.S559

[36] Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large database. *Proceedings of the 20th VLDB conference.*

[37] Tank, D. (2014). Improved Apriori algorithm for mining association rules. *International Journal on Natural Language Computing, 3*(1), 15-23. Doi:10.5815/ijitcs.2014.07.03

[38] Larose, D. T., & Larose, C. D. (2004). *Discovering knowledge in data: an introduction to data mining.* New Jersey: A John Wiley & Sons.

[39] Sun, D., Teng, S., Zhang, W., & Zhu, H. (2007). An algorithm to improve the effectiveness of Apriori. *6th IEEE International Conference on Cognitive Informatics*, 6-8 August, Lake Tahoe, 385-390.

[40] Browne, P., Morgan, S., Bahnisch, J., & Robertson, S. (2019). Discovering patterns of play in netball with network motifs and association rules. *International Journal of Computer Science in Sport, 18*(1), 64-79. DOI:10.2478/ijcss-2019-0004

[41] Han, J., & M. Kamber (2006). *Data Mining. Concepts and Techniques (2nd ed.).* United States: Morgan Kaufmann Publishers.

[42] Ross, T. J. (2016). *Fuzzy Logic with Engineering Applications (4th edition).* USA: Wiley.

[43] Carlsson, C., & Fuller, R. (2003). A fuzzy approach to real option valuation. *Fuzzy Sets and Systems, 139*(2) 297-312.

[44] Karataş, F., Koyuncu, İ., Tuna, M., ve Alçın, M. (2020). Bulanık mantık üyelik fonksiyonlarının fpga üzerinde gerçeklenmesi. *Bilgisayar Bilimleri ve Teknolojileri Dergisi, 1*(1), 1-9.

[45] Shen, Q., & Chouchoulas, A. (2002). A rough-fuzzy approach for generating classification rules. *Pattern Recognition, 35*(11), 2425–2438.