Applying Fuzzy Technique in Software Team Formation Based on Belbin Team Role

Mazni Omar, Bikhtiyar Hasan, Mazida Ahmad, Azman Yasin, Fauziah Baharom, Haslina Mohd, Norida Muhd Darus
School of Computing, Universiti Utara Malaysia, 06010 UUM Sintok, Kedah, Malaysia.
mazni@uum.edu.my

***Abstract*—Team roles play significant impact in determining the project success. In order to ensure that team can work together, it is essential to ensure that the team members are assigned to the right role with the right characteristics. One of the prevalent team roles is Belbin team role. The team role can guide project manager to form effective team. However, assigning the correct team role is very challenging. Thus, this study demonstrates the use of fuzzy technique to form a software team based on Belbin team role. By using this technique, it can help decision maker to form effective team that have balance characteristics. In order to validate the proposed technique, future works will be carried out by using empirical data in industrial setting.**

***Index Terms*—Belbin Team Role; Fuzzy Technique; Software Team; Team Formation.**

I. Introduction

Team roles an important role in determining the project success in software engineering field [1,2]. To ensure the optimal outcome of the project the team is working on, it is essential to ensure that the team comprises of the members with right roles and right characteristics [3]. Assigning correct role to correct team members can be challenging for project managers. The difficulty in executing this in correct way usually stems from the manager's inexperience in assigning the roles to team members [4,5]. According to Belbin [6], a good performance of a team is connected with balanced team in terms of team roles among the members.

The concept of team roles is not new. Early researchers such as Benne and Sheats in (1940-1950), identified some roles such as Harmoniser, Initiator-contributor and Energiser [7]. Nowadays one of the popular team roles is Belbin team role, the theory is centred on the team roles and how they should be matched in order to avoid conflicts and build sound teams that are optimally managed, team role defined as the predisposition to behave, contribute and interrelate with others in a specific way [8].

Several researchers have examined team formation and how groups progress into efficient teams through selection of group processes the accomplishment of assigned tasks [9] [10]. For example, in [10], team formation model highlights the four sequential phases to include: forming, storming, norming, and performing. In this context, the first phase of forming denotes a period at which members of a team determine their positions, procedures to follow, and the rules to be guiding the group, the next phase called storming commences when conflict occurs as a team member resists the influence of the group and rebels against accomplishment of the task. Norming phase commences as the group forms cohesiveness and commitment to its responsibilities, decides fresh ways to work together to accomplish the common goals and sets norms for suitable behaviors. Performing as the final phase occurs when the team achieves proficiency in working together to attain its goals and attains more flexibility in applying the procedures for working together. Since the success of the task depends on all the stages in the sequence, the first stage is the most important, as it is a precondition for all subsequent ones. For this reason, this study only focuses on the forming phase by focusing on how team members are assigned to a specific role based on specific characteristic using a specific technique.

II. Background of the Study

Forming effective team members is essential for many software organizations, especially those of small and medium size, as they operate with tight budgets and have fewer individuals to consider when forming teams for specific projects. On the other hand, in larger organizations, with many employees, it is much easier to ensure that the individuals possess diverse experience, which can be matched to the constraints and skill requirements of each project. Clearly, success or failure of software product mostly depends on the development team. Thus, when deciding on its composition, three major team formation methods— self-formation, random-formation, and instructor-formation—can be adopted [11, 12].

However, in industrial settings, team formation is typically responsibility of the manager, who uses his/her experience and judgment when determining team composition, while this is a prevalent approach, empirical evidence has shown that it does not always yield optimal results [13] . Failures typically occur when time and cost are the main constraints and the available employee pool comprises of individuals with mixed types of expertise [13, 14, 15]. Thus, in such circumstances, a systematic mechanism must be adopted and utilized in ensuring that the optimal team selection is consistently made.

Many researchers have argued that for the Team Roles formation to be achieved in software development, nine roles have to be taken into consideration, including Shaper, Plant, Resource Investigator, Evaluation Monitor and Coordinator [16, 17, 18]. Others are Complete Finisher, Team Worker, Specialist and Implementer [19, 17, 18, 20]. However, previous studies have revealed that the Shaper and Plant Team Roles in software engineering [19, 17, 18, 20].

In this study, the Shaper and Plant in Belbin Team Roles were chosen based on the research and experiments that different researchers made in the field of software engineering with Belbin Team Roles. The roles of Shaper and Plant in Belbin Team Roles are significant to the software engineering team; hence the Shaper role is used for the leader while the Plant role is used for other member [21, 22, 23].

During team formation, there are several techniques that can be considered to form teams, for instance Analytic Hierarchy Process, Case-based Reasoning, Multi-Attribute Utility Theory and Fuzzy technique [9]. However, several researchers like [10, 24, 25] have provided other various techniques to team formation, such as the use of multi-dimensional trust revealed that the reliable evaluation has a considerable value in solving the problem of team formation. Nevertheless, [26] opined that such techniques still require some other qualities, like the potential team members' proficiency, characteristics of project, and the team members' tasks.

Fuzzy technique has gained popularity, because it allows analysing imprecise data and classifying selected criteria. Initial evaluation of this technique showed that it can indicate whether every team possesses equal distribution of the key criteria. By incorporating this technique into the chosen team formation method, each team can enhance its chances of performing effectively. In particular, this technique can facilitate decision makers when forming highly productive project teams. However, at present, studies that demonstrate the applicability of this technique in forming software team members are limited [3]. In particular, there is a significant gap in the knowledge on the factors determining a balance of team members based on Belbin roles. Thus, this study will fill this gap by providing a team formation method based on Belbin team roles by using a fuzzy technique.

III. TEAM FORMATION TECHNIQUES

Different techniques have been developed for team formation. Selecting a proper technique for team formation is based on the kinds of issues being dealt with, the attributes of the method of team formation, and the aims of the teams [27]. According to [28], the common team formation techniques include fuzzy, data development analysis (DEA), ELimination and Choice Expressing REality (ELECTRE), goal programing, and Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). Within the remainder of this section, the authors give an overview of each of these techniques.

A. Fuzzy

Fuzzy is an extension of classical set theory, which allows for solving many problems related to dealing with imprecise and uncertain data. It has many advantages, including considering insufficient information and the evolution of available knowledge, allowing imprecise input, and allows for a few rules to encompass problems with great complexity. Some disadvantages of this technique include the fact that fuzzy systems can sometimes be difficult to develop and, in many cases, they can require numerous simulations before being suitable for use in the real world [29].

Fuzzy is an established technique that has been used in engineering, economic, environmental, social, medical, and management problems. Many of these types of problems take advantage of the availability of imprecise input. These types of applications favour a method that embraces vagueness and can be tested numerous times before real-world application [30].

B. DEA

DEA uses a linear-programming technique to measure the relative efficiencies of alternatives [31]. It rates the efficiencies of such alternatives against each other, with the most efficient alternative receiving a rating of 1.0, with all other alternatives receiving ratings of a fraction of 1.0. DEA has several advantages, including being capable of handling multiple inputs and outputs, efficiency can be analysed and quantified, and it can uncover relationships that may be in hidden with other techniques, an important disadvantage is that it does not deal with imprecise data and assumes that all input and output data are exactly known. In real-world situations, however, this assumption may not always be true [32]. The results can be sensitive, depending on the inputs and outputs. DEA is used wherever efficiencies must be compared. This technique is commonly used in economic, medical, utilities, road safety, agriculture, retail, and business problems. These categories are especially useful because they contain precise data that could be utilized for input, which bypasses one of the method's major deficiencies [30].

C. ELECTRE

ELECTRE, along with its many iterations, is an outranking technique based on concordance analysis. Its major advantage is that it takes into account uncertainty and vagueness. One disadvantage is that its process and outcomes can be difficult to explain in layman's terms. Further, due to the way preferences are incorporated, the lowest performances under certain criteria are not displayed, the outranking method causes the strengths and weaknesses of the alternatives to not be directly identified, and results and impacts are not verified [33]. ELECTRE has been used in energy, economics, environmental, water management, and transportation problems. Like other methods, it also considers vagueness and uncertainty, which many of the previously-mentioned applications appear to need [30].

D. Goal Programming

Goal programming is a pragmatic programming technique that is able to choose from an infinite number of alternatives. One of its advantages is that it can handle large-scale problems. Its ability to produce infinite alternatives provides a significant advantage over some methods, depending on the situation. A major disadvantage, however, is its inability to weight coefficients. Goal programming has been applied in production planning, scheduling, healthcare, portfolio selection, distribution system design, energy planning, water reservoir management, timber harvest scheduling, and wildlife management problems. Many of these applications have been used in combination with other methods to accommodate proper weighting. Finally, by doing so, it eliminates one of its weaknesses while still being able to choose from infinite alternatives [30].

E. TOPSIS

TOPSIS is an approach to identify an alternative which is closest to the ideal solution and farthest from the negative ideal solution in multi-dimensional computing [34]. It has numerous advantages, including being easy to use and programmable, and the number of steps remains the same regardless of the number of attributes [35]. A disadvantage is that its use of Euclidean Distance does not consider the correlation of attributes. It is difficult to weight attributes and keep consistency of judgment, especially with additional attributes. TOPSIS has been used in supply chain management and logistics, design, engineering and manufacturing systems, business and marketing management, environmental management, human resources management, and water resources management. This is another method where its ease of use has kept its application popular. Its simplicity and ability to maintain the same number of steps regardless of problem size has allowed it to be utilized quickly to review other methods or to stand on its own as a decision-making tool [30].

IV. PROPOSED METHOD

In this paper, software team formation method based on Belbin team role using fuzzy technique was used to determine whether each team has equal team role. Applying fuzzy technique involves three main steps which are; 1) Fuzzification, 2) Fuzzy Inference and 3) Defuzzification [36]. As discussed in Section 2, only two roles are significant in SE team based on Belbin team roles, which are Shaper (Sh) and Plant (Pl) roles. Thus, the two roles are the input parameters. The two input parameters were used to monitor the score of the roles, whereas the output representing the optimal weight for team members. Figure 1 shows the framework for this research.

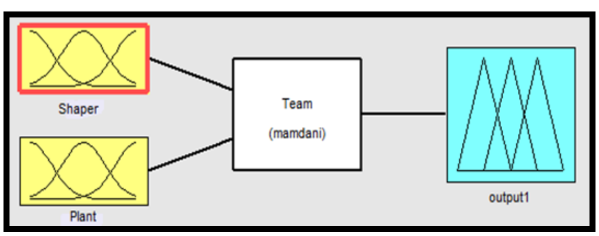

Figure 1: Main research framework

A. Membership Function for Input Parameters

The input parameters represented the roles for team members (Shaper and Plant). These two roles were determining the membership functions. Each role was classified into three levels, which were ‘Low’, ‘Normal’ and ‘High’. These three level classified adapted from previous work such as [37]. Figure 2 shows the representation of membership function for roles.

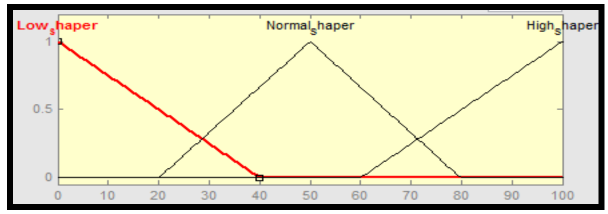

Figure 2: Membership function

Figure 2 represents two axes which are (X and Y), X-axis represent the inputs value, since the value between (0-100), and Y-axis represent the fuzzy number which lay between (0-1).

B. Fuzzy Rule-Based Construction

During this stage, rules construction was developed by calculating the mean value of the roles for each member in the team. A fuzzy rule was constructed based on three levels, which were, ‘Low’, ‘Normal’ and ‘High’. Figure 3 shows the constructed rules calculated according to member’s roles. In this study, 14 rules were constructed. The rules were constructed according to the roles value after classified them to the levels (High, Normal, and Low). In addition, the levels of the membership for each member determine the optimal weight for the members in the team. The rules were processed using Mamdani-style inference in Matlab toolbox in order to get crisp output. Mamdani was used because this method is widely accepted for capturing expert knowledge, and it allows us to describe the expertise in more intuitive, more human-like manner.

1. If (input1 is Low_Shaper) and (input2 is Low_Plant) then (output1 is Low_Membership) (1)
2. If (input1 is Low_Shaper) and (input2 is Normal_Plant) then (output1 is Low_Membership) (1)
3. If (input1 is Low_Shaper) and (input2 is High_Plant) then (output1 is Normal_Membership) (1)
4. If (input1 is Normal_Shaper) and (input2 is Low_Plant) then (output1 is Low_Membership) (1)
5. If (input1 is Normal_Shaper) and (input2 is Normal_Plant) then (output1 is Normal_Membership) (1)
6. If (input1 is Normal_Shaper) and (input2 is High_Plant) then (output1 is High_Membership) (1)
7. If (input1 is High_Shaper) and (input2 is Low_Plant) then (output1 is Normal_Membership) (1)
8. If (input1 is High_Shaper) and (input2 is Normal_Plant) then (output1 is High_Membership) (1)
9. If (input1 is High_Shaper) and (input2 is High_Plant) then (output1 is High_Membership) (1)
10. If (input1 is Normal_Shaper) or (input2 is Normal_Plant) then (output1 is Normal_Membership) (1)
11. If (input1 is Normal_Shaper) or (input2 is High_Plant) then (output1 is High_Membership) (1)
12. If (input1 is High_Shaper) or (input2 is Normal_Plant) then (output1 is High_Membership) (1)
13. If (input1 is High_Shaper) or (input2 is High_Plant) then (output1 is High_Membership) (1)
14. If (input1 is High_Shaper) or (input2 is not High_Plant) then (output1 is High_Membership) (1)

Figure 3: The Constructed Rule

C. Defuzzification

The final part of applying fuzzy is converts the fuzzy result into crisp value, crisp values refer to real number. In this study, the real number refers to the inputs roles value. This step focuses on the membership functions used. The output will be converted to crisp values in accordance with the three triangular membership functions, “low”, “normal” and “high” in Figure 4.

D. Distributing members among the teams

After determining the optimal weight for each member according to the roles value by using fuzzy technique, the members will distribute among the teams randomly, the members in each of the teams would be added together and compared the summations to each another. The process of distributing the leader and members among the teams is illustrated in Figure 5. In this figure, L represents the leader, and M represents the other team members.

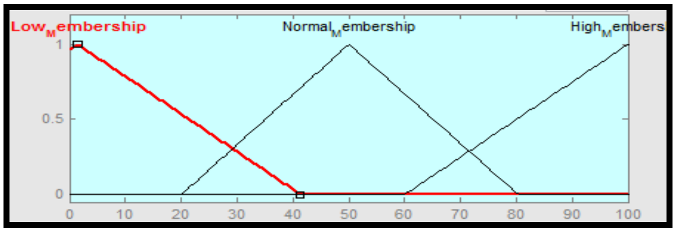

Figure 4: Membership functions for output

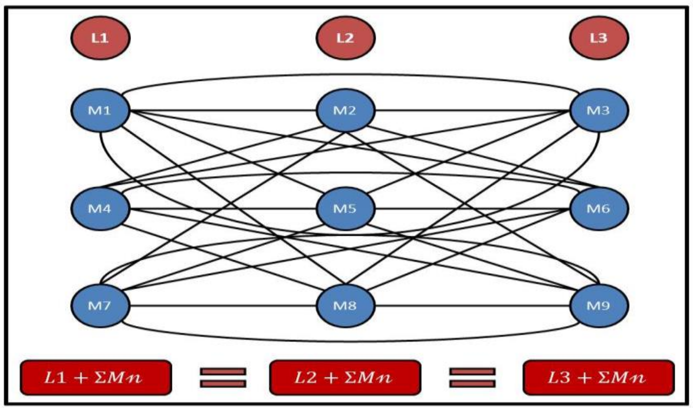

Figure 5: Distributing member's process

If the total weight for the teams equal, the process moves to the next step. Otherwise, the process going to change between the members until having a balance weight for all teams, whereas balance refer to, that all teams have optimal equal weight with existing both roles (Shaper and Plant).

V. Conclusion and Future Works

The main aim of this study is to apply fuzzy technique to form software team based on Belbin team role. The technique is used to ensure that the team member has an equal team role. When the team has an equal team role, the team will exhibit balance characteristics that can help to achieve successful teamwork. Future works will focus on the validation of the proposed method by using empirical data from industrial setting.

Acknowledgment

The authors wish to thank Universiti Utara Malaysia (UUM) for funding this study under High Impact Group Research Grant Scheme (PBIT), S/O project: ―12867.

References

[1] P. Ralph and P. Kelly, "The dimensions of software engineering success.," *Icse*, pp. 24–35, 2014.

[2] C. Ebert and P. De Neve, "Surviving global software development," *IEEE Softw.*, vol. 18, no. April, 2001.

[3] S. L. Syed-Abdullah, M. Omar, and M. F. I. M. Idris, "Team achievements equality using fuzzy rule-based technique," *World Appl. Sci. J.*, vol. 15, no. 3, pp. 359–363, 2011.

[4] B. Senior, "Construct Validity and Team Building," *ISBN*, no. August, pp. 1–39, 2005.

[5] S. E. Humphrey, F. P. Morgeson, and M. J. Mannor, "Developing a theory of the strategic core of teams: a role composition model of team performance.," *J. Appl. Psychol.*, vol. 94, no. 1, pp. 48–61, 2009.

[6] Belbin, "A comprehensive review of Belbin team roles," *Belbin UK*, pp. 1–26, 2014.

[7] D. Partington and H. Harris, "Team role balance and team performance: an empirical study," *Emerald*, vol. 18, no. 8, pp. 694–705, 1999.

[8] Belbin, "Frequently asked questions," *Belbin UK*, pp. 1–2, 2012.

[9] S. W. J. Kozlowski, & B. S. Bell,"Work Groups and Teams in Organizations Work Groups and Teams in Organizations," *ILR Collection*, pp. 1–70, 2003.

[10] B. W. Tuckman, "Developmental sequence in small groups," *Psychological Bulletin*, vol. 63 no. 6, pp. 384–399, 1965.

[11] R. van Cann, S. Jansen, and S. Brinkkemper, "Optimal Team Composition in Distributed Software Development," *Collab. Outsourcing A Journey to Qual.*, no. 2005, p. 160, 2012.

[12] S. S. Hamilton, "Optimizing team selection for educational group projects," *United States Mil. Acad. West Point*, 2010.

[13] S. Kr.Misra and A. Ray, "Software developer selection: A holistic approach for an eclectic decision," *Int. J. Comput. Appl.*, vol. 47, no. 1, pp. 12–18, 2012.

[14] M. R. J. Qureshi, S. A. Alshamat, and F. Sabir, "Significance of the teamwork in agile software engineering," *Lahore Leads Univ.*, vol. 26, no. 1, pp. 117–120, 2014.

[15] Z. Ezziane, M. Maruthappu, L. Gawn, E. a. Thompson, T. Athanasiou, and O. J. Warren, "Building effective clinical teams in healthcare," *J. Health Organ. Manag.*, vol. 26, pp. 428–436, 2012.

[16] P. K. Schoenhoff, "Belbin's company worker, the self-perception inventory, and their application to software engineering teams by Belbin's company worker," State University, 2001.

[17] G. Beranek, W. Zuser, and T. Grechenig, "Functional group roles in software engineering teams," *ACM SIGSOFT Softw. Eng. Notes*, vol. 30, no. 4, p. 1, 2005.

[18] A. Ounnas, H. Davis, and D. Millard, "A Framework for Semantic Group Formation in Education.," *Educ. Technol. Soc.*, vol. 12, no. 4, pp. 43–55, 2009.

[19] S. Fatahi and a R. Lorestani, "Design and Implementation of the Expert System for Balancing Team Formation on the Basis of Belbin Team Role," *Engineering*, vol. I, 2010.

[20] S. M. Henry and K. Todd Stevens, "Using Belbin's leadership role to improve team effectiveness: An empirical investigation," *J. Syst. Softw.*, vol. 44, no. 3, pp. 241–250, 1999.

[21] M. Rajendran, "Analysis of team effectiveness in software development teams working on hardware and software environments using Belbin Self-perception Inventory," *Emerald*, 2005.

[22] A. Ounnas, H. Davis, and D. Millard, "A Framework for Semantic Group Formation," *2008 Eighth IEEE Int. Conf. Adv. Learn. Technol.*, pp. 6–10, 2008.

[23] J. Wasiak and L. Newnes, "Guiding team selection and the use of the Belbin approach," *DS 48 Proceeding*, pp. 1113–1120, 2008.

[24] T.-L. (Bill) Tseng, C.-C. Huang, H.-W. Chu, and R. R. Gung, "Novel approach to multi-functional project team formation," *Int. J. Proj. Manag.*, vol. 22, no. 2, pp. 147–159, 2004.

[25] T. Venkatamuni and A. Rao, "Reduction of product development time by team formation method in lean manufacturing," *Indian J. Sci. Technol.*, vol. 3, no. 5, pp. 578–582, 2010.

[26] R. Van Cann, S. Jansen, and S. Brinkkemper, "Team composition in distributed software development," *Univ. Utr.*, no. 2005, 2012.

[27] M. Saeed and A. Trab. *Software Engineering : Testing Real-Time Embedded Systems Using Timed Automata Based Approaches*. *Brunel University*, 2012.

[28] K. Mukherjee, and A. Bera, "Application of goal programming in project selection decision—A case study from the Indian coal mining industry," *European Journal of Operational Research*, vol. 82 no.1, pp. 18–25, 1995.

[29] J.-F. Balmat, F. Lafont, R. Maifret, R., and N. Pessel, "A decision-making system to maritime risk assessment," *Ocean Engineering*, vol. 38 no. 1, pp. 171–176, 2011.

[30] M. Velasquez and P. T. Hester, "An Analysis of Multi-Criteria Decision Making Methods," *Int. J. Oper. Res.*, vol. 10, no. 2, pp. 56–66, 2013.

[31] E. Thanassoulis, M.Kortelainen, and R. Allen, "Improving envelopment in data envelopment analysis under variable returns to scale," *European*

Journal of Operational Research, vol. 218 no. 1, pp. 175–185, 2012.

[32] Y.-M., Wang, R., Greatbanks, and, J.-B. Yang, "Interval efficiency assessment using data envelopment analysis," *Fuzzy Sets and Systems*, vol. 153 no. 3, pp. 347–370, 2005.

[33] P. Konidari and D. Mavrakis, "A multi-criteria evaluation method for climate change mitigation policy instruments," *Energy Policy*, vol. 35 no. 12, pp. pp. 6235–6257, 2007

[34] X.-S, Qin, G. H. Huang, A. Chakma, X. H., Nie, and Q. G. Lin, "A MCDM-based expert system for climate-change impact assessment and adaptation planning–a case study for the Georgia Basin, Canada," *Expert Systems with Applications*, vol. 34 no 3, pp. 2164–2179, 2008.

[35] Y. T. İç, "An experimental design approach using TOPSIS method for the selection of computer-integrated manufacturing technologies, "*Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 245–256, 2012.

[36] V. Varun, B. Govindarajan, and S. Nayak, "Speed Control of Induction Motor using Fuzzy Logic approach," National Institute of Technology - Rourkela, 2012.

[37] P. Rodjito, "Position tracking and motion prediction using Fuzzy Logic," Colby College, 2006.