International Symposium on Mathematical Sciences & Computing Research (iSMSC) 2015 (iSMSC' 15)

Making Programmer Suitable for Team-Leader: Software Team Composition Based on Personality Types

[1]Abdul Rehman Gilal, [2]Jafreezal Jaafar, [3]Shuib Basri,
[4]Mazni Omar, [5]Muhammad Zahid Tunio
[1,2,3]Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Malaysia
[1,]Sukkur Institute of Business Administration (IBA), Pakistan, [4]School of Computing, Universiti Utara Malaysia
[5]Department of Computer System Engineering, Dawood University of Engineering & Technology Pakistan
[1]a-rehman@iba-suk.edu.pk , [2]jafreez@petronas.com.my , [3]shuib_basri@petronas.com.my ,
[4]mazni@uum.edu.my, [5]zahid.tunio@duet.edu.pk

***Abstract*— The profuse use of software has turned the world into global village where everything is accessible at finger tips. The past studies have confirmed the rapid increase in the demand of software whereas its quality supply has drastically decreased to 6%. As high demand and low supply normally generate numerous problems, many researchers, therefore, have raised their concern to develop software affordable, less time consuming and feasible to achieve organisational ends. The findings of the past research studies have determined the fact that besides technical skills, human resources (i.e., personality type for team composition) is of pivotal importance for developing software which has not been seriously addressed. This study has tried to address this prevailing problem by focusing on patterns of personality types of programmer role monitored with team-leader. Additionally, to draw the attention of practitioners, the results are validated with several classification techniques and results appeared with high accuracy. The study has implications on both software developers and researchers having their interest in role of team composition in software development.**

Keywords- Human aspects; Personality types; team leader; programmer; gender; software development; team composition; rule-based; rough sets.

I. INTRODUCTION

The successful and progressing orgnisations of the world have attributed quality software as one of the key sources to progress by leaps and bound. By now, there is fueling strong demand for software for different purposes. But, it is regretfully added that the supply of quality software is decreasing drastically from software development workplaces. For instance, the past studies by Wysocki [1], Standish-Group [2], Glass [3], and Keil et al. [4] have shown their concern over the ratio of IT development projects that have continuously failed to achieve their desired ends. In the same vein, the Standish Groups [5] carried out longitudinal research study to determine the fall of successful software development in which several software development workplaces were surveyed. The findings of this study revealed that just 6% software serve the organizational purposes whereas 42% software become ineffective during developing process and majority of 52% software are challenged for their irrelevancy, time consumption, and expensive nature.

There could be several other reasons causing software development failure, but the human aspect is considered the most substantial reasons of all. Because, it is believed [6] that the development of software is not only the technical fragment of work but it also requires the human resources and their technical expertise to develop the software. Hence, the software development can be successful if it integrates team members with technical and social skills. Similarly, Capretz & Ahmed [7] also asserted that the software development team becomes ideal if its members are equipped with technical and social skills. Additionally, isolation of skills either social or technical will never ensure the quality team development for software [8].

Keeping in view the grave importance of team composition for developing successful software, the current study suggests the rules for effective software team composition in order to reduce ambiguities. For this purpose, Myers-Briggs Type Indicator (MBTI) personality type is used to assess personality types amongst software team members. The MBTI personality types is chosen because it is widely used and accepted amongst researchers in the field of software engineering. In addition, the results emanating from this research study only benefits the teams comprising of four to six members who develop a small-scale software project size. Finally, this study only focuses on the role of "programmer" with "team-leader" and "other programmers in the teams" which are the most important contributors in software development teams.

II. RELATED WORK

One cannot claim that the success of software could only be achieved by set of technical principles. In fact, yet there is no specific way of constructing software since each and every application has its explicit demands. It is, therefore, not an easy task to identify and encounter all possible ambiguities that could occur during the process of software development.

The past research on software development also suggests that software development is a complex sociotechnical activity [9]–[11]. The vast research stream mainly asserts that software development is not only to develop the technical product but it also includes the social process which involves several actors with different backgrounds working together to achieve the same goal [12]–[14]. In the same vein, Curtis et al., [15] maintained that the software development process should be treated as communication, negotiation, and learning activity. Therefore, it is the main

objective of this study to see the effective equations of team members' composition with personality types which can ensure the learning, communication, and negotiation processes effectively. Moreover, as software development involves people in process, therefore issues related with human can never be ignored in this regard.

Numbers of research studies have been conducted in the past for finding the relationship between social and technical skills. In them, software development team composition with personality type of the members is also very impacting area. But unfortunately, unlike social science, the research on software development team composition has not reached at the maturity level [16], [17]. Moreover, past studies have left and created several ambiguities for practitioners to use the suitable team composition models for building ideal teams. For instance, Gorla and Lam [18] have recommended the E (extrovert) personality trait for programmer whereas Ahmed and Capretz [19] suggested the I (introvert) personality trait. For the same issue, Cruz and da Silva [20] and MacDonell [21] claim that the validity of models for effective team composition of software development is still dubious.

Moreover, to cope this prevailing problem in software team composition, Gilal et al.,[17] found the relationship between personality types together with gender for software development teams' composition. This study investigated the performance variations among software development team members caused by genders' personality types. For instance, the male-dominated teams created the reasons for the females for being ineffective in teams if the personality type of female was with "E" trait. Furthermore, the study revealed that the female-leader were more convenient with only female or majority-female (i.e., having female in majority) groups. On the contrary, male-leaders were adjustable with all kind of team compositions. Critically, this study was just based on tabulated calculation and could not give any statistical or predictive evidences. However, this study also opened new vista for further research on the gender with personality types to reach appropriate conclusions. Moreover, the study by Mazni et al., [22] also studied the relationship with project success and personality types. This study focused the homogeneous and heterogeneous personality based teams' performance and its findings suggested that heterogeneous teams were more effective than the homogeneous teams.

Moreover, in order to measure the personality type of an individual the Myers-Briggs Type Indicator (MBTI) has been widely used in social sciences. It is also reliable and valid tool among software engineering researchers [17], [23]–[26]. For instance, Furnham [27] used MBTI in training and consultancy areas, and Gorla & Lam [18], Bradley & Hebert [28], and Cunha & Greathead [29] used MBTI in the area of Information Systems and Software Engineering.

An individual's personality type in MBTI is assessed on four dimensions: social interaction (extroversion (E) and introversion (I)), decision making (thinking (T) and feeling (F)), information gathering (sensing (S) and intuition (N)), and dealing with the external world (judging (J) and perceiving (P)) [7]. The MBTI test allows individual personality type preferences to be classified according to the 16 types that results as a combination of four dimensional pairs, which are Introversion (I) and Extroversion (E); Thinking (T) and Feeling (F); Sensing (S) and Intuitive (N) ; and Judging (J) and Perceiving (P). The 16 possible personality combinations are formed from four dimensions shown in the following Table 1:

Table T1: The 16 MBTI Personality Types

ISTJ (1)	ISFJ (2)	INFJ (3)	INTJ (4)
ISTP (5)	ISFP (6)	INFP (7)	INTP (8)
ESTP (9)	ESFP (10)	ENFP (11)	ENTP (12)
ESTJ (13)	ESFJ (14)	ENFJ (15)	ENTJ (16)

III. METHODOLOGY

The main purpose of this research paper was to find the suitable personality types of programmer with the team-leader by gender classification in software development teams. In other words, it could also be said that this research paper tried to find the effective personality types' equations between team-leader and programmer in software development by focusing gender as impacting variable on programmer role. Keeping in view, member-role, gender, and personality type of individual were considered as the predictor variables of this research. In addition, the impact of these predictor variables was measured on an outcome variable i.e., team-performance. The following table T2 shows the variables and possible inputs used in this study.

Table T2: List of Variables used in this research.

Variable	Input
Predictor	
1. Member role	team-leader
	Programmer
2. Personality types	**16 types of MBTI** 1 = ISTJ; 2 = ISFJ 3 = INFJ; 4 = INTJ 5 = ISTP; 6 = ISFP 7 = INFP; 8 = INTP 9 = ESTP; 10 = ESFP 11 = ENFP; 12 = ENTP 13 = ESTJ; 14 = ESFJ 15 = ENFJ; 16 = ENTJ (see related work for explanation)
3. Gender	1=Male
	2=Female
Outcome	
1. Team performance	Quality(0)=Ineffective
	Quality(1)=Effective

In order to find the possible combination of personality types between team-leader and programmer, this study applied predictive approach (i.e., rough sets) on the data collected from Universiti Utara Malaysia (UUM). Prior to data collection, students were asked to develop the software

development projects in teams/groups for a semester. Furthermore, the progress of groups were observed and monitored to check the quality points of the projects. In the end, those all projects were submitted and were evaluated by requirement engineer to see their quality of development in the light of projects' requirement. Agile development methodology were followed by students during development of projects. Because, small/medium teams were composed with 4-6 members for short period of time. As, it is the attractiveness of the agile development methodology that it is suitable for small/medium teams for small development projects. Moreover, students were asked to submit the projects with line of code, and documentation of the projects was not required.

Overall, total 184 students participated in the experiments of project development in 45 teams. In which, 17 projects were declared as effective-projects and 28 were said ineffective-projects by requirement engineer (i.e., based on the requirements). Importantly, this research only focused the successful software teams for finding efficient personality types' equations for team-leader and programmer. Therefore, only 68 members' personality type counted from those 17 effective-projects. Moreover, 17 team-leaders were counted (i.e., each team one leader) and 51 programmers appeared in those shortlisted projects. It should also be noted here that this paper employed the term "effective" to refer results obtained from those projects which met the requirements and other way around for term "ineffective".

Apart from statistical information, predictive approach of rough sets was applied on the overall data by following steps:

1. Data was split into two major and basic sets: training and testing, with 70% and 30% standard ratio [30]–[32].
2. Genetic Algorithm (GA) was applied on training set by using ROSETTA toolkit for finding possible equations (or rules) of personalities.
3. Standard Voting (SV), Voting with Object tracking, and Naïve Bayes classification techniques were then applied on the results emanated from GA algorithm (or 2nd step) for finding prediction accuracy.
4. Filtering only effective results for suggesting possible effective personality types for team-leader and programmer.

IV. RESULTS AND DISCUSSION

This section thoroughly discusses the possible equations of programmers' personality type with the team-leader personality types. In order to extract the reliable results, GA was not the only algorithm which was applied but Johnson's Algorithm (JA) was also applied to extract the trends from training set. Eventually, the GA results emerged as efficient with the objective of the research. Moreover, the application GA algorithm extracted 60 rules from training data set. Among these 60 rules, 22 rules were shortlisted for further discussion and the rest 38 rules were dropped for showing ineffective endings in results. The following table T3 is showing the efficient rules for team-leader and programmer composition in team.

Table T3. Rules for team-leader and programmer personality types' in team composition

Rule No	Rules/Equations
1	Leader_p(16) AND Progr_p(16) => quality(0) OR quality(1)
2	Leader_p(15) AND Progr_p(13) => quality(1)
3	Leader_p(15) AND Progr_p(15) => quality(1)
4	Leader_P(16) AND Progr_p(2) => quality(1)
5	Leader_p(15) AND Progr_p(5) => quality(1)
6	Leader_p(3) AND Progr_p(4) => quality(1)
7	Leader_p(3) AND Progr_p(16) => quality(1)
8	Leader_p(3) AND Progr_p(15) => quality(1)
9	Leader_p(1) AND Progr_p(16) => quality(1)
10	Leader_p(7) AND Progr_p(1) => quality(1)
11	Leader_p(15) AND Progr_p(11) => quality(1)
12	Leader_p(2) AND Progr_p(4) => quality(1)
13	Leader_p(2) AND Progr_p(6) => quality(1)
14	Leader_p(13) AND Progr_p(11) => quality(1)
15	Leader_p(13) AND Progr_p(15) => quality(1)
16	Leader_p(3) AND Progr_p(1) => quality(1)
17	Leader_p(16) AND Progr_p(13) => quality(1)
18	Leader_p(16) AND (Progr_p(15) AND gender(2))=> quality(1)OR quality(0)
19	Leader_p(16) AND (Progr_p(15) AND gender(1))=> quality(1)
20	Leader_p(7) AND (Progr_p(15) AND gender(1)) => quality(1)
21	Leader_p(14) AND (Progr_p(14) AND gender(1))=> quality(1)
22	Leader_p(13) AND (Progr_p(14) AND gender(2))=> quality(1)

Prior to the discussion of the results, it is important to discuss the structure of the rules presented in table T3 above. First of all, the rules (or personality type equations) are divided into two parts: condition and outcome. For instance, the side before "=>" sign is highlighting the "IF" part of the equation and the part after "=>" is "THEN" part. It is, therefore, the IF part could also be called Left Hand Side (LHS) and the THEN part as Right Hand Side (RHS). Keeping in view, Leader_p() denotes that the leader personality type comes among 16 MBTI personalities. In the same vein, Progr_p() is representing the personality type of programmer with particular number of MBTI. Moreover, the terms gender(1) and gender(2) refer to the male and the female gender respectively. Finally, quality(1) denotes that the equation is evidenced as an efficient from dataset after applying predictive techniques.

Basically, table T3, given above, possessed two type of rules for the formation of programmer with team-leader. In which, the first type of rules did not discuss the gender classification of programmer, but only team-leader and programmer personality types. Whereas, the second type of rules were composed of team-leader, programmer personality types, and gender of programmer. Technically,

both types of rules were important to consider for programmer personality types' composition with team-leader since the dataset was small and it would not have covered the overall range of personality types' possibilities. Therefore, this research measured that the composition of programmer with team-leader is generic for both genders if the gender is not mentioned. It means that the equation for programmer without gender classification can be used to compose team-leader with any type of gender programmer.

Based on the results of above given table T3, the first 17 rules were sorted as generic rules for both gender programmer (i.e., male or female) with team-leader. In which, team-leader with personality type number 13, 15, and 16 (or EJ (Extrovert and Judging) combo personality type leader) were mostly seemed adjusted with E trait personality holder programmers. In other words, E trait programmers were more convenient with the leaders of E trait. For instance, personality type number 13, 15, and 16 appeared as team-leader for 9 programmers (see table T3 rule number 1, 2, 3, 4, 5, 11, 14, 15, and 17 above). In which, majority of programmers were E trait (see table T3 rule number 1, 2, 3, 11, 14, 15, and 17 above) and two types of rules remained efficient with I (Introvert) trait but with S (Sensing) trait. In the same vein, team-leader with IJ (introvert and Judging) combo was found moderated with I and E traits programmers. As rule number 6, 7, 8, 9, and 16 from table T3 given above showed that programmer with I and J traits can also work well with team-leaders possessing I and J traits (see rule number 6 and 16 from table T3 above). Moreover, the rule number 10, 12, and 13 showed programmers and leaders with I trait personality respectively and their results appeared quite efficient for both of them could work together with an ease. Therefore, one can claim that I trait programmer can work happily with I trait team-leaders provided S trait is in the personality types of programmer or the team has equal number of members from both I and E traits.

On another hand, 5 rules were combination of team-leader + (programmer+gender) personality types. These rules were richer because they gave the explanation of programmers' gender with team-leader personality types. Keeping in this view, rule number 18 and 19 discovering number 18 clearly says that if the personality type is ENFJ and the gender is female with ENTJ team-leader then they may create 60% efficient outcomes. But, the guaranteed efficient outcomes can be ensured on a condition when male programmer work together having similar personality types (see rule number 19 above). Similarly, ESFJ (personality type number 14) is suitable for male programmer for the same type of personality team-leader. But, it does not assure at any level that the female programmer with ESFJ personality type will create equation with ESFJ team-leader or not. However, female programmer with ESFJ can be guaranteed efficient if team-leader's personality T trait is exchanged with F trait. It means that ESFJ female programmer can produced efficient results if the team-leader is ESTF. Lastly, N and F traits were found fruitful only if there was male gender and the leader possessing I and P traits.

V. VALIDATION OF THE RULES

After extracting the rules to determine the effective programmer for team from the data by using GA algorithm technique, it seemed an appropriate to finds their validity. For this purpose, testing dataset was used with Standard voting, voting with object tacking, and Naïve Bayes classifiers on the results extracted from training set (extracted rules). Among them, Naïve Bayes produced very high predication accuracy with 78.57% whereas 71.42% and 69.04% were obtained from Standard Voting and Voting with Object Tracking respectively.

Table T4. Results of Prediction Accuracy

Classifier	Accuracy
Naïve Bayes	78.57%
Standard Voting	71.42%
Voting with Object Tracking	69.04%

Moreover, many researchers have benched mark their study with 70% accuracy. For instance, Bakar [33] and Hvidsten [34] stated that results will be acceptable and effective if the prediction accuracy reaches 70% or above. Hence, this study also sets 70% as the benchmark of effectiveness. Based on it, only voting with object tracking couldn't generate the benchmark accuracy but Naïve Bayes and Standard Voting obtained the benchmark level. Therefore, rules proposed from this study can be used for finding effective programmer with team-leader and other programmers within the team.

VI. CONCLUSION

This research drew the pivotal importance of human aspect issues within the software development workplaces. It is considered one of the seven factors for software development failure in workplaces. Moreover, this research participated in software development literature by giving equations for finding effective programmer for team composition. For that purpose, academic population was focused to simulate the development environment.

Overall, the experiments started from data distribution with 70%-30% of training and testing sets. GA algorithm's results were chosen for further discussion and testing because it produced more effective results than JA algorithm. The key significance of the findings of this research, see table T3, lies in a fact that programmer's role was explained from different angles. Such as, E trait team-leader were found more efficient with E trait programmers than others while I trait team-leader appeared convenient working with I and E traits programmers. The results also suggested that I trait programmers were easy with team-leaders possessing I traits provided programmers possessed S trait of personality. Additionally, the findings also suggested that ENFJ personality male programmer rendered acute efficiency working with ENTJ personality team-leader. On the other hand, ESJF personality type emerged as an ideal mean for female programmer if team-leader was

ESTF. Moreover, this research also validated the results with few classification techniques. In which, Naïve Bayes obtained higher accuracy with 78.57% where the benchmark was only 70%. The standard voting, other classification technique, also obtained 71.42% prediction accuracy. Thus, the determination of the validity of the results through different classification techniques suggested practitioners to implement them for finding effective programmers.

VII. RECOMMENDATIONS AND FUTURE WORK

The present study also provides some valuable suggestions for future researchers who may wish to make software development research more mature. First, the future research may include large data that can generate more patterns conducive for team composition. Second, the present study employed certain techniques of data mining to suggest the model for programmer's role; therefore, future research study may undertake other techniques so as to establish a better accuracy in the results. Third, since this study solely accentuated on the programmers' role in software development, thus future studies are also recommended to look for different roles than programmers' role to make software development more feasible. The last but not the least, industrial data is also recommended for the future studies to make findings more generalizable and applicable without ambiguities.

REFERENCES

[1] B. Wysocki Jr, "Some Firms, Let Down by Costly Computers, Opt to 'De-engineer,,'" Wall Str. J., vol. 150, no. 84, 1998.

[2] T. Big and A. Small, "CHAOS MANIFESTO 2013," 2013.

[3] R. L. Glass, "Short-term and long-term remedies for runaway projects," Commun. ACM, vol. 41, no. 7, pp. 13–15, 1998.

[4] M. Keil, A. Rai, J. E. Cheney Mann, and G. P. Zhang, "Why software projects escalate: The importance of project management constructs," Eng. Manag. IEEE Trans., vol. 50, no. 3, pp. 251–261, 2003.

[5] B. Bang, "BIG BANG BOOM," pp. 1–12, 2014.

[6] A. R. Gilal, M. Omar, and K. I. Sharif, "A Rule-based Approach for Discovering Effective Software Team Composition," Journal of ICT, 13, 2014, pp: 1 – 20, 2014.

[7] L. F. Capretz and F. Ahmed, "Making sense of software development and personality types," IT Prof., vol. 12, no. 1, pp. 6–13, 2010.

[8] T. Dingsøyr and T. Dybå, "Team Effectiveness in Software Development Human and Cooperative Aspects in Team Effectiveness Models and Priorities for Future Studies," no. 7465, pp. 27–29, 2012.

[9] D. Robey and M. Newman, "Sequential patterns in information systems development: an application of a social process model," ACM Trans. Inf. Syst., vol. 14, no. 1, pp. 30–63, 1996.

[10] S. Sawyer, "Effects of intra-group conflict on packaged software development team performance," Inf. Syst. J., vol. 11, no. 2, pp. 155–178, 2001.

[11] W. Scacchi, "Socio-technical interaction networks in free/open source software development processes," in Software Process Modeling, Springer, 2005, pp. 1–27.

[12] D. Robey, "Research Report—Modeling Interpersonal Processes During System Development: Further Thoughts and Suggestions," Inf. Syst. Res., vol. 5, no. 4, pp. 439–445, 1994.

[13] S. Sawyer and P. J. Guinan, "Software development: Processes and performance," IBM Syst. J., vol. 37, no. 4, pp. 552–569, 1998.

[14] L. S. Kirsch, "Portfolios of control modes and IS project management," Inf. Syst. Res., vol. 8, no. 3, pp. 215–239, 1997.

[15] B. Curtis, H. Krasner, and N. Iscoe, "A field study of the software design process for large systems," Commun. ACM, vol. 31, no. 11, pp. 1268–1287, 1988.

[16] E. M. Trauth, "Theorizing gender and information technology research," Encycl. Gend. Inf. Technol., vol. 2, pp. 1154–1159, 2006.

[17] A. R. Gilal, J. Jaafar, M. Omar, and M. Z. Tunio, "Impact of Personality and Gender Diversity on Software Development Teams' Performance," in International Conference on Computer, Communication, and Control Technology (I4CT 2014),, 2014, no. 2014 IEEE 2014, pp. 261–265.

[18] N. Gorla and Y. W. Lam, "Who should work with whom?: building effective software project teams," Commun. ACM, vol. 47, no. 6, pp. 79–82, 2004.

[19] L. F. Capretz and F. Ahmed, "Why do we need personality diversity in software engineering?," ACM SIGSOFT Softw. Eng. Notes, vol. 35, no. 2, p. 1, Mar. 2010.

[20] S. S. J. O. Cruz, F. Q. B. da Silva, C. V. F. Monteiro, C. F. Santos, and M. T. dos Santos, "Personality in software engineering: preliminary findings from a systematic literature review," 2011.

[21] L. McLeod and S. G. MacDonell, "Factors that affect software systems development project outcomes," ACM Comput. Surv., vol. 43, no. 4, pp. 1–56, Oct. 2011.

[22] M. Omar, S.-L. Syed-Abdullah, and N. M. Hussin, "Analyzing personality types to predict team performance," 2010.

[23] J. S. Karn, S. Syed-Abdullah, A. J. Cowling, and M. Holcombe, "A study into the effects of personality type and methodology on cohesion in software engineering teams," Behav. Inf. Technol., vol. 26, no. 2, pp. 99–111, 2007.

[24] J. Karn and T. Cowling, "A follow up study of the effect of personality on the performance of software engineering teams," in Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering, 2006, pp. 232–241.

[25] A. R. Gilal, M. Omar, and K. I. Sharif, "DISCOVERING PERSONALITY TYPES AND DIVERSITY BASED ON SOFTWARE TEAM ROLES," in International Conference on Computing and Informatics, ICOCI 2013, 2013, pp. 259–264.

[26] M. Omar, S.-L. Syed-Abdullah, and N. M. Hussin, "Developing a Team Performance Prediction Model: A Rough Sets Approach," in Informatics Engineering and Information Science, Springer, 2011, pp. 691–705.

[27] A. Furnham, "The big five versus the big four: the relationship between the Myers-Briggs Type Indicator (MBTI) and NEO-PI five factor model of personality," Pers. Individ. Dif., vol. 21, no. 2, pp. 303–307, 1996.

[28] J. H. Bradley and F. J. Hebert, "The effect of personality type on team performance," J. Manag. Dev., vol. 16, no. 5, pp. 337–353, 1997.

[29] A. D. Da Cunha and D. Greathead, "Does personality matter?: an analysis of code-review ability," Commun. ACM, vol. 50, no. 5, pp. 109–112, 2007.

[30] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," Neural Networks, IEEE Trans., vol. 13, no. 2, pp. 415–425, 2002.

[31] J. W. Grzymala-Busse, "A new version of the rule induction system LERS," Fundam. Informaticae, vol. 31, no. 1, pp. 27–39, 1997.

[32] R. Menon, G. Bhat, G. R. Saade, and H. Spratt, "Multivariate adaptive regression splines analysis to predict biomarkers of spontaneous preterm birth," Acta Obstet. Gynecol. Scand., vol. 93, no. 4, pp. 382–391, 2014.

[33] A. B. Nauman, R. Aziz, A. F. M. Ishaq, and M. Mohsin, "An analysis of capabilities of Pakistan as an offshore IT services outsourcing destination," in Multitopic Conference, 2004. Proceedings of INMIC 2004. 8th International, 2004, pp. 403–408.

[34] T. R. en Hvidsten, "Fault diagnosis in rotating machinery using rough set theory and ROSETTA," 1999.