

A Code Level Based Programmer Assessment and Selection Criterion Using Metric Tools

Ezekiel U. Okike

Department of Computer Science
University of Botswana
Gaborone, Botswana

Abstract—this study presents a code level measurement of computer programs developed by computer programmers using a Chidamber and Kemerer Java metric (CKJM) tool and the Myers Briggs Type Indicator (MBTI) tool. The identification of potential computer programmers using personality trait factors does not seem to be the best approach without a code level measurement of the quality of programs. Hence the need to evolve a metric tool which measures both personality traits of programmers and code level quality of programs developed by programmers. This is the focus of this study. In this experiment, a set of Java based programming tasks were given to 33 student programmers who could confidently use the Java programming language. The codes developed by these students were analyzed for quality using a CKJM tool. Cohesion, coupling and number of public methods (NPM) metrics were used in the study. The choice of these three metrics from the CKJM suite was because they are useful in measuring well designed codes. By examining the cohesion values of classes, high cohesion ranges [0,1] and low coupling imply well designed code. Also number of methods (NPM) in a well-designed class is always less than 5 when cohesion range is [0,1]. Results from this study show that 19 of the 33 programmers developed good and cohesive programs while 14 did not. Further analysis revealed the personality traits of programmers and the number of good programs written by them. Programmers with Introverted Sensing Thinking Judging (ISTJ) traits produced the highest number of good programs, followed by Introverted iNtuitive Thinking Perceiving (INTP), Introverted iNtuitive Feelingng Perceiving (INTP), and Extroverted Sensing Thinking Judging (ESTJ)

Keywords—computer programs; program quality; class cohesion; programmers; personality traits

I. INTRODUCTION

Programming is a challenging task, which requires appropriate skills as well as appropriate temperamental suitability. Among the skills often demonstrated by professional and successful programmers are logical and analytical thinking, problem understanding and interpretation, detailed understanding of a programming language's syntax and a good communication ability. Capretz and Ahmed [1] identified some of the skills required in computer programming to include strong analytical and problem solving skills, communication skills, interpersonal skills, ability to work independently, active listening skills, innovative skills, organizational skills, openness and adaptability skills, fast learning skills and team playing skills. Apart from possessing these skills, success as a computer programmer may also be influenced by personality types such

as Extroversion (E), Introversion (I), Sensing (S), iNtuition(N), Thinking(T), Feeling(F), Judging(J) and Perceiving(P) (Okike and Olanrewaju[2]; Capretz and Ahmed [1]; Capretz[5]; Da Cunha and Greathead[3]; Tueley and Bieman [6]; Bentley [4]).

Furthermore, the reliability of a computer software depends on the code level quality of the program which indeed results from programmers coding skill. For this reason, it becomes very necessary to evolve code level measurement of program quality and individual programmer personality traits in the selection process of career computer programmers. Hence matching coding skill with personality traits will enable the identification and selection of good computer programmers. This is the motivation for this paper.

A. Problem Statement

Programmers are widely perceived as Introverts, Sensors and Thinkers (Capretz and Ahmed [1], Sensing and iNtuitionist, (Da Cunha and Greathead [3], Introverts, iNtuitionists, Thinkers and Judges (INTJ) (Tieger, [20]). These assessments are purely based on personality traits factors without recourse to code level quality of programs or resulting software. The present study seeks to bridge the gap between programmer personality traits and the quality of programs written by programmers by making use of a two level metrics based on both personality traits and code level quality to assess and to select competent programmers who create quality software programs. The Quality of a Program (QoP) in this study is measured in terms of the Cohesiveness of the Program module (CoPm), Coupling Between Object classes (CBO) and Number of Public Methods (NPM). In software development, high cohesion (range [0,1]) and low coupling imply good design. In addition, number of methods (NPM) in a well-designed class is always less than 5 when cohesion is high, range[0,1] and coupling is low [17]. The cohesion degree of a component is high if it implements a single logical function, and cohesive component tend to have high maintainability and reusability ((Okike [7], Badri [8], Bieman and Kang[9]))

B. Study Objectives

The main objective of this study is to create a two level metrics which is based on programmers personality traits and the code level quality of program modules. This instrument should be useful in selecting programmers who create quality programs. Specifically, the objectives of this study are to:

- investigate the personality traits of skilled programmers using Myers Briggs type indicator (MBTI)
- investigate the code level quality of programs written by programmers using Chidamber and Kemerer Java Metric tool (CKJM)
- suggest the personality type indicator(s) of competent programmers.

C. Research Questions

The following research questions are investigated in this study.

- What are the personality traits of good computer programmers?
- Which personality traits designed quality (cohesive) programs?

D. Research Hypotheses

The following hypotheses are tested in this study:

- H1: Introverts design better codes than extroverts in terms of class cohesion

H0: Introverts do not design better codes than extroverts

- H1: Sensors design better codes than intuitives in terms of class cohesion

H0: Sensors do not design better codes than intuitives

- H1: Thinkers design better codes than feelers in terms of class cohesion

H0: Thinkers do not design better codes than feelers

- H1: Judges design better codes than Perceivers in terms of class cohesion

H0: Judges do not design better codes than perceivers

- H1: There is significant correlation between personality traits and code quality

H0: There is no correlation between personality traits and code quality

The rest of this paper is divided into 7 sections. Section 2 is a presentation of the conceptual model of the study. Section 3 is the literature review. Section 4 explains the research methodology. Section 5 presents the result of this study with appropriate discussion. Section 6 is the conclusion while section 7 is the list of references

II. CONCEPTUAL FRAMEWORK

The framework for this study is based on Capretz and Ahmed [1] model : Mapping programmers and skills to personality type as shown in figure 1 below and Okike [7] Metric calculation Process using Chidamber and Kemerer metric tool as shown in figure 2. Arising from these two models is a hybrid adapted from the two to achieve the objectives stated in section 1.3.

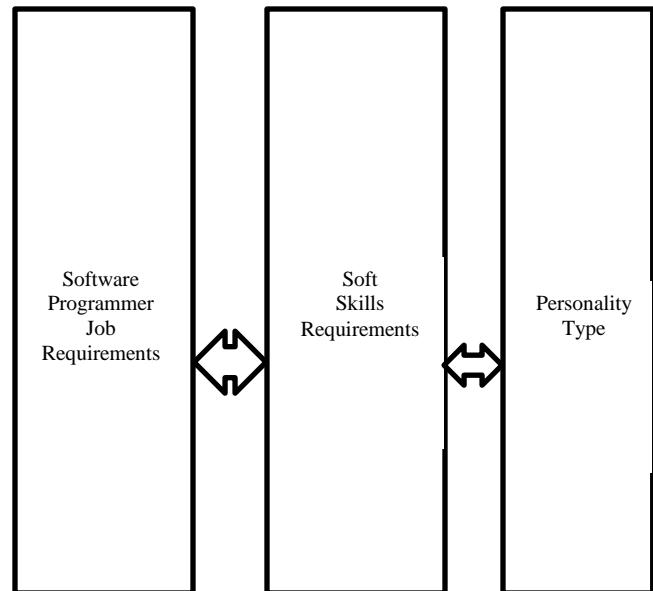


Fig. 1. Mapping Programmers and skills to personality type Adapted from Capretz and Ahmed [1]

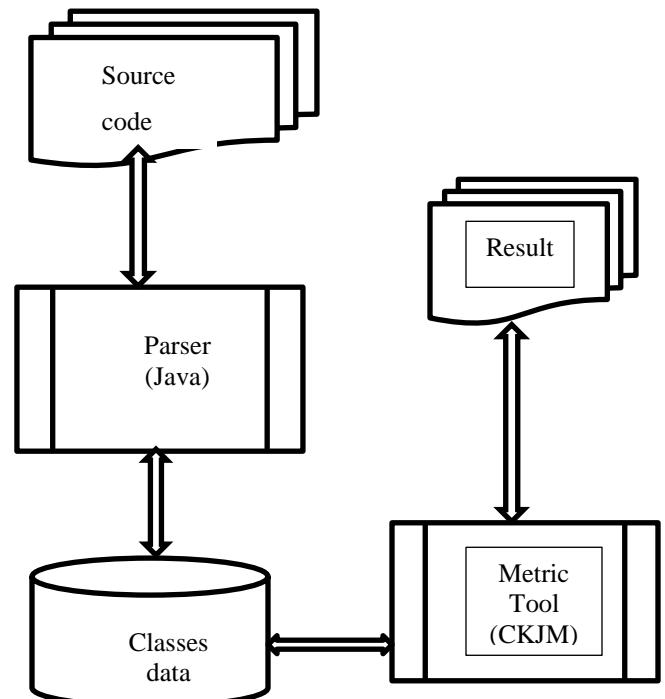


Fig. 2. Metric calculation processSource: Adapted from Okike [7], and Badri [8]

A. Proposal of a two level Metrics Model for selecting programmers

Figure 3 below shows a hybrid model metric tool for selecting programmers who create cohesive software. Since code level quality is measured by high cohesion, any metric tool which measures cohesion in software would be good candidate. In this study, the Chidamber and Kemerer Java Metric (CKJM) tool is used with particular focus on the Lack

of Cohesion in Methods (LCOM) metric. The hybrid tool has 2 levels as shown in figure 3 namely: level 1 – programmer personality trait measurement using the Myers Briggs Type Indicator (MBTI) and level 2- program quality measurement using Chidamber and Kemerer Java metric (CKJM) tool.

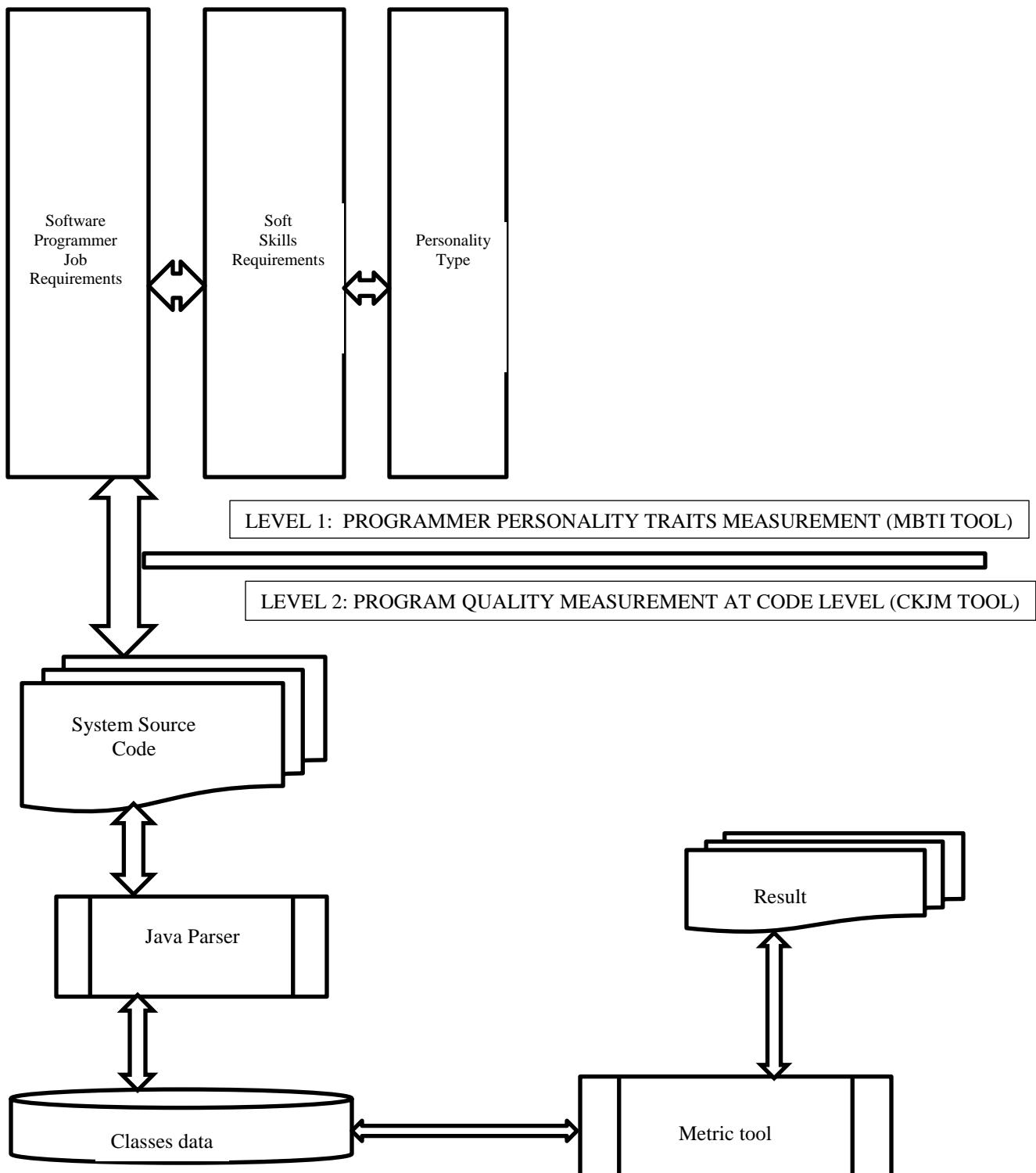


Fig. 3. A hybrid Metric Modelling Tool for Programmers Selection

III. LITERATURE REVIEW

Code level measurement of program quality have been studied using class cohesion, coupling and other metrics from the Chidamber and Kemerer metric suite [7,8,9,16,17,19]. High cohesion, range [0,1] and low coupling imply good design. The term cohesion is defined as the “intramodular functional relatedness” in software [22]. Chidamber and Kemerer [19] first defined a cohesion measure for object oriented software- the Lack of cohesion in Methods (LCOM) metric. Okike [7] studied class cohesion measurement in object oriented systems using Chidamber and Kemerer Metric suite and Java as case study. The study involved 6 different types of Java based industrial systems with over 3000 classes. The result of the study showed that the Lack of Cohesion in Methods metric (LCOM) defined by Chidamber and Kemerer was suitable in measuring class cohesion in the studied systems. In addition the study showed that the LCOM metric satisfy measurement theory conditions, and although the metric is prone to outliers; a new metric was defined which normalizes the LCOM metric such that outliers were eliminated. Furthermore, a pedagogical evaluation and discussion about the Lack of Cohesion in Methods metric using field experiments is presented in Okike[16], while a normalized Lack of Cohesion in Methods metric is presented in Okike [17]. In both studies, the usefulness of LCOM metric alongside Coupling between Object (CBO) and Number of Public Methods (NPM) in the evaluation of well-designed classes were clearly established . Hence by measuring cohesion using the LCOM, CBO and NPM metrics in this study, well designed codes by individual programmers were identified

Furthermore, the Myers Briggs Type Indicator(MBTI) has been widely used by researchers to measure the personality traits of individuals in various capacities and dimensions. Okike and Olanrewaju [2] investigated problem solving and decision making skills of 30 student programmers using the MBTI tool. A decision problem representing a programming task was given to the students. The students were expected to produce computer programs which solves the given problem. The MBTI, an automated personality traits questionnaire based tool was administered on the students. The responses from students were automatically analyzed in order to identify the personality traits of each student. The program code or codes written by each students was also analyzed using a Chidamber & Kemerer Java Metric (CKJM) tool, and the results matched with their corresponding MBTI to determine the problem solving and decision making skill of each programmer by looking at the quality of the resulting program code. The study concluded that The result of this study indicates that among the various personality traits, the Introverted Sensing Thinking Judging (ISTJ) appear to have the best problem solving and decision making skill followed by Introverted Intuitive Feeling Judging (INFJ) compared to other personality traits. However, in all, candidates with personality traits such as Introverted Sensing Feeling Perceiving(ISFP), Introverted Intuitive Thinking Perceiving (INTP), Extroverted Intuitive Feeling Perceiving (ENFP), Introverted Sensing Feeling Judging (ISFJ), Extroverted Intuitive Thinking Judging (ENTJ), Extroverted Sensing

Feeling Judging (ESFJ), Extroverted Intuitive Feeling Judging (ENFJ), Introverted Sensing Thinking Perceiving (ISTP), and Introverted Intuitive Feeling Perceiving(INFP) are likely to have averagely problem solving and decision making skills while individuals with Extroverted Sensing Feeling Perceiving (ESFP) and Extroverted Sensing Thinking Perceiving (ESTP) traits appear to have poor problem solving and decision making skills.

Okike [10] investigated the role of personality traits in students' achievements in Computing Science. Results from the study suggests that the strongest motivator for a choice of career in the computing sciences is the desire to become a computing professional rather a students inherent temperamental ability (personality traits). Equally, students' achievements in the computing sciences do not depend only on personality traits, motivation for choice of course of study, and reading habits but also on the use of Internet based sources more going to the university library to use book materials available in all areas.

Okike [11] studied the bipolar factor and systems analysis skills of 60 students analysts at the University of Botswana. The study evolved a new approach to construct a type matrix from a personality type frequency matrix. This approach was used to select the best systems analyst based on personality traits factors.

Bentley [4] reviewed personality traits and programmer characteristics and presented some of the traits that can be indicators of success or failure in computer programming. Weinberg [13] explored the psychology of computer programming and noted that there could be variations in individual productivity due to personality type factor. Capretz [5] investigated personality types of software engineers based on the combined Jung and Myers Briggs bipolar. The study suggested that they were more (Introvert Sensing Thinking Judging (ISTJ) software engineers than other types in his data. Chung [15] studied the cognitive abilities in computer programming using 523 form four secondary school students in Hong Kong. Test administered to the students included mathematics, space, symbols, hidden figures and programming ability. Results of this study suggested that performance in mathematics and spatial tests were significant predictors in programming ability. Similarly, Bishop-Clark and Wheeler [14] investigated the Myers-Briggs personality type and its relationship to computer programming. Using 114 students, the study sought to know if college students with certain personality types performed better than others in an introductory programming course. In this study, results suggested that sensing students performed significantly better than intuition students in programming assignments while judging students performed better than perception students on computer programs although the results were not significant statistically.

IV. STUDY METHODOLOGY

A set of Java based programming task was given to some 33 student programmers who could use the Java programming language confidently. A Chidamber and Kemerer Java metric tool (CKJM) [18] was used to analyse the quality of program codes written by each participating programmer. In addition

the Myers Briggs Type indicator (MBTI) was used to measure the personality traits of each participating programmer. In this way, a two level metrics based approach was evolved namely:

Level 1: Human metric tool (MBTI)

Level 2: Code level metric tool (CJMK)

The Human metric tool is based on the Myers Briggs Type Indicator tool. Each participating programmer completed and submitted the automated MBTI questionnaire and was subsequently scored by the tool as to the appropriate personality trait.

At level 2, the programmers were given the same programming task, and each of them developed appropriate Java codes. The codes were evaluated automatically by applying the CKJM tool. The CKJM tool calculates for each program class the following six metrics when used in any experiment [18]

- WMC: Weighted methods per class
- DIT : Depth of Inheritance Tree
- NOC: Number of children

TABLE I. CLASS DESIGN AND PROGRAMMING ABILITY OF STUDENTS USING CHIDAMBER AND KEMERER METRIC SUITE AND MBTI

COLS/ 1	2	3	4	5	6	7	8	9	10	11
S/N	Programmer MBTI	Lines of Code	WMC	DIT	NOC	CBO	RFC	LCOM	CA	NPM
1	ENFJ	97	8	1	0	0	15	1	0	2
2	ENFP	72	2	1	0	0	11	1	0	2
3	ENFP	70	5	1	0	0	13	10	0	2
4	ENTJ	124	7	1	0	0	17	22	0	3
5	ENTJ	84	2	1	0	0	13	1	0	2
6	ENTJ	139	7	1	0	0	16	21	0	7
7	ESFJ	85	2	1	0	0	9	1	0	2
8	ESFP	130	7	1	0	0	8	13	0	4
9	ESTJ	187	19	6	0	4	46	115	0	5
10	ESTJ	121	6	1	0	0	19	15	0	6
11	ESTJ	59	5	1	0	0	7	1	0	2
12	ESTJ	122	18	6	0	4	42	97	0	4
13	ESTJ	34	2	1	0	1	7	1	0	2
14	ESTP	81	6	1	0	0	17	3	0	6
15	INFJ	89	2	1	0	0	17	1	0	2
16	INFJ	85	2	1	0	0	6	1	0	4
17	INFP	90	2	1	0	1	25	1	0	2
18	INFP	38	5	1	0	0	6	6	1	5
19	INTP	24	2	1	0	0	13	1	0	2
20	INTP	127	20	6	0	3	40	142	0	3
21	INTP	111	2	1	0	0	17	1	0	2
22	ISFJ	129	7	1	0	0	20	21	0	6
23	ISFJ	55	2	1	0	0	21	1	0	8
24	ISFP	63	2	1	0	0	12	1	0	2
25	ISTJ	40	2	1	0	0	12	1	0	2
26	ISTJ	63	2	1	0	0	14	1	0	2
27	ISTJ	58	2	1	0	0	12	1	0	2
28	ISTJ	109	4	1	0	1	25	0	0	3
29	ISTJ	56	9	1	0	0	10	16	1	9
30	ISTJ	250	14	1	0	0	6	6	1	5
31	ISTJ	119	2	1	0	1	31	1	0	2
32	ISTP	125	9	1	0	0	20	34	0	3
33	ISTP	57	2	1	0	0	12	1	0	2

Source: Adapted from [21]

- CBO: Coupling between object classes
- RFC: Response for a class
- LCOM: Lack of cohesion in methods
- Ca: Afferent coupling
- NPM: Number of Public Methods for a class

For the purpose of this paper, the LCOM, CBO and NPM metrics are mainly considered in the assessment of code quality . This follows from earlier research as shown in [7,16,17]. High cohesion range [0,1] and low coupling implies good design. Also the number of methods n in a well-designed class should be less than 5 [17:pg22].

V. RESULT AND DISCUSSION

Table 1 below shows the result of the experiment described above in section 3. The Myers Brigg Type Indicator (MBTI) of each programmer and the Lack of Cohesion in Methods (LCOM) metric of program classes are considered together.

Using the MBTI tool, the personality characteristics of the programmers were established as shown in column 2, while the corresponding program quality characteristics of program codes written by the programmers are shown from columns 3-11 as measured by the Chidamber and Kemerer Java Metric (CKJM) tool [18]. A comprehensive discussion about the Chidamber and Kemerer suite of metrics is presented in [19]. Furthermore, a pedagogical evaluation and discussion about the usefulness of Chidamber and Kemerer's metric suite, particularly the Lack of Cohesion in Methods (LCOM) metric is presented in [7, 16, 17]. Using the CKJM tool the following metric were computed for each class or program module written by a programmer: Weighted Method per class (WMC), Depth of Inheritance Tree (DIT), Number of Children (NOC), Coupling Between Object (CBO), Response for a Class (RFC), Lack of Cohesion in Methods (LCOM), Afferent Coupling (CA), and Number of Public Methods (NPM). Details about each of this metric have been discussed in [7,18,19].

A. Bipolar Factor Characteristics s of Candidates

Table 2 below presents the personality frequency matrix of the participating programmers [11].

TABLE II. PERSONALITY FREQUENCY MATRIX

Personality Type		Type Indicator	N
Extroversion(E)	Introversion (I)	EI	3
14	19	33	3
Sensing (S)	iNtuition (N)	SN	3
20	13	33	3
Thinking (T)	Feeling (F)	TF	3
21	12	33	3
Judging (J)	Perceiving (P)	JP	3
21	12	33	3

From this table, the dominant personality traits are Thinking (T) =21, Judging (J) = 21, Sensing(S) = 20, and iNtuition (N) = 19. Arising from Table 2, a type matrix table is presented in Table 3. Diagonals of type matrix tables must sum up to the total number of participants [11]

TABLE III. TYPE MATRIX TABLE

E	N	T	P
14	13	21	12
20	19	21	12
S	I	J	F

VI. DISCUSSION

From Table 1, programs with LCOM value in the range [0,1] are cohesive, and hence well designed. These programs were written by candidates with serial numbers 1,2,5,7,11,13,15,16,17,19,21,23-28,31 and 33. The corresponding personality traits of these candidates are ENFJ, ENFP, ENTJ, ESFJ, ESTJ, INFJ, INFJ, INFP, INTP, INTP, ISFJ, ISFP, ISTJ, ISTJ, ISTJ, ISTJ, ISTP (research questions bullet 1 and 2). Overall the number of well designed cohesive programs based on personality traits are shown in Table 4 below

TABLE IV. PERSONALITY TYPE AND GOOD PROGRAM DESIGN

Personality Type	No of Cohesive Programs
ENFJ	1
ENFP	1
ENTJ	1
ESFJ	1
ESTJ	2
INFJ	2
INFP	1
INTP	2
ISFJ	1
ISFP	1
ISTJ	4
ISTP	1

Considering the bipolar factors – Extroversion (E), Introversion (I), Sensing (S), iNtuition (N), Thinking (T), Feeling (F), Judging (J), and Perceiving (P), the number of well designed program codes are shown in Table 5 below

TABLE V. GOOD PROGRAMS BY BIPOLAR FACTOR

Bipolar Factor	No of good programs
Extroversion (E)	10
Introversion (I)	6
Sensing (S)	10
Intuition (N)	8
Thinking (T)	10
Feeling (F)	8
Judging (J)	12
Perceiving (P)	6

Table 5 also provides answers to research questions (bullets 1 and 2) of this study. From this study, introverts appear not have better code design ability than extroverts. In fact, extroverts could be better programmers than introvert (hypothesis bullet 1).

Sensors could design better codes than iNTuitives (hypothesis bullet 2). Thinkers could design better codes than feelers (hypothesis bullet 3). Judges could be better code designers than perceivers (hypothesis bullet 4). The study suggests that there is significant relationship between personality traits and code quality (hypothesis bullet 5). This result is also supported in [2].

VII. CONCLUSION

In this study, a model for measuring both the personality traits of individual programmers and the quality of programs developed by these programmers at two levels has been presented. The model could be used when selecting competent computer programmers since the quality of well designed computer program can be measured by the level of cohesiveness of the program module or class [7,8,9,16,18,19]. In addition, good computer programmers appear to have strong personality traits such as judging, extroversion, sensing, thinking, intuition, feeling, and could have introversion and perceiving abilities. This conclusion supports previous studies presented in [1, 2,3,20]. Further details about the peculiarities of these traits are fully discussed in [22].

REFERENCES

- [1] F. Capretz, and F. Ahmed, "Making sense of software development and personality types," ITPro, vol. 12, No. 1 January/February 2010.
- [2] E. U. Okike and A. Olanrewaju, "Problem solving and decision making: consideration of individual differences in computer programming skills using Myers Briggs Type Indicator (MBTI) and Chidamber and Kemerer Java Metric (CKJM)". *Journal of Applied Information Science and Technology*, Vol. 7. No. 1, pp. 27-34 2014 .
- [3] D. A. Da Cunha, and D. Greathead, " Does Personality matter? An Analysis of code – review ability," Communications of the ACM. Vol. 50. No. 5, pp. 109-111, May 2007
- [4] J. E. Bentley,"Laziness, Impatience, Hubris: Personality Traits of a great Programmer". Unpublished
- [5] F. L. Capretz, "Personality types in Software Engineering", *Int. J. Human-Computer Studies*. 58, pp207-214, 2003
- [6] R. T. Turley, and J. M. Bieman, "Competencies of exceptional and non exceptional software engineers," *J. Systems Software*. 28:19-38, 1995
- [7] E. U. Okike , " Measuring class cohesion in Object Oriented System Using Chidamber and Kemerer Metric Suite and Java as case Study". Ph.D Thesis Department of Computer Science, University of Ibadan, 2007. Unpublished
- [8] L. Badri and M. Badri, "A proposal of a New Class cohesion Criterion: An experimental study" *Journal of Object Technology*, vol. 3, no. 4, pp. 145-159, April 2004
- [9] M. Bieman and B. K. Kang , "Cohesion and reuse in Object Oriented Systems", Proceedings of the Symposium on Software Reusability (SSR'95), Seattle WA, pp. 259-262, April 1995
- [10] Ezekiel U Okike, "Investigating students' achievement in computing science using human metric", *International Journal of Advanced Computer Science and Applications (IJACSA)*, Vol. 5. No. 4, pg. 180-186, 2014
- [11] E. U. Okike. "Bipolar Factor and Systems Analysis Skills of Student Computing Professionals at University of Botswana, Gaborone," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 5. No. 3, 2014
- [12] M. Omar, and S Syed-Abdullah, "Identifying effective software engineering (SE) team personality types composition using rough set approach," IEEE 2010
- [13] G. M. Weinberg, "The Psychology of Computer Programming", 2nd ed. Van Noststrand Reinhold:New York, 1998
- [14] C. Bishop-Clark and D. Wheeler, "The Myers-Briggs personality type and its relationship to computer programming," *Journal of Research on Computing in Education*. Vol. 26 Issue 3, pg. 358-371, 1994
- [15] C. Chung, "Correlates of problem solving in programming" CUHK Educational Journal Vol. 16. No. 2 pp185-190, 1986
- [16] Ezekiel Okike, "A Pedagogical Evaluation and Discussion about the Lack of Cohesion in Method (LCOM) Metric Using Field Experiment", *IJCSI International Journal of Computer Science Issues*, Vol. 7, Issue 2, No 3, pg. 36-43, March 2010
- [17] Ezekiel Okike, "A Proposal for Normalized Lack of Cohesion in Method (LCOM) Metric Using Field Experiment", *IJCSI International Journal of Computer Science Issues*, Vol. 7, Issue 4, No 5, pg. 19-26, July 2010
- [18] S. Diomidis "Tool writing: A forgotten art?". *IEEE Software*, 22(4), pp.9-11, July/August 2005.
- [19] S. R. Chidamber and C. F. Kemerer , "Towards a metric suite for Object Oriented Design, Object Oriented Programming Systems, Languages and Applications", *Special Issue of SIGPLAN Notices*, vol. 26, No.6, pg 476-493, 1994
- [20] Tieger, Paul D. and Barbara Barron- Tieger , "Do what you are. Boston: Little, Brow, and company, 2001.
- [21] A. Olanrewaju, "Problem solving and decision making: consideration of individual differences in computer programming skills using Myers Briggs Type Indicator (MBTI) and Chidamber and Kemerer Java Metric (CKJM)", M.Sc Dissertation, Department of Computer Science, University of Ibadan Nigeria, 2012. Unpublished.
- [22] E. Yourdon, and L. Constantine, "Structured design: Fundamentals of a Discipline of computer program and systems Design, Englewood Cliffs, New Jersey: Prentice-Hall, 1979