Journal of Science Research (2013) Vol. 12: 167-178

A Study on the Interplay of Cognition in Computer Programming and Code Inspection Skills in an Academic Environment

***Olalekan S. Akinola and Oyedeji Oluwatosin**

Abstract
This study examined the interplay of cognition in code inspection and programming skills among software developers. It has been observed that the style of programming employed differs from one programmer to another even when solving the same problem. Also, large variations in individual performance in software development have been observed. Some form of innate programming ability might account for some of the differences. The cognitive styles of the participants were determined using Myers-Briggs Type Indicator (MBTI) tool, their programming skills were assessed by giving them programming task to do and the code inspection skill was also assessed by giving them a programming code seeded with logical errors for them to debug. The performances of the participants were mapped against their cognitive styles to see if there is any form of relationship among the three variables. It was obtained from the experiment that the participants with the cognitive style ESFJ have a strong positive correlation between programming and code inspection skills compared with other cognitive styles though statistical tests result showed that all the cognitive styles performed at the same level.

Key words: Cognition, Code Inspection, Computer programming, MBTI.

Introduction

Cognition is a sub discipline of psychology exploring internal, mental processes. It studies how people perceive, remember, think, speak and solve problems. Cognition can be defined as the processes an organism uses to organize information. This includes acquiring information (perception), selecting (attention), representing (understanding) and retaining (memory) information, and using it to guide behaviour (reasoning and coordination of motor outputs) [1]. In science, cognition refers to mental processes. These processes include attention, remembering, producing and understanding language, solving problems and making decisions.

Cognitive style is defined as the different ways people receive, organize and process information. It acts as a mechanism controlling attention, thought, and actions and represents the person's internal preference for using a unique type of thinking [2, 3]. Each individual has different cognitive styles which are related to mental behaviours, which individuals apply habitually when they are solving problems. Cognitive style is an immutable characteristic of personality [4].

Cognition is studied in various disciplines such as psychology, philosophy, linguistics, science and computer science. This study experimentally examined the role of cognition in software inspection and programming skills among software developers using Java programming language as the software developers' programming language. In essence, the study examined how the uniqueness of a software professional is related to the programming and software inspection skills.

An inspection is one of the most common sorts of review practices found in software projects. The goal of the inspection is to identify defects. In an inspection, a defect is any part of the work product that will keep an inspector from approving it. For example, if the team is inspecting a software requirements specification, each defect will be a text

***Olalekan S. Akinola and Oyedeji Oluwatosin**
**Department of Computer Science, University of Ibadan, Nigeria*
solom202@yahoo.co.uk
ibukunmi@gmail.com

ISSN 11179333

in the document which an inspector disagrees with.

A software developer also known as a computer programmer or coder, is a person who writes computer software. The term computer programmer can refer to a specialist in one area of computer programming or to a generalist who writes code for many kinds of software. A programmer's primary computer language (C, C++, Java, Lisp, Python etc.) is often prefixed to the above titles, and those who work in a web environment often prefix their titles with web [5]. Computer programmers write, test, debug, and maintain the detailed instructions, called computer programs that computers must follow to perform their functions.

Programming style is a set of rules or guidelines used when writing the source code for a computer program [6]. It is often claimed that following a particular programming style, will help programmers to read and understand source code conforming to the style, and help to avoid introducing errors. Programming style used in a particular program may be derived from the Coding Conventions of a company or other computing organization, as well as the preferences of the author of the code.

This paper addresses the question of whether or not, it is possible that cognition style determines the individuals that are good in programming and code inspection skills? Also, which cognitive style has the best software developing and code inspection skills and what is the role of cognition in programming and code inspection? Specifically it investigates if there is a specific cognitive (Personality) type that members who perform better at computer programming and code inspection displays. The research was motivated by the fact that students who are taught under the same classroom and laboratory conditions perform differently at programming.

Related Works

The fields of cognition and programming are related in three main ways. First, cognitive psychology is based on a "computational metaphor", in which the human mind is seen as a kind of information processor similar to a computer. Secondly, cognitive psychology offers methods of examining the processes underlying performance in computing task. Thirdly, programming is a well- defined task, and there are an increasing number of programmers, which make it an ideal task in which to study cognitive processes in a real-world domain [7].

Rebecca, et al. [8] carried out a research work on "Aspects of Cognitive Style and programming". Their work describes the results of an experiment to test two cognitive characteristics that have been shown to be important in other conceptual areas: working memory space and field dependency. The results show that whilst working, memory space appears to have only a marginal influence on levels of achievement on the course, field dependency is an important factor in determining success.

Garry [9] carried out a study work on "Visual Basic programming Impact on Cognitive Style of College Students: Need for Prerequisites". This research investigated the impact, learning a visual programming language, visual basic, has on hemispheric cognitive style, as measured by the Hemispheric Mode Indicator (HMI). The study determines if a computer programming course helps students to improve their cognitive abilities in order to perform well. The study found that the hemispheric cognitive style remained the same after a semester course in Visual Basic. College age students' cognitive style was not impacted.

Akinola et al. [10] carried out a study on the effect of cognition on programming. This paper describes the performances of the students which were mapped against their cognitive styles to check for any form of relationship between the two variables (cognitive styles and programming skill) using MBTI. A critical qualitative examination of the results obtained showed that MBTI Cognitive style ISTJ seems to have better programming skill than the participants with other cognitive styles. Statistical test performed on the data show that all the different cognitive styles obtained in this

work with the participants have the same programming skill level.

Garry et al. [11] study on "A Theory of the Relationships between Cognitive Requirements of Computer Programming Languages and Programmers Cognitive Characteristics", formulates a theory that investigates the possible effects of two human cognitive characteristics on the difficulties of learning specific programming languages. They conclude that,

(i) If the cognitive requirements for a programming language are beyond the cognitive characteristics of a programming student, the student may burn out.
(ii) If the cognitive requirements are below the student's cognitive characteristics the student may be bored.
(iii) If they are similar to them, the student is able to meet the challenges.

James et al. [12] study on "A Cognitive-Based Mechanism for Constructing Software Inspection Teams" describes software inspection as a well-known and effective means of defect detection. It is also a cost-effective way of verifying documents. An inspection can only be as effective as the individuals contributing to its overall success; hence, an improved selection mechanism promises a significant return in terms of the overall effectiveness of the process. This paper presents an alternative process based upon an individual's cognitive style mechanism and argues that a team with diverse information processing strategies, as defined by the selection mechanism, will maximize the number of different defects discovered.

Using Bloom's cognition taxonomy, Mc Meekin et al. [13] found that software developers who performed a code inspection prior to modification tend to modification tend to operate at higher cognition levels; and vice-versa. They also went further to find out that, inspection techniques that utilize a more structured reading process are associated with higher cognition levels [14]. According to Jung's theory [12, 15], people may be categorized as either thinking or feeling in terms of their judgment tendencies. People in the "thinking" category focus on substantial research results. In making their decisions, they rely on firm, objective and analytic research. In the decision process, they depend on cognition over cause and effect. In contrast, "feeling" people focus on their feelings and consider the relative values of an issue. They attach importance to the understanding of personal values and group values before decision-making. In the decision process, they depend on effect over cognition and tend to use logic to support feelings.

Jung's theories also extend to how people prefer to process the information they are gathering. He divides this characteristic into the categories: extraversion and introversion. Extraverted people process information through social interaction. That is they develop their ideas through interaction with others. They verbalize ideas in order to emphasize them, and they prefer face-to-face meetings rather than memos and written communication. On the other hand, introverted people process information internally. They prefer to develop ideas and make decisions individually in an isolated environment. They seldom verbalize ideas or opinions, and they prefer written forms of communication to presentations and meetings.

Myers [16] extended Jung's theory by proposing an additional dimension, describing a person's preference between gathering and evaluating information. "Perceptive" people would like to gather information rather than evaluate information; therefore, they have difficulty in stopping this process. They keep on looking for new information before making a decision and often move from one project to another. They prefer to remain flexible and do not like fixed plans. They tend to be spontaneous, curious, adaptable, and open to new events and changes. "Judging" people prefer evaluating information to gathering information. They

prefer to produce order and develop "blueprints," and they tend to create environments that are regulated and controlled. They develop cognitive schemas as a basis of perceiving, in order to shorten information gathering processes in the future. They tend to finish one task before starting another and plan activities entirely before commencing.

Although no single definite classification of these ideas is ideal, many of the classification systems have a great deal in common. The Myers-Briggs Type Indicator (MBTI) has great popularity within the personality psychology community; its simplicity of implementation and its explicitness that a person must fall into one or the other of the ordered pair makes it a good classification tool for this research.

This personality characterization test has been successfully administered via questionnaires with automatic analysis [17], successfully applied to managerial practice [18] and extensively validated [16]. It is important to note that the MBTI does not intend to measure intelligence, aptitude, or achievement. Rather, it reflects what an individual prefers [19].

Research Methodology

This study determines if the cognitive style of an individual influences the programming and/or software inspection skills of that individual. To achieve this, two experiments were carried out to determine the programming and code inspection skills of 50 students in the department of Computer Science, University of Ibadan, Nigeria. The participants in the experiments passed through the same learning process in Java programming and software inspection for a period of thirteen weeks of lectures each for the First and Second Semesters of 2011/2012 Sessions. Only students who had no residual knowledge of computer programming before were purposefully selected. Hence, it was assumed they acquired the same level of programming ability and code inspection skills.

Three major tools were used: one to measure the cognitive style of the participants in the experiment, second to measure their programming skills and third, to measure their software inspection skills.

Cognitive Style Determination Phase

To measure the cognitive style of the participants, the MBTI standard tool was adopted. The primary feature of the theory behind the MBTI is that each person's personality fits into only one of 16 types. These categories are based on four features of personality, each consisting of two opposite preferences. According to the MBTI theory [20], all people have an innate preference that determines how they will behave in all situations.

The MBTI is a theory of types, that is why a person can have only one preference. Although, it is possible for people to develop the complimentary style (an introvert, for example, could learn to be more extroverted when speaking in groups) the primary preference will always dominate the person's personality. A person's MBTI score determines his or her type, a label based on his or her dominate preference for each of the four dimensions. Since, there are two preferences within each dimension, there are 16 potential personality types.

According to Daniel, et al. [21], MBTI puts forward sixteen different personality types and each "type" is given a four-letter name, such as ENFP (Extrovert iNtuitive Feeling Perceiving) according to the four basic dimensions (Tables 1a and b). Each personality type is said to be different from the others i.e. ESTJ is a different personality type from ISTJ and ISTJ is a different personality type from ESTP etc.

Table 1a: The Four Opposite Dimensions of MBTI and the Restricted Two Opposite Dimensions (In Bold) of MBTI

Extraversion	Introversion
Sensing	iNtuition
Thinking	Feeling
Judgment	Perception

Table 1b: The 16 Different Personality Types

ISTJ	ISFJ	INFJ	INTJ
ISTP	ISFP	INFP	INTP
ESTP	ESFP	ENFP	ENTP
ESTJ	ESFJ	ENFJ	ENTJ

MBTI looks at where we take our inspiration from, how we prefer to receive and process information, and how we choose to order our lives. It is widely used in business, organizations, and governments, and is especially useful in such areas as team building, relationship counselling, management and leadership skills, communication, training and career development.

Programming Skill Determination Phase

To measure the programming skills of the participants, a standard practical programming question was given to the participants after they have learnt Java programming language for a period of two semesters. The student participants were asked to write a program in-line with the specification of the problem for one hour. Thereafter, their programming skills in terms of programming styles used, syntax, efficiency of the program, accuracy and internal comments were determined. The programming question was graded over ten (10) marks based on the programming skills listed above.

Software Inspection Skill Determination Phase

To measure the software inspection skills of the participants a programming code seeded with bugs was given to the participants to detect the defects seeded with bugs after they have learnt code inspection techniques and their advantages. The participants were initially taught how to conduct code inspection in a CSC 233 (Programming and Algorithms) class. Some forms of defects seeded into the five-page (A4 size paper) code were incorrect functionality (or interpretation) and incomplete statement. Ten logical errors were seeded into the codes, such as the use of wrong logical operators (<, >, && and ||) or division by wrong divisor variable. The inspection was done for one hour. The exercise was graded over ten (10) marks.

Results

The results obtained for the Cognitive Style Determination Phase is shown in Table 3 and Figure 1. Table 2 shows that a total of 50 students participants were involved in the experiments. The table and Figure 1 show that majority of the participants were of type ESTJ and ISTJ, followed by ESFJ and ISFJ which have the same number. Cognitive styles such as ESFP and ENFP were not found among the participants; hence the cognitive types of the participants were not evenly distributed.

Table 2: Cognitive Style Distribution of Participants

ISTJ 10	ISFJ 4	INFJ 1	INTJ 1
ISTP 1	ISFP 1	INFP 1	INTP 1
ESTP 1	ESFP 0	ENFP 0	ENTP 2
ESTJ 19	ESFJ 4	ENFJ 2	ENTJ 2

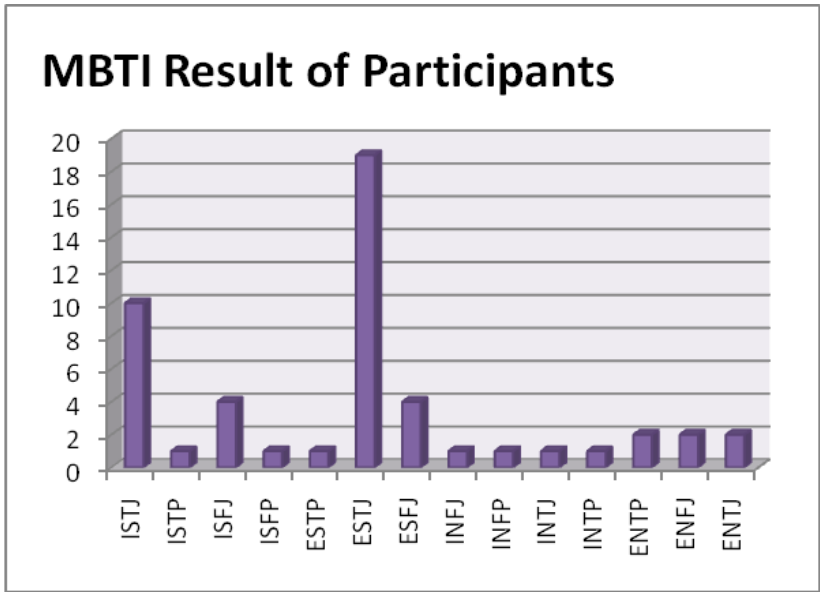

Fig. 1: MBTI results of participants.

Table 3 and Figure 2 show the breakdown of the combined cognitive types into their component types so as to have a better picture of the cognitive characteristics of the participants. It shows that we have more Extroverts' than Introverts, Sensing than iNtution, Thinkers than Feelers and Judging than Perception characteristics. Table 4 shows the result of the participants in programming skill assessment and code inspection with their respective cognitive styles. From the result of the programming and code inspection skills, participants that scored five and above were regarded as having good programming and code inspection skills. Only eighteen participants had good programming skill and twenty nine participants had good code inspection skill.

Table 3: Distribution of Participants into their Dichotomies

MBTI Dichotomy	No of Participants	MBTI Dichotomy	No of Participants
Extraversion	30	Introversion	20
Sensing	**40**	**iNtuition**	**10**
Thinking	**37**	**Feeling**	**13**
Judgment	43	Perception	7

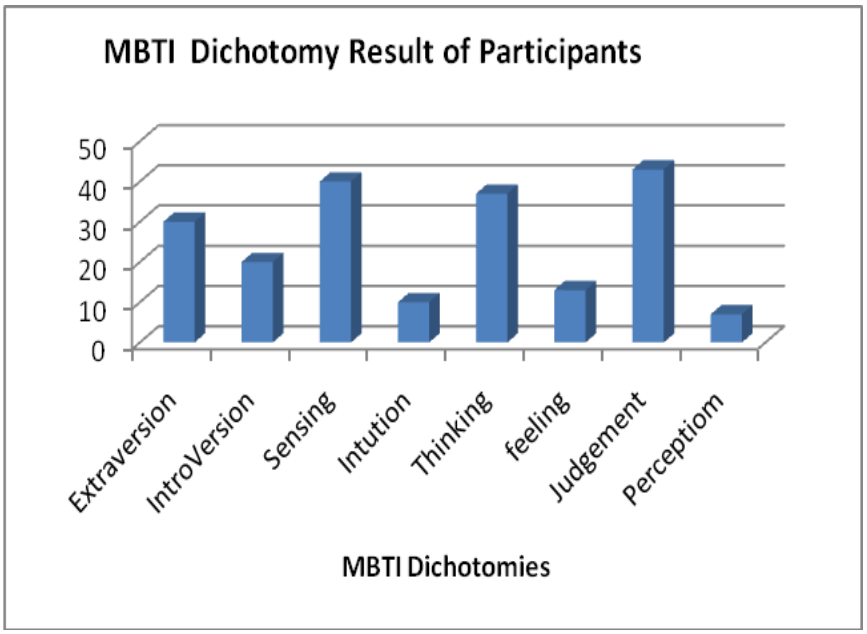

Fig. 2: Chart showing the participants cognitive type.

Table 4: Result of Programming Skill and Code Inspection

Participants	Score in Programming	Code Inspection	MBTI Type
1	3	3	INTP
2	3	8	ISTJ
3	7	4	ISTJ
4	3	3	ISTJ
5	3	7	ESTJ
6	4	4	ISFP
7	5	6	ESTJ
8	6	5	ESFJ
9	5	4	ISFP
10	4	6	ISTP
11	6	7	ISTJ
12	3	2	ESTJ
13	1	0	ENTP
14	4	4	ISFJ
15	3	5	INFJ
16	4	6	ESTJ
17	7	6	ESFJ
18	4	5	ESTJ
19	3	6	ISTJ
20	3	5	ESTJ
21	5	7	ISFJ
22	2	8	ENTJ
23	3	5	ESTJ
24	4	7	INFP
25	1	6	ESTJ
26	5	4	ESFJ
27	6	5	ESTP
28	7	3	ESTJ
29	7	2	ESTJ
30	8	9	ESTJ
31	1	4	ESTJ
32	4	5	ISTJ
33	7	5	ESTJ
34	2	4	ISTJ
35	2	4	ESTJ
36	3	4	ESTJ
37	3	5	ISTJ
38	3	6	INTJ
39	3	6	ENTJ
40	7	7	ESTJ
41	5	4	ESTJ
42	2	2	ENFJ
43	7	5	ENTP
44	4	6	ESTJ
45	1	2	ENFJ
46	2	3	ISFJ
47	4	6	ISTJ
48	8	7	ISTJ
49	2	4	ESFJ
50	8	3	ESTJ

Further Statistical Analysis

Table 5 shows the descriptive statistics from the raw score of the participants using the cognitive styles with high occurrence among the participants. The table shows that ISTJ and ESTJ dichotomies have the same maximum programming result and their inspection skills are roughly the same too. Table 6 shows the rank correlation coefficient between the Programming and Inspection of ISTJ, ESTJ and ESFJ respectively. From the paired samples correlation, it shows that there is a strong positive but statistically insignificant correlation between programming and inspection skills of ESFJ cognitive style while a weak positive correlations were obtained for ISTJ and ESTJ.

Table 5: Descriptive Statistics

	N	Min	Max	Mean	Std. Error
ISTJ Programming	10	2	8	4.30	0.633
ISTJ Inspection	10	3	8	5.50	0.500
ESTJ Programming	19	1	8	4.47	0.521
ESTJ Inspection	19	2	9	4.89	0.411
ESFJ Programming	4	2	7	5.00	1.080
ESFJ Inspection	4	4	6	4.75	0.479

Table 6: Paired Samples Correlations

		N	Correlation	Sig.
Pair 1	ISTJ Programming & ISTJ Inspection	10	0.263	0.463
Pair 2	ESTJ Programming & ESTJ Inspection	19	0.054	0.826
Pair 3	ESFJ Programming & ESFJ Inspection	4	0.806	0.194

Table 7 shows the t-Test which was used to test if there is a significant difference between the programming and code inspection skills of the cognitive styles. From the result, there was no statistical significant difference in the skills within the cognitive styles. There were little differences in the average result obtained from their skills. Table 8 shows the One-Way ANOVA on the raw scores of participants among the cognitive styles obtained. The result obtained shows that there is no significant difference in the programming and inspection skills among the cognitive styles. Significance levels of 0.86 and 0.60 were respectively obtained for programming and inspection at an error probability level of $\alpha = 0.05$. Post Hoc test also confirmed that there was no significant difference among all the cognitive styles.

Table 7: Paired Samples t-Test

		df	Sig. (2-tailed)
Pair 1	ISTJ Programming - ISTJ Inspection	9	0.119
Pair 2	ESTJ Programming - ESTJ Inspection	18	0.523
Pair 3	ESFJ Programming - ESFJ Inspection	3	0.761

Table 8: ANOVA Analysis

		Mean Square	F	Sig.
Programming	Between Groups	0.703	0.148	0.86
	Within Groups	4.761		
Inspection Combined	Between Groups	1.420	0.513	0.60
	Within Groups	2.768		

Discussion

From the results analysis, it can be deduced that the participants with MBTI cognitive style ESFJ have better programming and inspection skill compared with participants of other cognitive styles. The ISTJ, ISFJ, ESTJ and ESFJ all fall under the same category – 'Guardian' according to Keirsey's type name and ESFJs are specially referred to as 'Providers' [22]. According to Briggs, this cognitive style work with determination and like to work with others to complete task accurately and on time [23]. These account for their being good in programming and code inspection.

On a detailed level, it can be said that the Extraversion/Introversion dichotomy of the MBTI has little or no effect on the programming and code inspection skills of the individual. Hence, the Extraversion/ Introversion dichotomy of an individual need not necessarily influence the programming and code inspection skills of an individual. Also, the Thinking/Feeling dichotomy has little effect on the programming and code inspection skills of an individual. Hence, the way in which an individual makes decision need not necessarily influence his programming and code inspection skill. The Sensing/ Intuitive dichotomy on the other hand was shown to have influenced the programming and code inspection skills of some participants. Most of the participants with the Extroversion and Sensing dichotomy performed better in the programming and code inspection assessment than the participants with the Thinking function.

However, statistical test performed on the data shows that all the different cognitive styles obtained in this work with the participants have the same programming and code inspection skill levels and that there is a

strong positive rank correlation between programming and inspection skills of ESFJ cognitive style. This result may be attributed to the fact that the programming and code inspection experience of the participants used were very low. They have not been exposed to the real industrial setting. Natural ability as well as experience to program and inspect code may be some factors that might influence the effectiveness of programmers or code reviewers and not really their personality traits.

Our result is in-line with some existing works. For instance, according to O' Brien et al. [24] in their work on "Myers-Briggs Type Indicator and Academic Achievement in Engineering Education", it was shown that there was no significant difference in dimensions of Extroversion/Introversion, Thinking/ Feeling, Judging/Perception but significant main effect did emerge in relation to sensing/ Intuition. They proposed that students with intuitive cognitive styles achieved significantly higher grades than those with sensing styles.

According to James Miller and Zhichao Yin [12], in their work on "A cognitive-Based Mechanism for Construction Software Inspection Team", the effect of cognitive differences among software reviewers was relatively weak and other confounding variables such as natural ability in finding defects are significantly impacting some of their results. Akinola et al. [10] in their work on "Does cognition affect programming skill?", statistical test performed on the data shows that all the different cognitive styles obtained in their work have the same programming skill level.

Some works (for example, Chandler et al. [25], Capretz [26, 27] and David [15]) have shown that majority of good programmers/ software engineers are always in the ISTJ and ESTJ MBTI dichotomy. Majority of participants in this study also followed this trend as demonstrated in Table 3.

Conclusion

From the results of the experiment, the statistical tests result shows that all the cognitive styles performed at the same level. Also, considering the 4-pair dichotomies, participants in the Sensing dichotomy performed better at programming and code inspection. Further studies would have to be carried out by extending this study to participants in higher levels and in different academic environments as well as in the industry; in which each dichotomy can be empirically looked into, in order to see how it really affects the individual programmer.

Acknowledgement

We specially appreciate the voluntary participation of the 200 Level students of Computer Science, University of Ibadan, Nigeria during the 2011/2012 academic sessions.

References

(1) Nick Bostrom and Anders Sandberg. 2001. Cognitive Enhancement: Methods, Ethics, Regulatory Challenges. *SciEng Ethics* 15: 311-341.

(2) Natalia Stash. 2007. Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System, A Thesis at Technische Universiteit Eindhoven, University Press Facilities, Eindhoven, the Netherlands.

(3) Gideon Vinitzky and David Mazursky. 2011. The Effects of Cognitive Thinking Style and Ambient Scent on Online Consumer Approach Behaviour, Experience Approach Behaviour, and Search Motivation, *Psychology & Marketing*, Vol. 28(5): 496-519.

(4) Roberts, M.J. and Newton, E.J. 2000. Understanding Strategy Selection. *International Journal of Human-Computer Studies*, 54, 137-154.

(5) Programmer *http://en.wikipedia.org/wiki/ Programmer* Visited in September 2012.

(6) Programming Style, *http://en.wikipedia. org/wiki/Programmingstyle* Visited in September 2012.

(7) Tom, O. 1990. Psychology of Programming: Academic press Limited.
(8) Rebecca, M. and Norman, R. 2004. Aspects of Cognitive Style and Programming, Workshop of the Psychology of Programming Interest Group. *Proc. PPIG*, Carlow, Ireland. Pp. 1-9.
(9) Garry, L.W. 2011. Visual Basic Programming Impact on Cognitive Style of College Students: Need for Prerequisites, *Information Systems Education Journal*, 10(4) pp 74-83.
(10) Akinola, O.S., Akinkunmi, B.O. and Abodunde Bukola. 2011. Does cognition affect programming skill? *International Journal of Research and Reviews in Computer Science (IJRRCS)* Vol. 2, No. 4, August 2011, ISSN: 2079-2557.
(11) Garry, L.W. 2005. A Theory of the Relationships between Cognitive Requirements of Computer Programming Languages and Programmers' Cognitive Characteristics. *Journal of Information Systems Education*, Vol. 13(1).
(12) James, M. and Zhichao, Y. 2004. A Cognitive-Based Mechanism for Constructing Software Inspection Teams. *IEEE Transactions on Software Engineering*, Vol. 30, No. 11, pp 811-825.
(13) McMeekin, D.A., von Konsky, B.R., Chang, E. and Cooper, D.J.A. 2008. Measuring Cognition Levels in Collaborative Processes for Software Engineering Code Inspections, proceeding of: IT Revolutions - First International ICST Conference, IT Revolutions 2008, Venice, Italy.
(14) McMeekin, D.A., von Konsky, B.R., Chang, E. and Cooper, D.J.A. 2009. Evaluating Software inspection Cognition Levels Using Bloom's Taxonomy, Proceeding of the 22nd Conference on Software Engineering Education and Training, 2009. CSEET '09.
(15) David Greathead. 2008. MBTI Personality Type and Student Code Comprehension Skill, PPIG, Lancaster 2008, pp. 1-13, *http://www.ppig.org/papers/20th-greathead.pdf* visited in June 2013
(16) Myers, I.B. and McCaulley, M.H. 1985. Manual: A Guide to the Development and Use of Myers-Briggs Type Indicator. Consulting Psychologists Press.
(17) Furnham, C. Jackson and Miller, T. 1999. "Personality, Learning Style and Work Performance," Personality and Individual Differences, Vol. 27, pp. 1113-1122.
(18) Nutt, P.C. 1989. Making Tough Decisions, Jossey-Bass.
(19) Doyle, J., Radzicki, M.J., Rose, A. and Trees, W. 1998. "Using Cognitive Style Topology to Explain Individual Differences in Decision Making," Worcester Polytechnic Inst. Research Report No. 13.
(20) MBTI, *www.myersbriggs.org* Visited in August 2012.
(21) Daniel, V., Luiz, F.C. and Yadenis, P. 2011. Personality types of Cuban software developers, *Global Journal of Engineering Education*, Volume 13, Number 2, pp. 77-81.
(22) Myers-Briggs Type Indicator. *http://en.wikipedia.org/wiki/Myers-Briggs_test#Correlations_to_other_instruments* Visited in September, 2012.
(23) Briggs Myers and Myers, P.B. 1995. Gifts Differing: Understanding Personality Type. Consulting Psychologists Press.
(24) O'Brien, T.P., Leonhard E. Bernold and Duane Akroyd. 1998. "Myers-Briggs Type Indicator and Academic Achievement in Engineering Education" *Int. J. Engineering* Ed. Vol. 14, No. 5, pp. 311-315.
(25) Chandler, J., Carter, J. and Benest, I. 2003. Extravert or Introvert? The Real Personalities of Computing Students. *4th Annual LTSN-ICS Conference*. NUI Galway.
(26) Capretz, L.F. 2003. Personality types in software engineering, *International Journal of Human-Computer Studies* 58(2): 207-214.
(27) Capretz, L.F. 2012. Software Development and Personality Traits, *Tenth International Conference on Computer Applications* (ICCA 2012), University of Computer Studies, Yangon – Myanmar, pp. 1-8.