By ALESSANDRA DEVITO DA CUNHA *and* DAVID GREATHEAD

DOES PERSONALITY MATTER? AN ANALYSIS OF CODE-REVIEW ABILITY

Developing a better understanding of the mental processes leading to success or failure in code review.

There is at least anecdotal evidence for factors greater than 10:1 with regard to ability on programming tasks performed by individuals [2]. In software development it is possible to notice, for example, that some professionals show skill in debugging while others are less successful. Little, however, is known about why there are such large performance variations.

This study, as part of the DIRC project,[1] aimed to investigate if there is a specific personality type that is correlated with performance on a code-review task. The research was partly inspired by several theorists [9, 10, 12] who pointed out that the programming process is comprised of various aspects, such as design and testing, which require different abilities. It was also noted that there were large variations in individual productivity and accuracy as described by Boehm [2]. Pressman [9] and Weinberg [12] reasoned this variation might be due to differences in personality type. To examine this, a study using university second-year undergraduate students was conducted including an analysis of their performance at a code-review task and an assessment of their personality types.

Pressman noticed that programmers with the same background performed differently at the same debugging task [9]. He said there may be some "innate human trait" behind such variation, as there are some programmers who are good at debugging while others

---

[1]DIRC is an Interdisciplinary Research Collaboration on the dependability of computer-based systems funded by the Engineering and Physical Sciences Research Council (EPSRC), the U.K. government's leading funding agency for research and training in engineering and the physical sciences.

In software development it is possible to notice, for example, that **some professionals show skill in debugging while others are less successful.** Little, however, is known about why there are such large performance variations.

are not. Weinberg reasoned this variation in productivity might be related to differences in the type of task. He explained that as each phase of the programming process requires different actions, each action will then benefit from some specific individual skills. Furthermore, Weinberg noted it is possible the performance in the task is a result of the programmers' individual characteristics.

MYERS-BRIGGS TYPE INDICATOR

There have been a number of studies carried out based on individual characteristics influencing performance at work [1, 3, 5, 11]. Individual characteristics can be assessed using any of several personality tests; however, many of the studies carried out within personality and computing science were done with the Myers-Briggs Type Indicator (MBTI).

The MBTI is one of the most widely used personality inventories [1, 11]. It is based on Jung's theory of psychological types and relates to three bipolar factors or dimensions: Extroversion-Introversion (EI), Sensing-Intuition (SN), and Thinking-Feeling (TF). When the MBTI was developed, Myers and Briggs added another pair of characteristics that related to Judgment and Perception (JP).

Capretz [3] analyzed the distribution of personality types of software engineers. He found that MBTI distribution was different among software engineers when compared to the general population, but did not examine if one type performed better than any other.

With regard to personality and career choice in general, the SN and TF dimensions are responsible for attraction to certain jobs [8] while EI and JP determine the attitude people maintain. Myers [6] explained that the EI dimension is concerned with the way people tend to "recharge their energy." Extroverts will focus their attention on other people through the external environment, while introverts will be fully recharged after staying with close friends or family (or by being alone) in an internal environment. This then has an influence on career choice in that this dimension influences people to choose an occupation related to their style. However,

interest in jobs is mainly determined by the SN and TF dimensions.

The SN dimension is related to how people acquire information: whether it is gained through the five senses in a concrete manner or through intuition using imagination and inspiration. The third dimension, TF, is concerned with how people make decisions. The decisions can be made in two ways—either following some logical sequence of facts or making decisions based on a people-centered opinion with more emphasis on feelings rather than logic.

These two dimensions (SN and TF) are responsible for the cognitive aspects that determine how people feel attracted to and are satisfied by their career choices. Both of these factors are also important in solving problems as they help in the improvement of problem-solving skills. It is important to emphasize here that one type is not regarded as being better than another—people simply need to be aware of how to take advantage of their type and work with their preferences and abilities.

A person's lifestyle is determined by the JP dimension. If people are controlled, orderly, and plan everything carefully then they most likely have a judgment orientation. On the other hand, if they are more flexible, spontaneous, and adaptable to new situations their lifestyle is more orientated toward perception. In a person's career, judgment and perception influence how they work.

PRACTICAL APPROACH

Personality was measured using the MBTI, which returns four letters to indicate the type of preference. The code-review task consisted of 282 lines of Java (in which there were 16 semantic bugs), which was accompanied by a two-page manual and API, including an example of the output the bug-free code would produce when executed. All 64 participants were second-year undergraduate students from Newcastle University in the U.K. The students, 52 males and 12 females, were all taking part in a computing module. Prizes were awarded for the best performance at the code-review task.

RESULTS

For the purpose of analysis, the various bugs were weighted according to their difficulty. We chose to categorize the bugs as easy, medium, and hard. In the end, there were five easy, four medium, and seven hard bugs, and this difference was taken into consideration when calculating the final scores for the code-review task.

In order to examine the possible links with MBTI type and code-review ability, a number of correlations were computed between the task score and each bipolar factor (EI, SN, TF, and JP). It was found that the only single bipolar factor that presented a significant correlation (a value of $p < 0.05$) with the performance in the task was between the Sensing-Intuition factor and the code-review score. It was found that people who were more intuitively inclined performed significantly better on the code-review task than sensing types.

	F	T
N	8.71	9.10
S	4.27	6.62

Mean code review score by SN/TF types.

As there were insufficient participants to conduct analysis based on their full MBTI type (for example, INTJ), it was decided to use a combination of two letters. It was decided that the SN and TF dimensions would be most interesting due to the relationship with problem solving and career choice. By using these two scales, scores could be considered on a four-group basis. Examining the mean code-review scores shows that the NT students scored 9.10 as compared to the non-NTs who scored 6.14 on average. This illustrates that, with this sample, the NT individuals were better at the code-review task than the non-NT people. The mean scores for these four types are included in the table here. As shown in the table, the most marked difference was between NT participants and SF participants, the SFs achieving on average less than half the score of the NT participants. A statistical test comparing NTs with non-NTs yielded a significant result, illustrating that NTs were better at the task than non-NTs.

DISCUSSION

NTs perceive the world through their intuition, that is, they gain their insights by meanings and potentials rather than by observing facts [7]. They also make their judgments by thinking, that is, by logical connections of the facts rather than by weighing the decisions. They are always "looking at the possibilities, theoretical relationships, and abstract patterns" [7]. The way in which they make their judgments is impersonal. NTs are known as "logical and ingenious" and they are best at solving problems within their field of special interest. It is therefore not surprising that NT programmers are better at finding bugs in programs.

As can be observed from the results, it is not merely the case that in this sample the NTs performed better at the task than the other letter types, but that the difference was so marked between NTs and their opposite type, SFs, who scored less than half as well as the NTs on this task. It is not the authors' intent to suggest that SFs be precluded from performing code-review tasks. However, it is worth remembering that this difference could be accounted for in the way NTs look at the world. If this is the case, then it would be advantageous to encourage code reviewers in general to be aware of the way NTs think as this may aid them in code review [4]. As Myers and McCaulley [7] state, people can generally act in a way that is not congruent with their type, even if it would not be their first choice in the everyday world.

It could simply be that NT people enjoy code-review type tasks more than non-NT people, and consequently in our study applied themselves more to the task at hand. This being the case, their performance may simply be a result of motivation. While a prize was awarded to the participants for performance on the task, this may not have been sufficient encouragement for unenthusiastic participants to apply themselves for an hour when compared with someone who enjoyed the task. A similar experiment could be carried out with different forms of motivation to examine the possibility of this influence.

It can be reasoned that as NTs present particular qualities that contribute to the success in the code-review task, there are some types that may be linked to success in other phases of the programming process. As Weinberg indicates, each part of the programming process has a specific task to be completed that requires different "combinations of skills and personality traits" [12]. Thus, it would be useful to look at the other stages of the programming process and the personality types involved. This is one particular avenue of future research.

It could also be true that the reason the NTs performed better than the other types on this task was something other than the fact it was a code-review task. One possibility is there was something in the nature of the program itself, or the type of bugs presented that aided the NTs in the task. In order to examine this, the authors are in the process of replicating the study, substituting the program used with another one, of a different type, written by a different programmer.

A further possibility is there is some unknown cog-

If a company organizes its employees according to their personality types and their potential abilities, **productivity and quality may be improved.**

nitive dimension that influences performance on the code-review task, which is correlated with the NT score. This would mean the MBTI type is indirectly rather than directly linked to performance on this task. If this is the case, then further research into alternative measures and tasks would begin to shed some light on this possibility.

It would also be advantageous to examine performance on code review and other tasks when considering the full, four-letter types (such as INTJ). This was not statistically practical with the current study due to the total sample size of 64. While at least one of each MBTI type was obtained within this sample, some of the types had few representations, thus making most statistical analysis inappropriate. In order, therefore, to examine the full, four-letter types, further research is required to increase the sample size. Despite the low sample size for this type of analysis, it is interesting to note how the results differ from those discussed by Capretz [3] in that, while Capretz reported 24% of participants as being ISTJ, the current study only reported 6% in that group (four participants). The largest number of participants of any one type was for the ENTP type, there being 10 participants (16%). It would be interesting to examine this ratio with a greater number of participants, and also with a sample of professional programmers.

CONCLUSION

It may be advantageous for software organizations to consider the strengths of their employees when assigning them tasks in the workplace. If some people, for whatever reason, are better able to perform code-review tasks than others then it would be prudent for software companies to capitalize on the strengths of their employees, and consider employees perhaps previously overlooked for this particular task. If a company organizes its employees according to their personality types and their potential abilities, productivity and quality may be improved [4].

While it may be the case that MBTI type may never be as clear a predictor of code-review ability than, say, another code-review task, it seems likely that there are more influences at work here than are immediately obvious. This research is the first step toward developing a greater understanding of the mental processes involved in code review (and ultimately other aspects of software engineering) that in turn would lead to greater dependability of computer-based systems if the processes leading to success or failure in code review are better understood and considered. **C**

REFERENCES
1. Bishop-Clark, C. and Wheeler, D. The Myers-Briggs personality type and its relationship to computer programming. *Journal of Research on Computing Education 26*, 3 (1994), 358–370.
2. Boehm, B.W. *Software Engineering Economics*. Prentice-Hall advances in computing science and technology series. Prentice-Hall, Englewood Cliffs, NJ, 1981.
3. Capretz, L.F. Personality types in software engineering. *International Journal Human-Computer Studies 58* (2003), 207–214.
4. Devito Da Cunha, A. The Myers-Briggs personality type as a predictor of success in the code review task. MPhil Dissertation, University of Newcastle, 2003.
5. Furnham, A., et al. Do personality factors predict job satisfaction? *Personality and Individual Differences 33*, 8 (2002), 1325–1342.
6. Myers, I.B. *Introduction to Type: A Description of the Theory and Applications of the Myers-Briggs Type Indicator, 12th edition*. Consulting Psychologists Press, Palo Alto, CA, 1990.
7. Myers, I.B. and McCaulley, M.H. *Manual: A Guide to the Development and Use of the Myers-Briggs Type Indicator, 5th edition*. Consulting Psychologists Press, Palo Alto, CA, 1985.
8. Myers, I.B. and Myers, P.B. *Gifts Differing: Understanding Personality Type*. Davies-Black Publishing, Palo Alto, CA, 1995.
9. Pressman, R.S. Chapter 19: Software testing strategies, in *Software Engineering: A Practitioner's Approach*. McGraw-Hill, New York, 1992, 654–658.
10. Shneiderman, B. *Software Psychology: Human Factors in Computer and Information Systems*. Winthrop Publishers, Cambridge, MA, 1980, 320.
11. Smither, R.D. *The Psychology of Work and Human Performance, 3rd Edition*. Longman, New York, 1998, 590.
12. Weinberg, G.M. *The Psychology of Computer Programming, 2nd Edition*, Van Nostrand Reinhold, New York, 1998, 288.

ALESSANDRA DEVITO DA CUNHA (a.devito-cunha@ncl.ac.uk) is a research associate at the Centre for Software Reliability at the University of Newcastle upon Tyne, Newcastle upon Tyne, U.K.
DAVID GREATHEAD (david.greathead@ncl.ac.uk) is a research associate at the Centre for Software Reliability at the University of Newcastle upon Tyne, Newcastle upon Tyne, U.K.