






Comparing Participants' Brainwaves During Solo, Pair, and Mob Programming

Makoto Shiraishi^(✉), Hironori Washizaki^(✉) , Daisuke Saito^(✉) ,
and Yoshiaki Fukazawa^(✉) 

Waseda University, Okubo, Shinjuku City, Tokyo, Japan
{makwhitestone,d.saito}@fuji.waseda.jp,
{washizaki,fukazawa}@waseda.jp

Abstract. Participants' feelings and impressions utilizing electroencephalography (EEG) and the effectiveness of code are compared for different types of programming sessions. EEG information is obtained as an alternate viewpoint during three programming sessions (solo, pair, and mob programming). Mind-Wave Mobile 2 (brainwave detector) is equipped to collect the attention levels, meditation levels, and EEG brainwaves. These data are utilized to distinguish efficiencies, weaknesses, and points of interest by programming session. The results provide preliminary information to distinguish between the three sessions, but further studies are necessary to make firm conclusions. Additionally, alternative methods or systems are required to analyze the collected data.

Keywords: Mob programming · Pair programming · Programming · Brainwaves · EEG · Meditation · Attention

1 Introduction

Mob programming is becoming more prevalent, especially in the field of agile development. Mob programming is a method to learn programming in a group. Although it is becoming popular, its effects and limitations remain unclear.

Herein analysis is conducted from a different perspective: EEG (electroencephalography). This new information is used to analyze participants' EEG brainwaves to identify efficiencies, shortcomings, and advantages of different types of programming sessions (solo, pair, mob programming). Pair programming is a method to learn programming with another individual, it has similar concepts to mob programming. Here, four hypotheses (RQs) are investigated.

- RQ1: Do pair and mob programming induce lower levels of concentration (attention) compared to solo programming?

Often tasks requiring high concentration levels are performed individually, suggesting that sessions involving multiple people will lower concentration levels. Working alone may be better for tasks that require high fixation and core interest. An investigation known as the Coding War Games found that software engineers worked more efficiently alone [2]. Another benefit of working alone is that the individual may become completely immersed in their work. However, a drawback is that the person is solely responsible for the performance and output. In contrast, people working in a group may find motivation and inspiration throughout the whole session because it is the group's responsibility to produce results and there is peer pressure to not disappoint the group. This research question examines the difference in attentiveness in groups vs. individual programming work.

- RQ2: Do group sessions induce higher levels of calmness (meditation) compared to solo programming?

The psychological benefits of collaborative learning are increased participants' self-esteem and reduced anxiety [3]. Thus, we approximate that group work is calmer than solo programming sessions. Typically, participants working in a larger group are more comfortable with an unfamiliar problem because other people are around to assist. They are more mentally assured and less worried about solving the task because someone else in the group may know the answer. It is important to understand the difference in calmness in these sessions to be able to create an environment that can be continued for a prolonged/maintained period of time. As an increase in calmness/meditation can lead to less tension and stress.

- RQ3: Does difficulty decrease when working in a group?

Previous research found that groups of three to five individuals perform best when solving complex problems, suggesting that groups are better suited than those working alone to address challenging problems [4]. Additionally, those in a group tend to retain more information, especially since the group can work together and share knowledge. Finally, working with others provides new insight and perspectives. If a significant difference can be seen within individual/group size, we will be able to assign tasks to groups or individuals according to the difficulty of the task. Thus, making it much easier for participants to program a task.

Here, task difficulty is measured using EEG brainwaves such as the frequency of low β and low α . An increase of $low\beta/low\alpha$ denotes an increasing difficulty [5].

- RQ4: What are the significant differences between pair and mob programming sessions?

Although solo and group work should have clear differences, the impact of group work in pairs or a larger group may also result in differences in attention and concentration. This question compares group work in teams of two to that in larger groups. It is possible that working in a larger group will decrease how the participants feel on the difficulty of solving problems/tasks.

A comparison between the roles (driver, navigator) is necessary to compare pair and mob programming. In both sessions, the driver's attention level and the navigators' meditation levels should be highest because the driver will be coding and perhaps more focused, while the navigator is more relaxed as no coding is actually done.

2 Background

Mob programming is an agile software development practice where a group works together on the same screen and changes roles at set time intervals. This approach began with Woody Zuill's programming ventures at Hunter Industries [6]. Assigned roles assigned the productivity and efficiency. During a mob programming session, one software engineer (the driver) codes for the entire group by utilizing a console and mouse. Other software engineers (navigators) audit the code and ensure that the driver's work is free from logical or syntactical errors. Mob programming sessions aim to complete specific programming improvement assignments while working together with the client [7]. Advantages include an organized cycle for utilizing distributed information and a greater code fulfillment due to the inherent sharing of information and experience among the participants [8].

A systematic literature review (SLR) revealed [14] that (1) experimentation with controls is vital and (2) most mob programming sessions are finished during working hours, but the eventual outcomes are rarely investigated. Participant surveys show definite impacts such as improved code quality and expanded profitability, but the results are subjective and lack scientific evidence. Logical proof from an expert's viewpoint confirms that mob and pair programming have logical confirmation [9–12]. Performing coding tests (which have been recognized by experts) during mob programming sessions can monitor the effects on code quality. Furthermore, the influence of the mob programming session length on the coding speed should be investigated.

The SLR highlights the importance of investigating and analyzing the impacts of mob programming. This study uses a new perspective (brainwaves) to discover empirical evidence.

3 Collection and Analysis of Brainwaves

In this study, we employed a new method, which is similar to a preliminary study of pair programming on brainwaves [13], to collect empirical evidence and answer our four RQs. Participants' brainwaves were used to understand and compare the mental state from a scientific viewpoint. We collected three main datasets: attention, meditation, and difficulty. We compared the numerical values of how the participants mentally feel in each programming session. Surveys, code testing time, code quality, and personal options are conventionally used to analyze mob programming. However, these methods do not provide clear logical proof or analysis. On the other hand, our new method provides a clear psychological view. Although this study could simply investigate the differences in code testing time and code quality after several mob programming sessions, herein we assess mental states using attention, meditation, and difficulty values.

A brainwave detector, called the MindWave Mobile 2, was used, and NeuroView software collected and analyzed the EEG.

MindWave Mobile 2 (Brainwave Device). The device measures and outputs the EEG (0–127.75 Hz) power spectra (alpha, beta waves, etc.). Additionally, it monitors attention and meditation. The device consists of a headset, an ear-clip, and a sensor arm [1].

EEG. EEG was performed to assess the electrical activity of the brain. Brain cells communicate with each other via electrical impulses. The MindWave Mobile 2 tracks and records brainwave patterns. Small flat metal discs called electrodes are attached to the scalp with wires. The electrodes analyze the electrical impulses in the brain and send signals to a computer that records the results [1].

Attention. The eSense Attention meter (0–100) indicates the intensity of a user's level of mental "focus" or "attention" during intense concentration or directed mental activity. Distractions, wandering thoughts, lack of focus, or anxiety lowers the meter level. Thus, the concentration levels during programming sessions can be measured [1].

Meditation. The eSense Meditation meter (0–100) indicates the level of a user's mental "calmness" or "relaxation." Meditation is a measure of a person's mental state not physical levels. Meditation is related to reduced activity for active mental processes in the brain. Distractions, wandering thoughts, anxiety, agitation, and sensory stimuli lower the meditation meter levels [1].

Difficulty. Difficulty felt by participants during a certain task can be calculated using

$$\text{low } \beta \text{ waves} / \text{low } \alpha \text{ waves} \quad (1)$$

collected from the MindWave Mobile 2 [5]. A previous study involving ten high school students used a typing practice with two difficulty levels (basic & advanced). The study proposed that **an increase of ($\text{low } \beta / \text{low } \alpha$) indicates a higher difficulty** (low β wave (13–16.75 Hz) and low α wave (7.5–9.25 Hz)) [5]. In this paper, we used this approach to evaluate the third RQ.

4 Experimental Method

We evaluated three types of sessions: solo, pair, and mob. Nine subjects (E1–E9), who were university or graduate students over 18 years old and with a certain level of programming skills (i.e., completed basic programming), participated in each experiment. Each experiment was repeated twice. The flow of the experiment was as follows:

1. Subjects were divided into six groups: 3 Solo groups, 3 Pair groups and 3 Mob groups. Each participant was involved in two sessions: Solo & Mob or Pair & Mob (Table 1).

Table 1. Participant overview

Session	E1	E2	E3	E4	E5	E6	E7	E8	E9
Solo programming	Sa			Sb			Sc		
Pair programming		Pa1	Pa2		Pb1	Pb2		Pc1	Pc2
Mob programming	Ma1	Ma2	Ma3	Mb1	Mb2	Mb3	Mc1	Mc2	Mc3

2. We asked the participants to wear an EEG on their heads and to meditate for 1 min to obtain EEG data during a neutral state. That is, we asked the participants to close their eyes and relax. This data was used to compare with the EEG data obtained during the programming period.
3. The two sessions were conducted consecutively with a 15-min break in between them. Each group was asked to answer up to three programming questions with varying degrees of difficulty.

Example Question: Given an integer number n , return the difference between the product of its digits and the sum of its digits. **Input:** $n = 234$ **Output:** 15 **Explanation:** Product of digits = $2 * 3 * 4 = 24$ Sum of digits = $2 + 3 + 4 = 9$, Result = $24 - 9 = 15$

4. The time limit was 30 min. Subjects were not required to finish questions. A new question was given once the previous question was completed. EEG brainwaves were recorded using the MindWave Mobile 2 while the subjects answered the questions.
5. For the Solo programming session, the participant coded alone, and was not allowed to communicate with anybody. However, for pair and mob programming sessions, participants could communicate with group members. For pair and mob programming sessions, there were two roles (driver and navigator). There was only one driver at a time, and the other participants were navigators. The participants switched roles after 15 min for pair programming and 10 min for mob programming sessions. Each participant performed both roles. The groups were assigned as shown in Table 1.

5 Evaluation of Results

The experiments assessed the RQs. Figures 1–2 show the data to answer the first two research questions, while Fig. 3 shows that for the third research question.

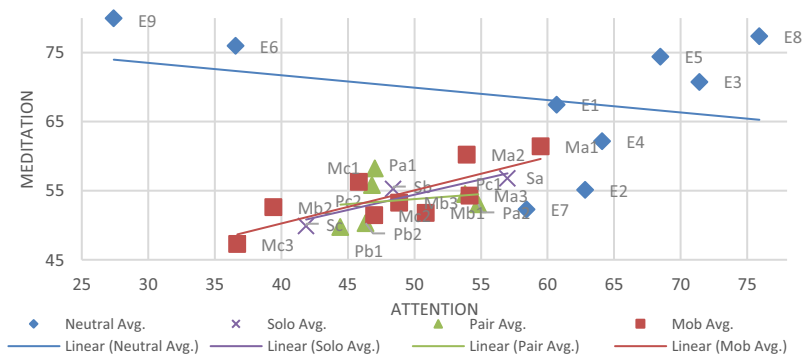


Fig. 1. Comparison of meditation and attention levels by programming state.

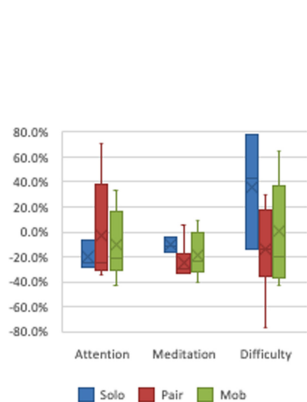


Fig. 2. Box and whisker of participants' aver-age percentage in change of attention, meditation, and difficulty from the neutral state by programming session.

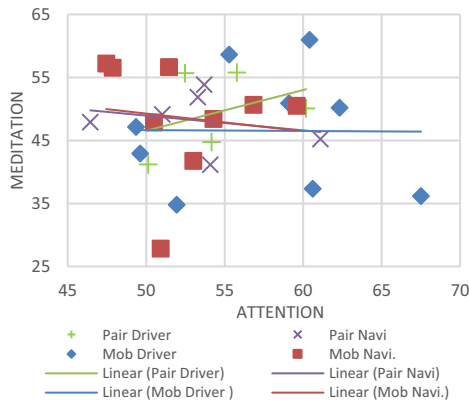


Fig. 3. Attention vs. meditation for participants in driver & navigator roles during pair and mob programming

5.1 RQ1: Do Pair and Mob Programming Induce Lower Levels of Concentration (Attention) Compared to Solo Programming?

In many situations, tasks requiring high concentration are often worked on alone. In Fig. 1, concentration (attention) levels were highest in the neutral state. Therefore, the average percentage change in each programming session (Fig. 2) decreased concentration. Pair programming showed the highest percentage of attention level change. However, both group programming sessions (pair and mob) had higher concentration levels than the individual programming session.

RQ1 Conclusion: Pair and mob programming induced **higher** levels of **concentration** compared to solo programming. It is noteworthy that solo programming had the lowest attention levels. Group work, especially pair programming, had the highest attention levels.

5.2 RQ2: Do Group Sessions Induce Higher Levels of Calmness (Meditation) Compared to Solo Programming?

In Fig. 1, all participants except two (E2, E7) showed the highest meditation level in the neutral state. Overall, a decrease was observed in the average for all three types of programming sessions. The highest average meditation level was solo, and the lowest was pair programming. Their difference was significant (around 13%). This result differs from a previous study [13], where a more relaxed state was observed in pair programming than solo programming.

RQ2 Conclusion: Pair and mob programming sessions induced **lower** levels of **meditation** compared to Solo sessions. The highest meditation level was during a solo session, and the lowest was in pair programming.

5.3 RQ3: Does Difficulty Decrease When Working in a Group?

The questions in each session had the same level of difficulty. However, differences in difficulty are observed by session type. The lower the difficulty value, the easier a task feels. Figure 2 shows the average change in difficulty by group relative to the neutral state. The solo experiment showed a 36% increase in difficulty, whereas the mob programming group had a 1% increase in difficulty compared to the neutral state. On the other hand, pair programming showed a decreased difficulty of around -14%. These results imply that working in a group setting is less difficult than working alone.

RQ3 Conclusion: Participants' difficulty **decreased** as the group became larger. The difference between working in a group and working alone was significant.

5.4 RQ4: Are there the Significant Differences Between Pair and Mob Programming Sessions?

Attention and Meditation. Both pair and mob programming sessions were positively correlated with attention and meditation, but the correlation was not significant (Fig. 1). **Only pair programming had a positive percent change** compared with the neutral value (Table 2). There was a somewhat **positive correlation (Pearson's Correlation**

Table 2. Comparison of the average percentage change from neutral state

Role	Attention (Rank)	Meditation (Rank)	Difficulty (Rank)
Pair Driver	2.4% (1)	−22.9% (3)	−49% (4)
Pair Navigator	4.6% (2)	−24.9% (4)	−44% (3)
Mob Driver	−4.9% (3)	−4.9% (1)	−4% (1)
Mob Navigator	−17.6% (4)	−21.4% (2)	−5% (2)

Coefficient [15]) **only for the driver in pair programming** (Fig. 3), but the rest (pair navigator and mob) had a negative or close to zero correlation. For **mob programming**, a **significant difference** was observed **between the roles** (driver and navigator). The attention and meditation levels decreased **more than 10%** from the **navigator to the driver**, suggesting that participants concentrate less and are more relaxed when they are navigators.

Difficulty. Participants had an easier time in a group than when working alone (Fig. 2). Pair programming and mob programming showed a significant difference. Participants in pair programming found it easier to tackle problems, but there was not a significant difference between the roles (Table 2).

RQ4 Conclusion: Our small experiment suggested that pair programming is more beneficial for participants. The largest factor was the difference in the roles during a programming session. The difference by role was smaller in the pair session, indicating that both roles were exerting similar amounts of effort.

6 Discussion

Four RQs were examined to obtain a deeper knowledge of mob programming and pair programming compared to solo programming. MindWave Mobile 2 was used to collect brainwave data from a new research perspective. The results indicate that group programming can realize more efficient learning than individual programming. As shown in the results, the meditation levels were lowest in solo work which suggests working alone may lead to less stress and tension compared to group work. Additional research is necessary to identify patterns. First, a larger dataset is necessary since this study involved only six participants. Due to time limitations, two programming sessions were conducted in a row. If the programming sessions occurred on different days, the participants would be more refreshed, which may affect the concentration and meditation results. Second, the EEG results have yet to be fully analyzed. The results here should be compared with other brainwave data.

7 Conclusion and Future Work

We preliminarily identify the differences between the brainwaves of programmers under different sessions (solo, pair, and mob). MindWave Mobile2 collected the levels of meditation, concentration, and EEG signals. Although differences between the participants in each session were distinguished, it is too early to yield conclusions. It is unclear whether the apparent differences between these programming methods are sustained by scientific and empirical evidence. Consequently, future research should investigate appropriate systems to look at brainwaves and summarize noteworthy quantities of data.

References

1. EEG - Ecg – Biosensors. <http://neurosky.com/>
2. DeMarco, T., Lister, T.R.: *Peopleware: Productive Projects and Teams*. Addison-Wesley, Upper Saddle River (2014)
3. Laal, M., Ghodsi, S.M.: Benefits of collaborative learning. *Procedia. Soc. Behav. Sci.* **31**, 486–490 (2012). <https://doi.org/10.1016/j.sbspro.2011.12.091>
4. Laughlin, P.R., Hatch, E.C., Silver, J.S., Boh, L.: Groups perform better than the best individuals on letters-to-numbers problems: effects of group size. *J. Pers. Soc. Psychol.* **90**(4), 644–651 (2006). <https://doi.org/10.1037/0022-3514.90.4.644>
5. Umezawa, K., Saito, T., Ishida, T., Nakazawa, M., Hirasawa, S.: An Electroencephalograph-Based Method for Judging the Difficulty of a Task Given to a Learner. In: 2017 IEEE 17th ICALT (2017). <https://doi.org/10.1109/icalt.2017.18>
6. Zuill, W.: Mob Programming - A Whole Team Approach. Agile Alliance (13 June 2019). <https://www.agilealliance.org/resources/experience-reports/mob-programming-agile2014/>
7. Kattan, H.M., Soares, F., Goldman, A., Deboni, E., Guerra, E.: Swarm or pair? In: *Proceedings of the 19th International Conference on Agile Software Development: Companion* (2018). <https://doi.org/10.1145/3234152.3234169>
8. Kattan, H.M., Oliveira, F., Goldman, A., Yoder, J.W.: Mob programming: the state of the art and three case studies of open source software. In: Santos, V.A.D., Pinto, G.H.L., Serra Seca Neto, A.G. (eds.) *WBMA 2017. CCIS*, vol. 802, pp. 146–160. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73673-0_12
9. Dybå, T., Arisholm, E., Sjöberg, D.I.K., Hannay, J.E., Shull, F.: Are two heads better than one? on the effectiveness of pair programming. *IEEE Softw.* **24**(6), 12–15 (2007). <https://doi.org/10.1109/ms.2007.158>
10. Hanks, B.: Empirical evaluation of distributed pair programming. *IJHCS* **66**(7), 530–544 (2008). <https://doi.org/10.1016/j.ijhcs.2007.10.003>
11. Radermacher, A., Walia, G.: Investigating student-instructor interactions when using pair programming: An empirical study. In: 2011 24th IEEE-CS CSEE&T (2011). <https://doi.org/10.1109/cseet.2011.5876117>
12. Salleh, N., Mendes, E., Grundy, J.: Empirical studies of pair programming for CS/SE teaching in higher education: a systematic literature review. *IEEE Trans. Software Eng.* **37**(4), 509–525 (2011). <https://doi.org/10.1109/tse.2010.59>
13. Busechian, S., et al.: Understanding the impact of pair programming on the minds of developers. In: *Proceedings of the 40th ICSE: NIER* (2018). <https://doi.org/10.1145/3183399.3183413>

14. Shiraishi, M., Washizaki, H., Fukazawa, Y., Yoder, J.: Mob programming: a systematic literature review. In: 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC) (2019). <https://doi.org/10.1109/compsac.2019.10276>
15. Pearson's Correlation Coefficient. (n.d.). Springer Reference. https://doi.org/10.1007/springerreference_83937

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

