

Paper Critique

Zuxing Wu, a1816653

1. Only one programming task

Description: There is no evidence that the findings would generalise to programming tasks other than JSON parser construction.

Mitigation strategy: This could be mitigated by including a variety of programming tasks in the experiment.

2. No clear date about the programming language ranking

Description: It states JavaScript and Java are the two most popular programming languages, but no clear date is provided. Besides, the current popularity of programming languages have changed to JavaScript and Python, as of 14 March, 2025.

Mitigation strategy: This could be mitigated by providing the date when the experiment was conducted.

3. Programming languages are not the only factor affects productivity

Description: The paper assumes one of the two most popular programming languages makes developers more productive. However, there are many other factors that could affect developers' productivity, such as the complexity of the task, the experience of the developers, and the tools (e.g. IDE, laptop) they use.

Mitigation strategy: This could be mitigated by controlling these factors in the same group of developers.

4. It is not representative enough

Description: Insights from this experiment have no clear potential and supportive evidence to guide future programming language choice for developers.

Mitigation strategy: This could be mitigated by conducting a more comprehensive experiment with a larger sample size and more programming tasks.

5. Small sample size

Description: The sample size of the experiment is small. They only recruited 5 participants.

Mitigation strategy: This could be mitigated by increasing the sample size. At least 30 participants are recommended.

6. Sample selection problem

Description: The participants are all students from an introductory programming course teaching JavaScript. They are highly likely to be not productive at all in JavaScript, not to mention Java.

Mitigation strategy: This could be mitigated by recruiting participants with at least 1 year of experience in both JavaScript and Java.

7. Why Java first, then JavaScript?

Description: The participants were required to complete the Java task first, then the JavaScript task. This could affect the results, as the participants might be tired after completing the Java task. Besides, the participants might have learned from the Java task and applied the knowledge to the JavaScript task.

Mitigation strategy: This could be mitigated by randomising the order of the tasks for each participant.

8. Coding environment is not reasonable

Description: participants were required to work with a command-line compiler and a basic text editor without syntax highlighting and content assist. Participants were not allowed to use the Internet while working on their tasks. It is not a common coding environment for developers.

Mitigation strategy: This could be mitigated by providing a more common coding environment for developers, such as an IDE with syntax highlighting and content assist. Internet access should also be allowed for developers to search for information, or look up the documentation.

9. Wrong method of calculating productivity

Description: The paper calculates productivity by the number of lines of code written per minute, which is not appropriate because these two languages have different syntax. However, they also include whitespaces, which is not reasonable.

Mitigation strategy: This could be mitigated by only counting the number of lines of code excluding whitespaces and comments.

10. Groundless assumption

Description: They assumed a task completion of 30 minutes for participants who did not completed either these tasks. This is groundless.

Mitigation strategy: This could be mitigated by setting a reasonable time limit for the tasks. If the participants did not finish the task within the time limit, they should be used to calculate productivity. This also shows the necessity of recruiting enough experienced participants so that they can finish the tasks within the time limit, rather than assuming a task completion of 30 minutes.

11. Unconvincing conclusion & recommendation

Description: They conclude that JavaScript is more productive than Java. However, the evidence is not convincing enough. They also cannot recommend software development teams to use JavaScript instead of Java without considering other factors.

Mitigation strategy: This could be mitigated by conducting a more comprehensive experiment with a larger sample size and more programming tasks, using more programming languages. They should also consider other factors that could affect developers' productivity. The language choice is not only decided by productivity, but also by the project type, technology feasibility, client requirements, and the experience of the developers, etc.