

metacodenition

PROJECT #127: A PEDAGOGIC IDE

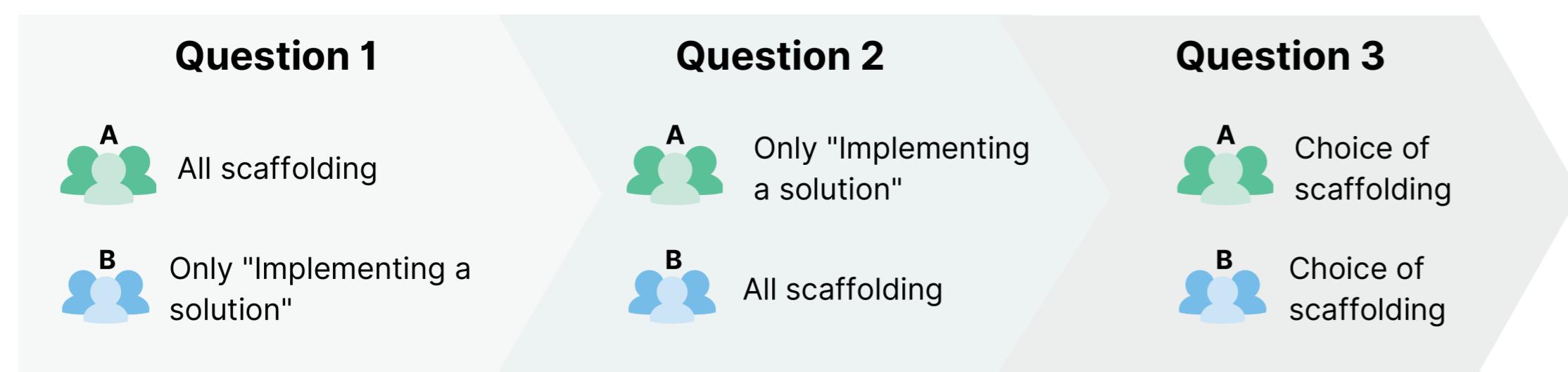
INTRODUCTION

Problem-solving is widely regarded as the most essential engineering skill. Most introductory programming courses focus on teaching the syntax and semantics of programming, but place relatively little emphasis on teaching problem-solving skills.

Existing problem-solving frameworks can be hard to teach as they require students to develop metacognitive awareness ("thinking about thinking"). Our tool, Metacodenition, provides metacognitive scaffolding for one of these frameworks while a student is solving a problem.

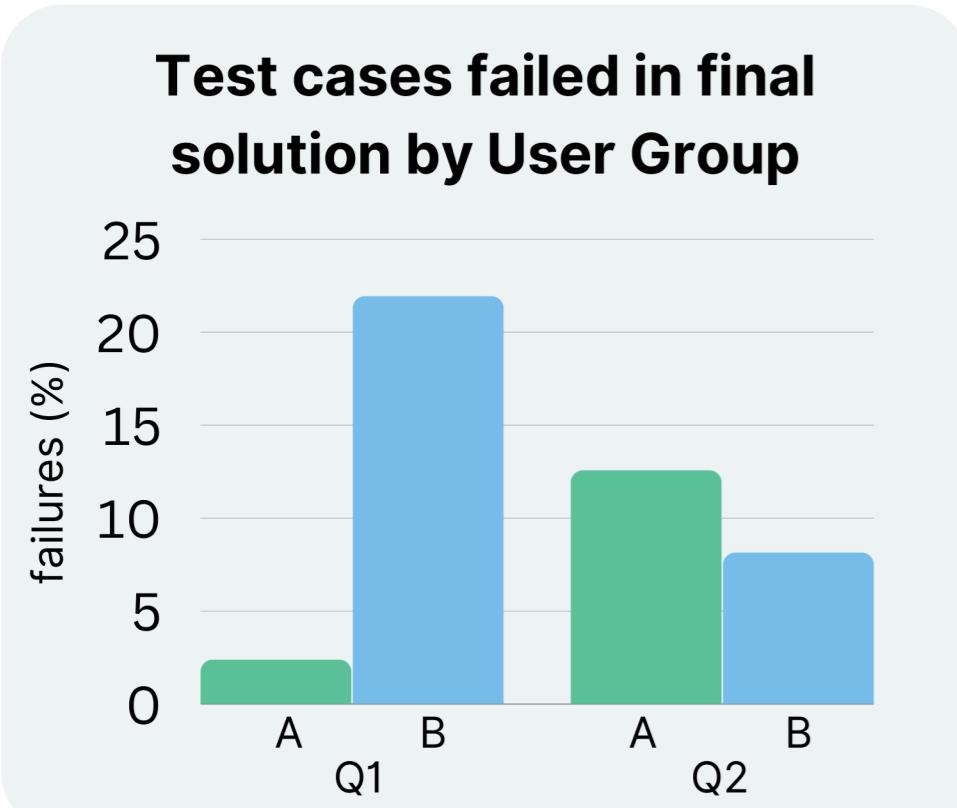
EVALUATION

- **Graded lab** in a **first-year** introductory programming course
- **Three questions** involving loops and arrays
- **Two user groups** - Group A and Group B
- **Optional questionnaire**



RESULTS

821 first-year engineering students used Metacodenition, of which 427 answered the optional feedback questionnaire.



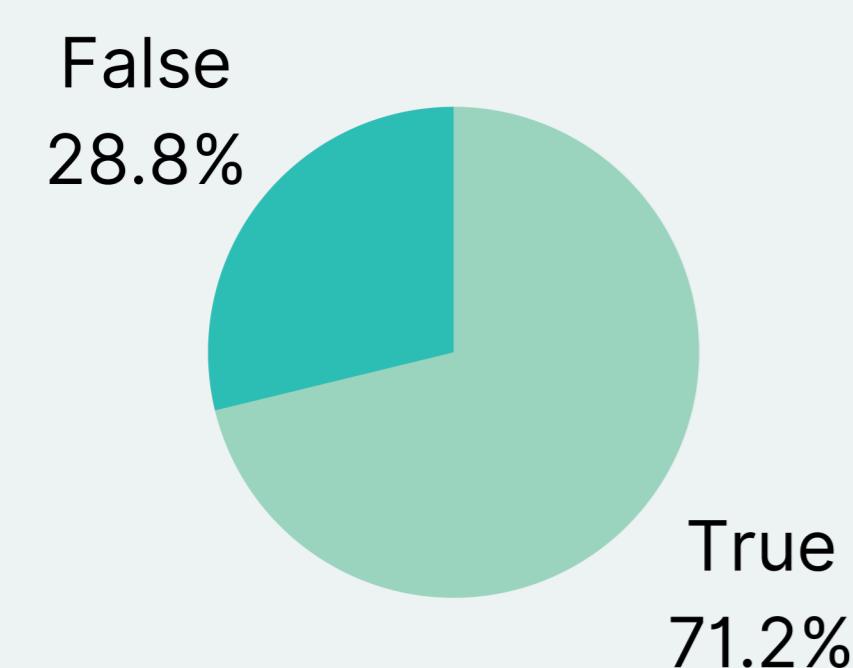
"I liked the test cases and the highlighting of the question sections. I found I fully understood all the edge cases and sentences in the question that I would typically miss."

"I really liked how I could write out a solution plan which outlined what I needed to do, and how it remained visible as I was writing my code."

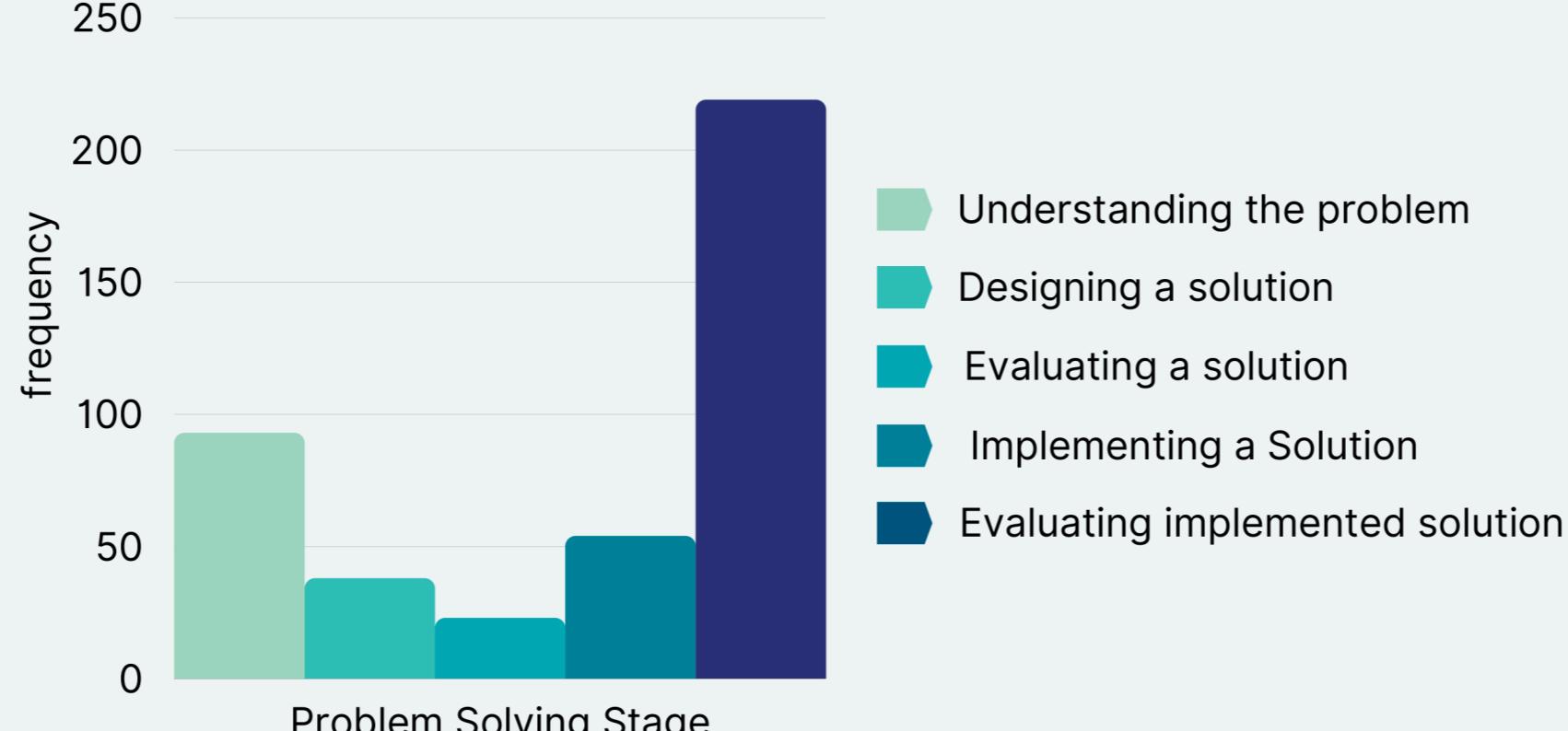
"Understanding the problem was a huge help. Not only did it allow for easy testing later but often there is a bit of ambiguity of what to do in a specific case so it's really nice to have that sorted before I started coding."

-Anonymous Students

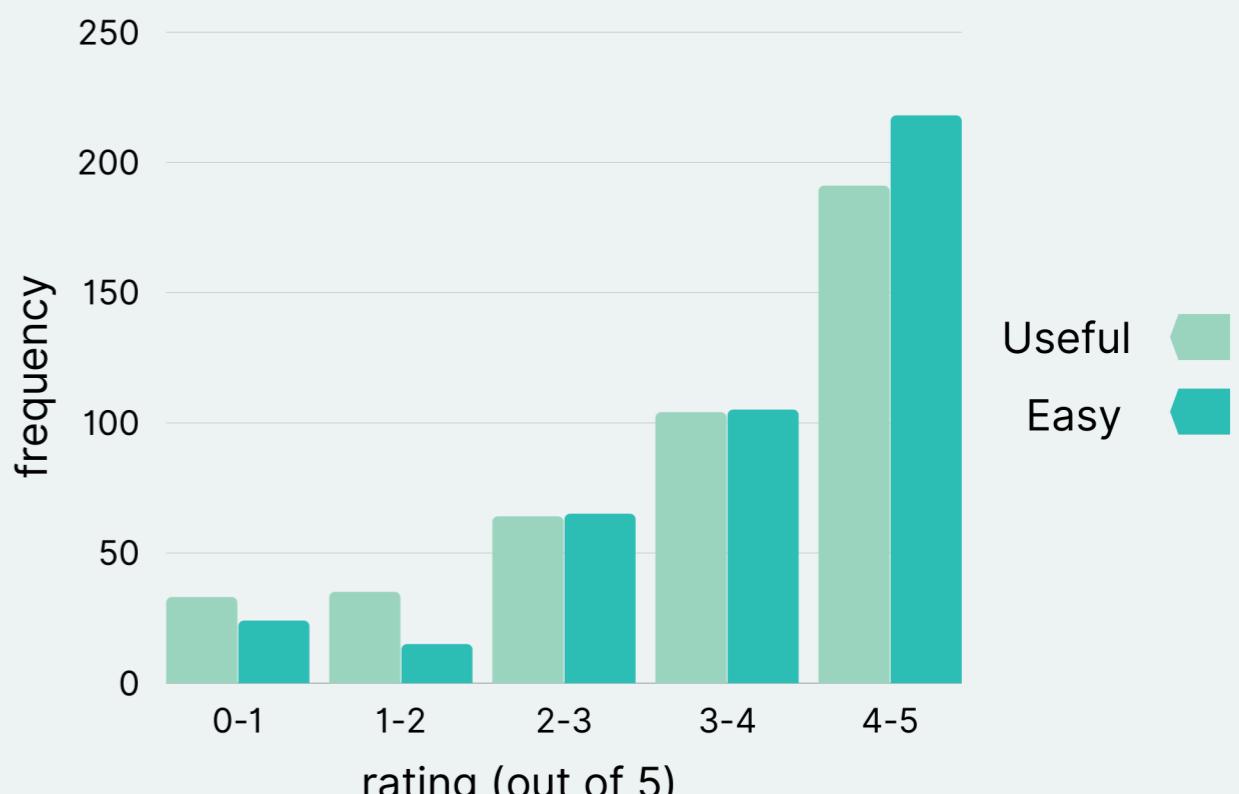
Responses to "I would use this tool again if it was optional."



Responses to "Which assistance was the most helpful?"



Responses to "How useful/easy was this tool?"



CONCLUSION

- Our evaluation indicated that the majority of students would use Metacodenition again if it was optional and perceived Metacodenition as a useful and easy-to-use tool.
- Students were more likely to submit a correct solution when given scaffolding.

FUTURE WORK

- Implement additional problem-solving scaffolding.
- Add support for language servers, more readable error messages.

REFERENCES

- [1] D. Loksa, A. J. Ko, W. Jernigan, A. Oleson, C. J. Mendez, and M. M. Burnett, "Programming, Problem Solving, and Self-Awareness: Effects of Explicit Guidance," presented at the Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, California, USA, 2016. [Online]. Available: <https://doi.org/10.1145/2858036.2858252>.
- [2] P. Denny, J. Prather, B. A. Becker, Z. Albrecht, D. Loksa, and R. Pettit, "A Closer Look at Metacognitive Scaffolding: Solving Test Cases Before Programming," presented at the Proceedings of the 19th Koli Calling International Conference on Computing Education Research, Koli, Finland, 2019. [Online]. Available: <https://doi.org/10.1145/3364510.3366170>.
- [3] D. Janzen and H. Saiedian, "Test-driven learning in early programming courses," presented at the Proceedings of the 39th SIGCSE technical symposium on Computer science education, Portland, OR, USA, 2008. [Online]. Available: <https://doi.org/10.1145/1352135.1352315>.
- [4] R. Garcia, "Evaluating Parsons Problems as a Design-Based Intervention," presented at the 2021 IEEE Frontiers in Education Conference (FIE), 2021, Lincoln, NE, USA, 2021.