

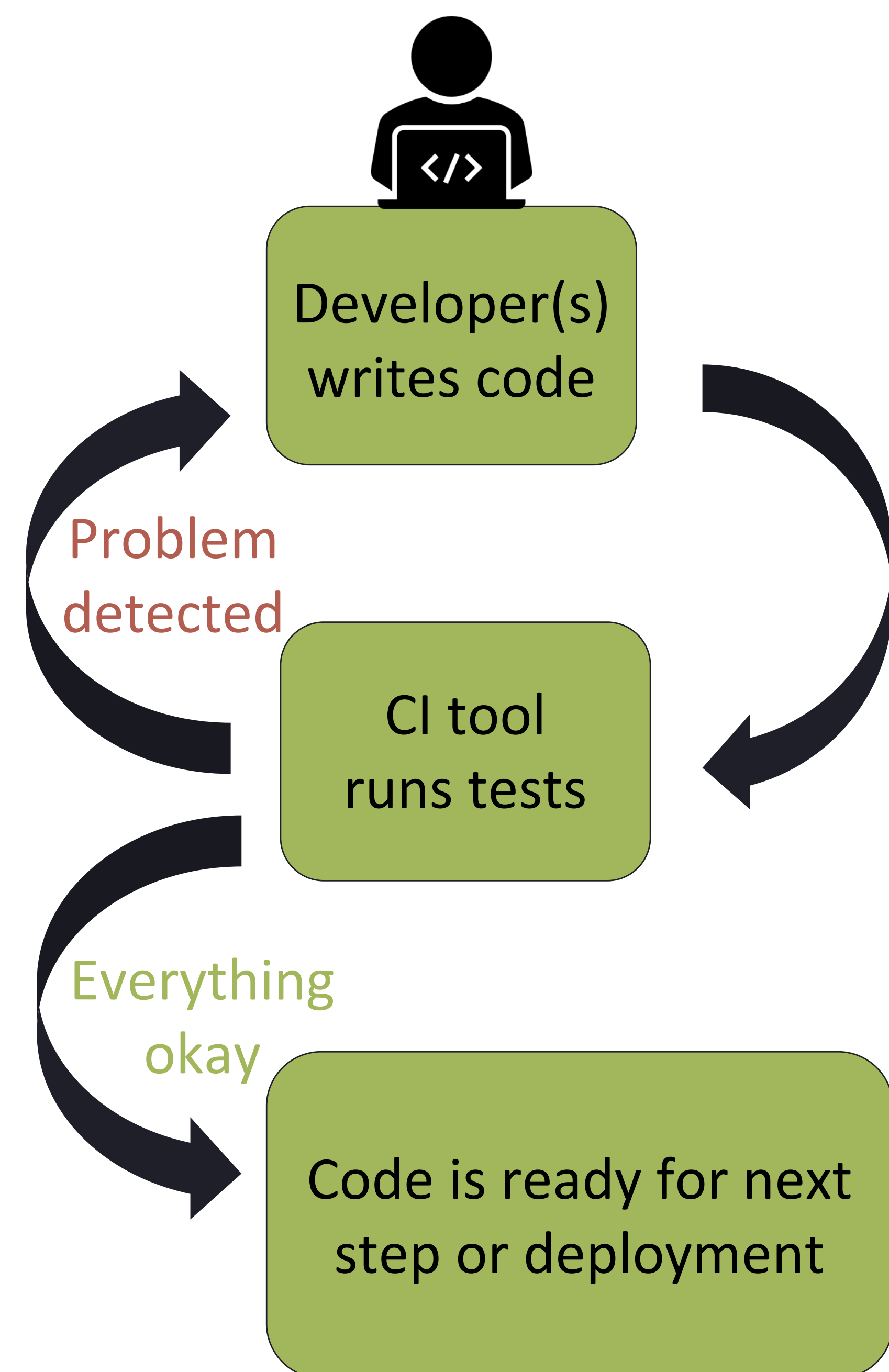
Providing timely feedback through continuous integration in computer science education

Gokce Akcayir¹, Denilson Barbosa², Carrie Demmans Epp¹

¹EdTeKLA Group, Dept. of Computing Science, University of Alberta, ² Dept. of Computing Science, University of Alberta

Continuous integration (CI) in Computer Science (CS) education

Continuous Integration is a professional practice that automatically attempts to build the software product every time new code is added.

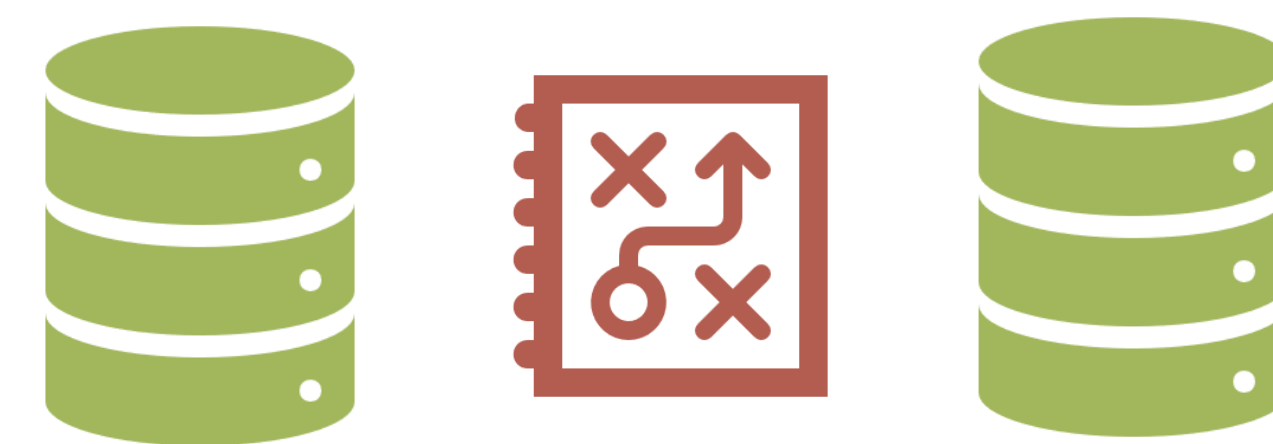


CI in CS education:

- makes student progress visible [5]
- increases engagement [4]
- helps prepare undergraduate students for their career [2]
- supports team collaboration [1]
- provides continuous rapid feedback [3]

CI in the database course

The **Database Management Systems** course covers fundamental algorithms and data structures



Problem

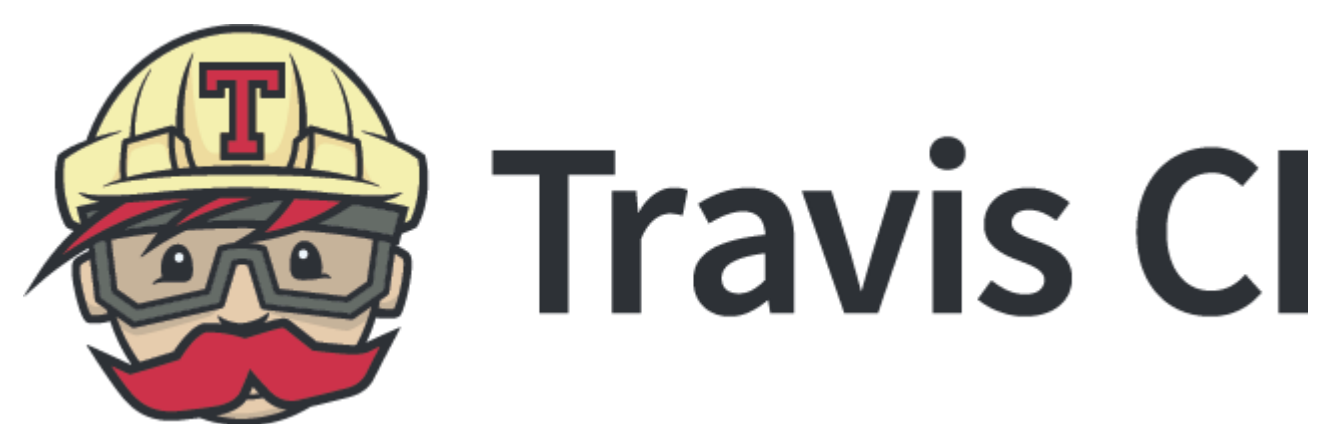
Lack of early feedback

Results in

Students having no idea about the correctness of their work until it has been graded



Possible solution



Research Purpose

To examine if CI use

- is feasible
- helps students
- reduces the amount of TA work



Method

Research design

Action research



Participants

Students and TA's of the course

Procedure

CI Integrated into 3rd year course

- students were taught about CI
- then, they were asked to use it before submitting their three programming assignments

Data collection

Focus groups and one-on-one interviews



Data analysis

Content analysis

Results

Benefits of CI in the course

- Engaging in a professional practice
- Students felt reassured about their assignment submissions
- Provided clear and informative feedback

Issues related to CI use in the course

- Feedback on limited aspects of their code (e.g., validating formatting)
- Student lack of familiarity with CI

References

- [1] Bruegge, B., Reiss, M., & Schiller, J. (2009). Agile Principles in Academic Education: A Case Study. In Proceedings of Sixth International Conference on Information Technology: New Generations, 1684–1686. <https://doi.org/10.1109/ITNG.2009.76>
- [2] Eddy, B. P., Wilde, N., Cooper, N. A., Mishra, B., Gamboa, V. S., Shah, K. M., Deleon, A. M., & Shields, N. A. (2017). A Pilot Study on Introducing Continuous Integration and Delivery into Undergraduate Software Engineering Courses. In Proceedings of 30th Conference on Software Engineering Education and Training (CSEET), 47–56. <https://doi.org/10.1109/CSEET.2017.18>
- [3] Schroeder, A., Klarl, A., Mayer, P., & Kroiß, C. (2012). Teaching agile software development through lab courses. In Proceedings of the 2012 IEEE Global Engineering Education Conference (EDUCON), 1–10. <https://doi.org/10.1109/EDUCON.2012.6201194>
- [4] Sheth, S., Bell, J., & Kaiser, G. (2012). Increasing Student Engagement in Software Engineering with Gamification. Retrieved from <https://academiccommons.columbia.edu/doi/10.7916/D8H99DF0>
- [5] Süß, J. G., & Billingsley, W. (2012). Using continuous integration of code and content to teach software engineering with limited resources. In Proceedings of 34th International Conference on Software Engineering (ICSE), 1175–1184. <https://doi.org/10.1109/ICSE.2012.6227025>