



Delft University of Technology

## Practicing Collaboration with Student Programming Projects

Steenbergen, Merel

**DOI**

[10.1145/3769994.3770048](https://doi.org/10.1145/3769994.3770048)

**Licence**

CC BY

**Publication date**

2025

**Document Version**

Final published version

**Published in**

Proceedings of 25th International Conference on Computing Education Research, Koli Calling 2025

**Citation (APA)**

Steenbergen, M. (2025). Practicing Collaboration with Student Programming Projects. In J. Leinonen, & R. Duran (Eds.), *Proceedings of 25th International Conference on Computing Education Research, Koli Calling 2025* Article 66 ACM. <https://doi.org/10.1145/3769994.3770048>

**Important note**

To cite this publication, please use the final published version (if applicable). Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.



# Practicing Collaboration with Student Programming Projects

Merel Steenbergen  
Delft University of Technology  
Delft, Netherlands  
m.a.steenbergen@tudelft.nl

## Abstract

Collaboration skills are essential for computer science (CS) students, yet many graduates enter the workforce lacking proficiency in collaboration skills. This research aims to identify missing collaboration skills in CS curricula, investigate the impact of prior programming experience on teamwork, and explore the use of software metrics to assess collaboration. Following the Educational Design Research approach, this thesis will lead to actionable recommendations for CS educators and supporting educational materials. The main expected contribution is a practical solution for integrating collaboration skill development into programming project courses which has been designed and prototyped over multiple iterations with testing and evaluation phases.

## CCS Concepts

• **Social and professional topics** → **Computing education; Software engineering education; Applied computing** → **collaboration learning.**

## Keywords

computing education, software engineering education, teamwork, collaboration, group projects

### ACM Reference Format:

Merel Steenbergen. 2025. Practicing Collaboration with Student Programming Projects. In *25th Koli Calling International Conference on Computing Education Research (Koli Calling '25)*, November 11–16, 2025, Koli, Finland. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3769994.3770048>

## 1 Context and Motivation

Collaboration is an integral part of Computer Science (CS) education. Software development in industry often happens in teams and it is expected of graduates of CS majors that they are experienced with the software development process. In practice, there is a gap between the skills, both with tooling and collaboration skills, of recent graduates and expectations of industry [19].

This lack in collaboration skills already becomes apparent during the study period. Students often lack effective approaches to address challenges within teams, and only learn these if they are actively taught and integrated into the course design [6]. Inequalities in domain knowledge between peers can lead to unbalanced teams [12]. While prior programming experience is known to affect

individual academic performance [22], its impact on team dynamics and collaboration remains under-explored.

Software engineering projects provide unique opportunities for measuring performance, as we can extract various metrics from the produced code. Code metrics such as Lines of Code (LOC), can be used to measure developer productivity [9]. However, code contribution is not the only aspect relevant to grading group projects. On the contrary, if collaboration is included in our learning outcomes, this should align with our assessment strategies.

This research will focus on the collaboration skills of CS students. First, we will explore which skills are missing from curricula, to subsequently develop teaching methods which are easily applicable by computer science educators, repeatable by integrating them into already existing group project courses, and measurable to ensure fair grading. Finally, these methods will be tested in practice and improved upon according to the Educational Design Research approach [20].

## 2 Background

The gap between skills of recent CS graduates and the expectations from industry professionals is widely researched. Interviews with managers of recent CS graduates show that the graduates struggle with using software tools, effectively communicating with co-workers and customers, writing unit tests, and other skills [14]. A literature review found similar results, determining that recent graduates lack technical abilities and professional qualities, but also collaboration skills such as oral communication, teamwork, and leadership [13]. Recent graduates also identify these problems and indicate they are struggling to meet expectations [3].

Software engineering projects provide unique opportunities for measuring performance, as we can extract various metrics from the produced code. These can be used as guidance for monitoring and grading students in group projects [15]. In coursework, they are used to determine if the task distribution is equal [10], or to identify different teamwork styles [8]. However, the more important a metric is in decision-making, the more likely it is to be manipulated [2]. Therefore, metrics should not be used as a stand-alone measure to determine the grade [1].

There has been research towards using software metrics for measuring developer behaviour, but this research does not link their results to observed behaviour, on which the grades are based [9]. This context that human assessors have is essential for grading fairly [1]. Despite the fact that observing behaviour is time-intensive, it adds valuable context. This makes it a focus of this research.

Other methods of measuring collaboration in education include questionnaires, diary studies, social network analysis, and lag sequential analysis [16]. However, in these research papers, collaboration can mean different things. A common task distribution strategy is to allow each student to do the task they do best and merge all



This work is licensed under a Creative Commons Attribution 4.0 International License. *Koli Calling '25, Koli, Finland*

© 2025 Copyright held by the owner/author(s).  
ACM ISBN 979-8-4007-1599-0/25/11  
<https://doi.org/10.1145/3769994.3770048>

the work after [21]. This requires little actual collaboration, but does lead to a shared product. In this research, the focus will be on group projects which aim to teach students collaboration, not just shared creation. This might indicate a teamwork-focused grading approach, rather than focusing on the final product. To find well-aligned strategies of grading teamwork skills, several models of soft skills exist [5, 7, 11, 16, 18]. Some of these are targeted to measuring soft skills in higher education, making them well-suited for our goal. Still it remains unclear which of these skills are applicable to computer science careers.

### 3 Research Goals

Simply assigning group programming projects does not guarantee that students develop collaboration skills. Building these skills requires time and effort from both students and educators. Soft skills are wide-ranging, and identifying relevant strategies for dealing with problems in teams is challenging. Identifying which teams are unbalanced is not trivial either, even with data from code repositories. Programming projects often aim to teach the full software development cycle, including the use of version control systems, and client interactions. Teaching collaboration on top of that is a challenge, which is what this thesis aims to support educators with.

The overall goal of this PhD research is to help CS teachers incorporate teaching collaboration skills into project courses. Teaching these skills should be accessible to any teacher without excessive effort. To frame the research into collaboration skills, some context on group projects in CS is required. This context will be provided by research into prior experience and software metrics. The following research questions guide this work:

- RQ1** Which collaboration skills are CS students missing?
- RQ2** How do we teach these collaboration skills to CS students?
- RQ3** What is the effect of a framework designed specifically to teach collaboration to CS students?

### 4 Methodology

The main framework for this thesis is the Educational Design Research approach [20]. This includes an exploratory problem-definition phase, which this project is currently in. Then, there are multiple phases of designing, prototyping, and testing, which are planned for the next few years of this PhD project. The exploratory phase consists of two research projects. First, for RQ1, interviews with early-career computer scientists have been conducted to find the teamwork skills they struggled with early in their career, which would have been beneficial to learn during their education. Then, curriculum research will be conducted to investigate how learning goals pertaining to teamwork are achieved, and which educational design choices are made.

After this stage, lessons learned will be taken into account to design and prototype a teaching module which can be incorporated into software project courses. This module will aim to teach teamwork in a computer science context. The first prototype will be based on the curriculum research and literature regarding teaching methods for teamwork in other disciplines. Possible interventions could be multidisciplinary participation, role-playing assignments, self-reflections, or peer tutoring [4, 17]. The testing phases will

lead to new insights and improved prototypes. Finally, a final evaluation will lead to the end result which should be applicable and repeatable.

### 5 Preliminary results

We conducted over 15 semi-structured interviews with early-career computer scientists focusing on specific teamwork skills and their application in the workplace and education. The results hint towards conflict resolution, dealing with uncertainty, and social skills as valuable skills to be taught during computer science degrees. Furthermore, the participants indicate this might be complicated due to lacking motivation of some team members. The stakes are higher in professional projects, while in education, students know they can get away with doing the bare minimum to pass a course.

These preliminary results suggest that professional social skills might be key to solving these problems, since that involves getting to know your team members and their motivation, possibly making social loafing and conflicts less prevalent and easier to solve. This skill is especially important in a computer science context, where neurodivergence and multidisciplinary were mentioned a few times as factors complicating teamwork.

### 6 Future Work

Professional social skills can be taught and assessed in a few different ways, but it is important that the learning goals of the course align with the design of the teaching methods. The next steps involve curriculum research to see how this alignment is currently achieved. The results from that research will be incorporated into the first prototype. Then, a few testing and evaluation iterations begin, after which the final result will emerge. The final evaluation round will be more extensive to conclude whether the solution meets the pre-determined specifications.

### References

- [1] Efthimia Aivaloglou and Anna van der Meulen. 2021. An Empirical Study of Students' Perceptions on the Setup and Grading of Group Programming Assignments. *ACM Trans. Comput. Educ.* 21, 3, Article 17 (March 2021), 22 pages. doi:10.1145/3440994
- [2] D. Campbell. 1976. Assessing the Impact of Planned Social Change.
- [3] Michelle Craig, Phill Conrad, Dylan Lynch, Natasha Lee, and Laura Anthony. 2018. Listening to early career software developers. *J. Comput. Sci. Coll.* 33, 4 (April 2018), 138–149.
- [4] Joni L. Dirks. 2019. Effective Strategies for Teaching Teamwork. *Critical Care Nurse* 39, 4 (08 2019), 40–47. doi:10.4037/ccn2019704 arXiv:https://aacnjournals.org/ccnonline/article-pdf/39/4/40/118950/40.pdf
- [5] E. B. Durán, , and A. Amandi. 2011. Personalised collaborative skills for student models. *Interactive Learning Environments* 19, 2 (March 2011), 143–162. doi:10.1080/10494820802602667
- [6] Ilenia Fronza and Xiaofeng Wang. 2017. Towards an approach to prevent social loafing in software development teams. In *Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (Markham, Canada) (ESEM '17)*. IEEE Press, 241–246. doi:10.1109/ESEM.2017.37
- [7] Heather Getha-Taylor. 2008. Identifying Collaborative Competencies. *Review of Public Personnel Administration* 28, 2 (06 2008), 103–109. doi:10.1177/0734371X08315434
- [8] Niki Gitinabard, Ruth Okoilu, Yiqao Xu, Sarah Heckman, Tiffany Barnes, and Collin Lynch. 2020. Student Teamwork on Programming Projects: What can GitHub logs show us?. In *Proceedings of the 13th International Conference on Educational Data Mining (EDM)*. 409–416.
- [9] Georgios Gousios, Eirini Kalliamvakou, and Diomidis Spinellis. 2008. Measuring developer contribution from software repository data. In *Proceedings of the 2008 International Working Conference on Mining Software Repositories (Leipzig, Germany) (MSR '08)*. Association for Computing Machinery, New York, NY, USA, 129–132. doi:10.1145/1370750.1370781

- [10] Michael Guttman, Aleksandar Karaka, and Denis Helic. 2024. Attribution of Work in Programming Teams with Git Reporter. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education* (Portland, OR, USA) (SIGCSE 2024, Vol. 1). Association for Computing Machinery, New York, NY, USA, 436–442. doi:10.1145/3626252.3630785
- [11] Ha Le, Janssen, Jeroen, , and Theo Wubbels. 2018. Collaborative learning practices: teacher and student perceived obstacles to effective student collaboration. *Cambridge Journal of Education* 48, 1 (Jan. 2018), 103–122. doi:10.1080/0305764X.2016.1259389
- [12] Rosalie J. Ocker. 2007. A Balancing Act: The Interplay of Status Effects on Dominance in Virtual Teams. *IEEE Transactions on Professional Communication* 50, 3 (2007), 204–218. doi:10.1109/TPC.2007.902656
- [13] Alex Radermacher and Gursimran Walia. 2013. Gaps between industry expectations and the abilities of graduates. In *Proceeding of the 44th ACM technical symposium on Computer science education (SIGCSE '13)*. Association for Computing Machinery, New York, NY, USA, 525–530. doi:10.1145/2445196.2445351
- [14] Alex Radermacher, Gursimran Walia, and Dean Knudson. 2014. Investigating the skill gap between graduating students and industry expectations. In *Companion Proceedings of the 36th International Conference on Software Engineering (ICSE Companion 2014)*. Association for Computing Machinery, New York, NY, USA, 291–300. doi:10.1145/2591062.2591159
- [15] Jan Jaap Sandee and Efthimia Aivaloglou. 2020. GitCanary: A Tool for Analyzing Student Contributions in Group Programming Assignments. In *Proceedings of the 20th Koli Calling International Conference on Computing Education Research* (Koli, Finland) (Koli Calling '20). Association for Computing Machinery, New York, NY, USA, Article 38, 2 pages. doi:10.1145/3428029.3428563
- [16] Verena Schürmann, Nicki Marquardt, and Daniel Bodemer. 2024. Conceptualization and Measurement of Peer Collaboration in Higher Education: A Systematic Review. *Small Group Research* 55, 1 (Feb. 2024), 89–138. doi:10.1177/10464964231200191 Publisher: SAGE Publications Inc.
- [17] Eun Hee Seo and Min Ji Kim. 2019. The Effect of Peer Tutoring for College Students: Who Benefits More from Peer Tutoring, Tutors or Tutees? *The New Educational Review* 58 (2019), 97–106. doi:10.15804/tner.2019.58.4.07
- [18] Chiara Succi, , and Magali Canovi. 2020. Soft skills to enhance graduate employability: comparing students and employers' perceptions. *Studies in Higher Education* 45, 9 (Sept. 2020), 1834–1847. doi:10.1080/03075079.2019.1585420
- [19] Sander Valstar, Caroline Sih, Sophia Krause-Levy, Leo Porter, and William G. Griswold. 2020. A Quantitative Study of Faculty Views on the Goals of an Undergraduate CS Program and Preparing Students for Industry. In *Proceedings of the 2020 ACM Conference on International Computing Education Research (ICER '20)*. Association for Computing Machinery, New York, NY, USA, 113–123. doi:10.1145/3372782.3406277
- [20] Jan van den Akker, Brenda Bannan, Anthony E. Kelly, Nienke Nieveen, and Tjeerd Plomp. 2013. *Educational Design Research*. SLO.
- [21] Anna van der Meulen and Efthimia Aivaloglou. 2021. Who Does What? Work Division and Allocation Strategies of Computer Science Student Teams. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. 273–282. doi:10.1109/ICSE-SEET52601.2021.00037
- [22] Chris Wilcox and Albert Lionelle. 2018. Quantifying the Benefits of Prior Programming Experience in an Introductory Computer Science Course. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (Baltimore, Maryland, USA) (SIGCSE '18). Association for Computing Machinery, New York, NY, USA, 80–85. doi:10.1145/3159450.3159480