Integrating Soft Skills Training into your Course through a Collaborative Activity

Géraldine Brieven
University of Liege
Liege, Belgium
gbrieven@uliege.be

Marcia Moraes
Colorado State University
Fort Collins, USA
Marcia.Moraes@colostate.edu

Dieter Pawelczak
Universität der Bundeswehr München
Munich, Germany
dieter.pawelczak@unibw.de

Simona Vasilache
University of Tsukuba
Tsukuba, Japan
simona@cs.tsukuba.ac.jp

Benoit Donnet
University of Liege
Liege, Belgium
benoit.donnet@uliege.be

Abstract

Nowadays, employers highly value soft skills, yet many students lack these fundamental abilities. Teaching soft skills involves fostering active student participation and facilitating communication of technical knowledge among peers. This approach presents challenges: (*i*) creating an engaging learning environment; (*ii*) ensuring students get timely feedback; (*iii*) finding an approach that is not too time-consuming for instructors to prepare.

The ***C**ollaborative **D**esign & **B**uild* (CDB) activity, described in this paper, was designed to respond to these challenges. It simulates a real-life scenario, triggering students' interest. The success of this collaborative activity hinges on students working together in a structured chain, where each team builds upon and contributes to the success of the others. This fosters student engagement and accountability as they realize the impact of their actions on the entire chain. This pedagogical approach has already been adopted by four universities abroad. This paper shows how it can be deployed in different courses.

Finally, it also discusses how students perceived the activity through four soft skills: collaboration, communication, problem solving and critical thinking. These skills were selected based on their relevance, both in the context of the collaborative activity and in the job market. They are also aligned with the "4C's of 21st Century skills". Results show that while students initially struggled with soft skills, consistent practice throughout the semester boosted their confidence, especially in communication. This makes the activity particularly relevant in the classroom, as communication is considered as the most important soft skill for the future.

CCS Concepts

• **Applied computing** → **Collaborative learning**; • **Software and its engineering** → *Programming teams.*

SIGCSE TS 2025, February 26 – March 1, 2025, Pittsburgh, PA, USA.

ACM ISBN 979-8-4007-0531-1/25/2
https://doi.org/10.1145/3641554.3701877

Keywords

Collaborative learning; Soft skills; Assembly-line; Peer-feedback; Problem-based learning; CS course

ACM Reference Format:
Géraldine Brieven, Marcia Moraes, Dieter Pawelczak, Simona Vasilache, and Benoit Donnet. 2025. Integrating Soft Skills Training into your Course through a Collaborative Activity. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE TS 2025), February 26-March 1, 2025, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3641554.3701877

1 Introduction

Soft skills are crucial for computer scientists [16, 21] due to the high demand from employers [8]. The main requested soft skills [16, 32] are, among others, *collaboration* (*soft skill 1*), *communication* (*soft skill 2*), *problem solving* (*soft skill 3*), and *critical thinking* (*soft skill 4*).

However, teaching them is challenging [13]. First (*Challenge 1*), to effectively acquire soft skills, students must train them, in combination with their hard skills, in an engaging learning environment [18, 27]. For instance, peer-based collaborative environments help engender enthusiasm for programming [6]. Second (*Challenge 2*), students need to receive feedback on what they are doing. In particular, that feedback should be based on specific assessment criteria [1], in order to reinforce students' learning towards soft skills acquisition. Finally (*Challenge 3*), because educational teams are usually overloaded [28, 29], a new pedagogical approach must not be too time-consuming to implement.

This paper deals with these challenges by proposing an activity framework that can be instantiated in various courses and topics across different sessions. The activity framework is referred to as "the ***C**ollaborative **D**esign & **B**uild* (CDB) activity" [3]. From the educational team perspective, the CDB activity is scalable. Whatever the number of participants, it takes about two days to prepare. During a CDB session, the instructors act as facilitators, making students the owners of their learning. This activity aims at simulating in the classroom how actual projects are carried out in the industry. This connection to their future careers reinforces students' motivation [1]. Similar to the industry, the CDB activity requires students to use both their hard and soft skills to solve problems in teams, with each team responsible for a specific task, such as analysis or testing. In practice, the CDB activity employs an assembly-line process [23], where problem solving is conducted through two distinct phases: the *design* phase followed by the *building* phase. Each phase may include one or more steps. Student teams are sequentially responsible for solving a problem at a given step within a specific timeframe. The current solution is then passed on to the next team for review based on specific assessment criteria [1]. After validation, the solution is taken over to the next step.

The activity was deployed in four different universities (four countries over three continents) during academic year 2023–2024. A total of 144 students were exposed to it through 8 sessions. This paper shows, through the analysis of surveys responses, how these students perceived it through the lens of the four soft skills of interest. In particular, we demonstrate that some students were initially uncomfortable with practicing soft skills. However, the organization of multiple sessions throughout the semester was shown to increase their confidence in using them, especially communication.

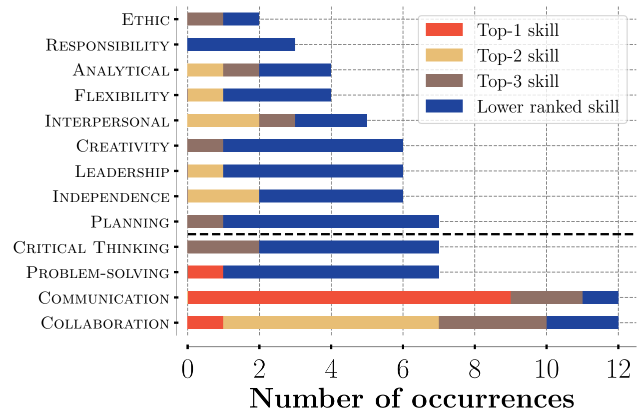


Figure 1: Most important soft skills in literature.

2 Related Work

In collaborative learning, students participate in a shared dialogue where they have opportunities to explain, refine, and expand upon their ideas [31]. Collaborative learning can be implemented in different ways, such as Pair Programming [24] or Peer Instruction [17]. Pair Programming involves students working on coding tasks, with one student writing the code while the other reviews and suggests improvements. In contrast, Peer Instruction fosters communication between students, encouraging them to work out a solution together. Another collaborative activity example is the Process Oriented Guided Inquiry Learning (POGIL) [31] which aims to develop students' mastery of discipline-specific concepts as well as "process skills" like communication, teamwork, and critical thinking. The CDB activity is a combination of these three previous active learning activities and it focuses on the following process skills: oral and written communication, teamwork, problem solving, critical thinking, information processing, and assessment. In the CDB activity, assessment is promoted through peer-feedback.

The CDB activity is also an application of Problem-Based Learning (PBL). The PBL definition [27] aligns with the CDB activity by being an active learning approach that aims to solve problems and enables students to learn new knowledge skills by working collaboratively within a small group.

Previous studies have shown that PBL effectively promotes the 4C's of 21st-century skills (Creativity, Communication, Collaboration and Critical thinking skills) [30]. That led us to focus on these skills, except that rather than considering creativity as such, we expand it to problem solving [14] which aligns better with the skills required in the CDB activity.

To validate these skills, we examined their importance as soft skills. Soft skills, defined as "personal attributes enabling someone to interact effectively and harmoniously with other people" [20], are broadly motivated in the literature. They also include a large set of skills that may vary slightly between studies. Much previous work ranks them, based on what is taught at the university and the needs from industries. Fig. 1 presents the top-13 soft skills based on 12 papers. These papers were selected based on their publication date (after 2016) and their special interest in Information Technology (IT) and software engineering professions [8, 9, 11–13, 16, 19, 21, 22, 25, 26, 32]. Fig. 1 shows the number of occurrences of each soft skill across the 12 papers. Furthermore, it specifies the number of papers that were ranking them in the first, second, third, or lower position.

The main observation is that communication is often ranked as the top-1 soft skill. Collaboration also appears as crucial for career success. The next most frequent skills are problem solving, critical thinking and planning. It is worth noting that the importance of critical thinking may have been underestimated here, as ChatGPT was emerging only in the last two years [10]. This paper does not discuss planning because instructors are the ones dividing the problem-solving process into steps for students to solve and setting time limits for each task. All in all, Fig. 1 confirms the relevance of the four 21st-century skills (soft skills) studied in this paper.

3 The CDB activity

The CDB activity is made up of two phases: the design and the building phases. The goal is to encourage students to reflect on what to do and how to do it before starting the development process. This teaches them to work methodically and avoid a trial-and-error approach where they spend most of their time on testing and patching their solution.

Fig. 2 illustrates how the activity is set up. The right side of Fig. 2 ("Classroom Configuration") shows that G groups of students are formed. Each group is divided into T teams and each team comprises S students. The goal of each group is to collaborate (*soft skill 1*) to solve T problems (*soft skill 3*) in a limited amount of time. The timing is given by the instructors, meaning that students do not have to plan their tasks. The left part of Fig. 2 shows how the T problems are getting progressively solved, in parallel, over time, following the T steps required to frame the problem-solving process. Typical steps in the *Design* phase are **analysis** (where students specify modules) and/or **modeling** (where students make diagrams, showing how the different modules will work). The *Building* phase includes at least one step, consisting of **implementing** the final solution. This conception and its steps are inspired by real professional life, where large-scale projects can be successfully developed only if teams collaborate and communicate efficiently with each other. Early

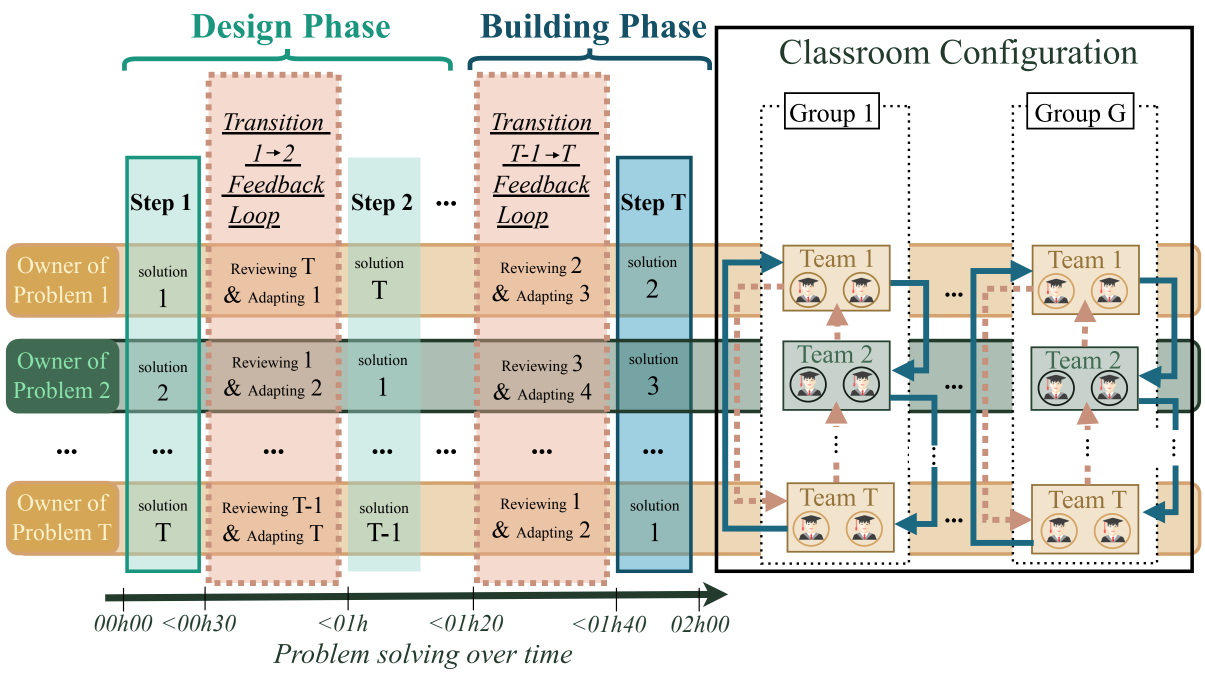


Figure 2: The Cdb activity framework with two students per team (S=2).

Criteria supporting step 1	OK ?	Comments (referred by tags if needed)
Input(s)	☐	
Output	☐	
Objects that are used :		
- have a relevant name	☐	
- have a type that exists in C	☐	
- have a purpose that is clearly expressed	☐	
Sp ...... :		
- is generic (i.e., doesn't restrict to an example)	☐	
- has a clear description	☐	
- has a functional description	☐	

Figure 3: An example of checklist to validate the consistency of problem definition and decomposition.

exposure to this way of working seems to be both important and motivating for students (*Challenge 1*).

Coming back to Fig. 2, the Cdb activity works as follows. At the beginning, each team is responsible for understanding and defining one problem (corresponding to step 1). For a given problem, the T steps are addressed sequentially, with each team writing a solution, turn-by-turn, for a specific problem. At the end of each step, the product resulting from each team's work moves to the next team, clockwise, similarly in each group, as shown by the plain arrows (Fig. 2, right part). It is crucial for each team to be clear enough in their work to enable the next team(s) to progress the solution (*soft skill 2*). Depending on the step, their work can be specifications of function, diagrams, or pieces of code, etc.

To articulate these steps, some transition periods allow each team (*reviewers*) to report a feedback on the work completed by the previous team (*submitters*) (*soft skill 4* and *Challenge 2*). During these periods, each team holds the two roles in parallel [4]. For instance, team 2 acts as the reviewers of team 1 (as indicated by the dashed arrow – Fig. 2, right part) while simultaneously being the submitters for team 3. The goal is to limit the impact of a "poor quality work" on subsequent step productions that rely on the previous ones.

The feedback is based on a checklist [1, 2] (see Fig. 3). The first column lists the criteria a step output should meet. They are selected from the checklist supporting the course evaluations. In the Cdb activity, they are kept quite general, as the purpose is making students responsible for putting forward a solution. We should not gradually disclose it through the criteria. The next two columns allow each team to specify whether a criterion is checked or not and write comments in them. Once filled, the checklist is returned to the submitters who should adapt their work based on the feedback received.

For the instructors, such an activity only requires the following preparation (*Challenge 3*): (*i*) select a profile of problems (e.g., array manipulation); (*ii*) define different steps this kind of problems should be solved through; (*iii*) define a checklist for each problem-solving step in order to support the reviewing phase; (*iv*) choose the size of each team and the time to allocate to each step and (*v*) write problem statements.

It is worth noting that the Cdb activity may be applied to other STEM disciplines than CS, as long as the selected problems could be solved following a sequence of steps.

Table 1: Setting up the CDB activity parameters for the four countries (with $N = 144$ participants across them).

Parameters	Belgium				Germany		USA	Japan
(Relation between them: $N = G \times T \times S$)	Session 1	Session 2	Session 3	Session 4	Session 1	Session 2	Session 1	Session 1
N (Total number of participants in the session)	50	43	35	41	24	12	35	26
G (Total number of Groups of students)	8	6	5	5	4	2	8	3
T (Number of Teams within a Group (≥ 2))	3	3	3	3	3	3	2	3
S (Number of Students per Team (≥ 2))	2 or 3	2 or 3	2 or 3	2 or 3	2	2	2 or 3	2 or 3

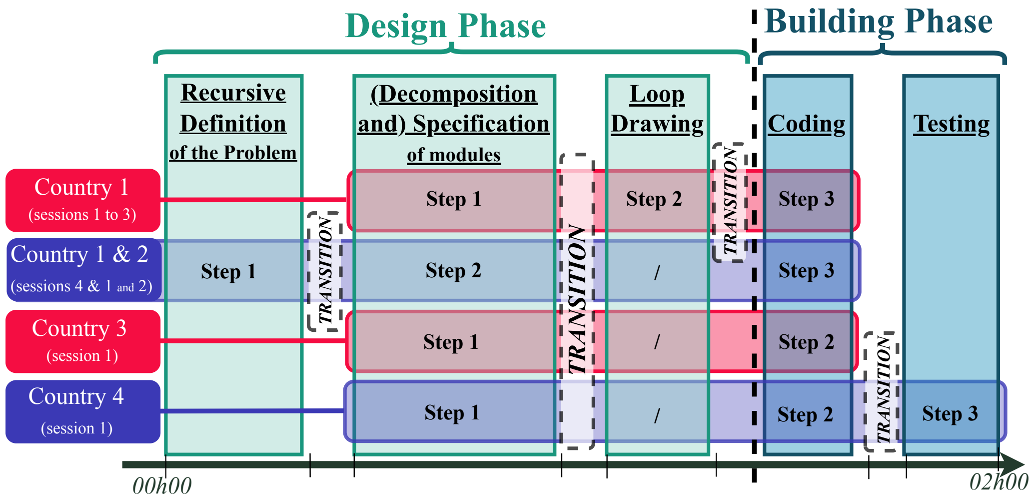


Figure 4: The different steps in the Design and Building phases, as applied in each session in each country.

4 Deployment and Analysis

In this section, Sec. 4.1 discusses the deployment of the CDB activity in four countries. Sec. 4.2 details how survey data was collected and Sec. 4.3 presents the results.

4.1 Context and Participants

During academic year 2023–2024, the CDB activity has been deployed in four different countries over three continents: Belgium (four sessions of the activity – Europe), Germany (two sessions – Europe), USA (one session – North America), and Japan (one session – Asia). In total, 144 students took part in the activity among these countries. Table 1 details the activity parameters for each of them.

In Belgium, the CDB activity targeted undergraduate students during their first year at the University. The first three sessions were organized in an introduction to programming course (CS1), while the last session was held in the next course (CS2) [15], during the second semester. The students in Germany were enrolled in a bachelor degree program for electrical engineering and IT. They were attending an advanced programming course on machine-oriented programming in the third semester with two preceding computer science courses. In USA, the activity targeted first year undergraduate STEM students that were registered in an introducing to programming course (CS1). In Japan, the students were undergraduates (first, second, even third year students), and a few exchange students. They were majoring in social sciences or life and environmental sciences. They were absolute beginners and this was their first programming course. In Belgium and Germany, students were learning C programming language while in USA and Japan, they were using Python.

Fig. 4 illustrates how the steps of the CDB activity were instantiated in each course. In the four countries, the first or the second step focused on specifying the module(s) to implement. For all the countries (except Japan), the last one consisted of coding the final solution. In Japan, coding came as second step, and the third one consisted of testing the code. In Belgium and Germany, a pre or intermediate step was added in the Design phase: modeling the problem or the solution, either formally (through a mathematical recursive definition of the problem), or informally (through a drawing representing the loop behavior [5]).

Typical problems students had to solve are "*Write a program which checks whether a given string contains two consecutive identical characters*" (Japan– session 1) or "*For a given interval [a,b[, display in the terminal all the numbers (from this interval) whose sum of digits equals 10.*" (Belgium– session 2).

Finally, it should be noted that the CDB sessions were not graded in order to limit any stress. Moreover, the students were allowed to choose their group and teammates to optimize group cohesiveness and enthusiasm, as reported by Ciani et al. [7].

4.2 Data Collection

At the end of each CDB session, students took an anonymous survey. All students gave their consent for using their answers in this study. The survey counted 24 questions examining the four

Table 2: Survey questions, listed by soft skill.

ID	Soft skill	Question
1	Collaboration	Working in teams, with each team responsible for a specific step in the solution, contributed to creating a more comprehensive outcome.
2	Communication	Written communication with the 2 other teams went well.
3	Problem solving	I could easily find a solution to step 2 (implementation).
4	Critical Thinking	I was comfortable analyzing someone else's production.

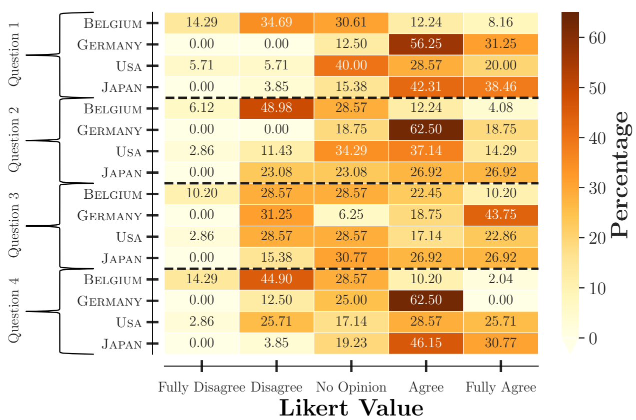


Figure 5: Survey results for session 1 ($N_{session1} = 135$). Questions are numbered and listed in Table 2.

soft skills of interest: (*i*) collaboration (four questions), (*ii*) communication (four questions), (*iii*) problem solving (seven questions), and, (*iv*) critical thinking (nine questions). The students' answers are on a 5-point Likert scale. The survey reliability was assessed using Cronbach's alpha ($\alpha = 0.89$), suggesting good internal consistency. Investigating all the answers is out the scope of this paper. Rather, we focus on the most relevant question for each soft skill (see Table 2).

4.3 Results

Fig. 5 shows students' perceptions of soft skills after their first session of the Cdb activity. With regard to the collaboration skill, the survey shows that advanced students recognize more advantages in teamwork than first-year students in their first semester. In Germany, Usa, and Japan, Fig. 5 shows that only a small proportion (from 0 to 11.4%) of students disagreed with the benefits of teamwork, while in Belgium (first semester), about 49% of the students did not see the point of working in teams. The students reflection on their communication skills reveals a similar trend. It is necessary that students learn the technical vocabulary before they feel comfortable in communicating technical details in writing. This may explain why most students in Belgium felt written communication with other teams was poor. In Usa and Japan, responses were mixed, with just over half (fully) agreeing communication went smoothly. In Germany, 81.25% reported good communication.

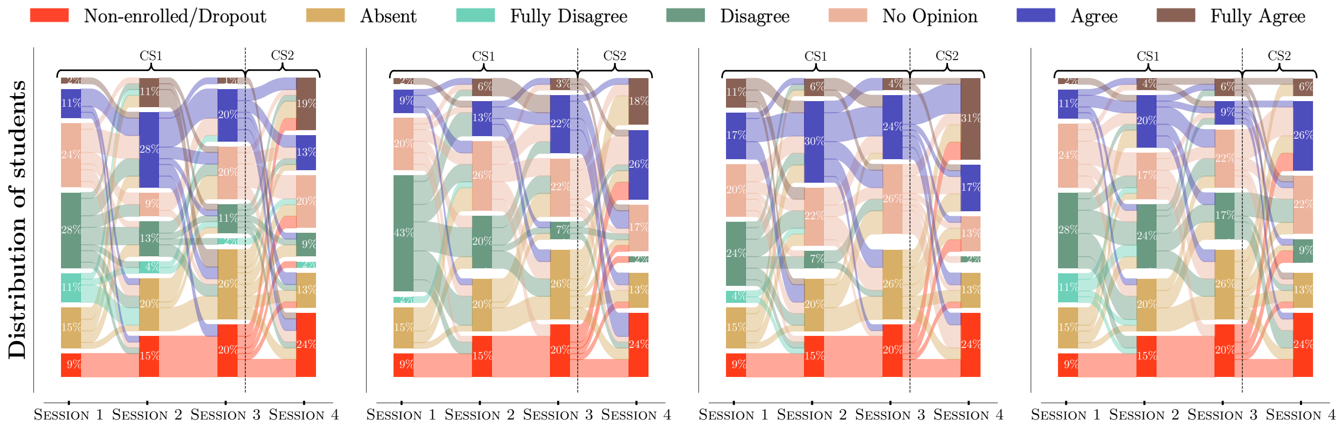
Results regarding problem solving differ the most from the three other soft skills. While more students in Germany and Usa were rating it more negatively, that is where students from Belgium felt the most comfortable. This may be due to the higher complexity of the problems given in Germany and Usa. That is also where the highest variance is found, likely because of the dependence on the solution design provided by the previous team(s). In Belgium, Germany and Usa, more than 30% of students found it difficult to construct the solution while in Japan, 54% found it easy and 30% had no opinion.

Finally, the results to the last question indicate that advanced students feel more comfortable analyzing and critiquing others' solutions. In Belgium, only 12.2% of students felt comfortable with critical thinking, while in Japan, over 75% reported feeling (strongly) confident in this skill. This could be due to the lower complexity of the problems in Japan (as suggested by students' answers to Question 3), in addition to their greater experience with social behavior, as some students study social science. These answers, along with those to Question 2, support the claim that if teams struggle with the written communication, the other teams will have more difficulty analyzing their work.

After this initial assessment of soft skills, we examine how the Cdb activity impacts these skills by tracking changes over multiple sessions. For this purpose, we only consider students attending at least two sessions and we focus on Belgium, where the largest number of sessions were organized and where the Cdb activity was the only teamwork activity across all the courses students were registered to. Fig. 6 presents four Alluvial diagrams[1] based on seven clusters. The "Non-enrolled/Dropout" cluster includes students who were not registered to the course holding the current session and those who dropped out[2]. Regarding the results, Fig. 6 illustrates an increase in absences over the sessions in the CS1 course. Fig. 6a, 6b and 6d show that most of the students who stopped participating in the Cdb activity had a poor or mixed opinion about their soft skills in the first session. This suggests that students need to feel a minimum of comfort with soft skills to feel motivated to work with others. That drop of unmotivated students resulted in increasing students' agreement in the second session, especially regarding collaboration, likely because more students were involved. However, except for communication, this trend did not continue into session 3. For collaboration, almost half of the students who rated it positively in session 2 did not attend session 3 (and among these, half of them were dropping

[1] An Alluvial diagram is a type of flow chart that represents changes in a network structure over time. In that sense, it helps identify patterns and trends (see https://en.wikipedia.org/wiki/Alluvial_diagram).

[2] In Belgium, it is common for first-year students to drop out because they feel their chosen curriculum does not match their real expectations or prerequisites. These students can be identified as they stop taking assignments, whose completion is necessary to pass the course. Some of them also notify the instructors they change their direction.



(a) Collaboration skill. (b) Communication skill. (c) Problem-solving skill. (d) Critical thinking skill.

Figure 6: Evolution of soft skills throughout the sessions in Belgium (66 participants across the four sessions).

out from the course). This caused a decrease of agreements that was not fully offset by the growing opinion of the participants in sessions 2 and 3. The problems in session 3 may have been also too difficult for their current level of hard skills, as problem complexity increased throughout the semester. This is supported by the decrease in positive opinions on problem solving and critical thinking, which are more closely linked to the complexity of the tasks. The more competent students, the better equipped to solve problems and analyze others' work. In session 4, perceptions of all soft skills improved, possibly due to students having gained experience from their CS1 course, as well as the problems better aligned to students' hard skills. Fig. 6c, shows that 74.6% of the participants found easy to implement the solution.

Overall, students generally moved to higher agreement clusters ("Agree" and "Fully agree") over the sessions, indicating more confidence in soft skills. Among the four soft skills, communication saw the most significant improvement, with participants' agreements rising from 13.9% in session 1 to 67.8% in session 4. This makes the CDB activity particularly relevant in the classroom, as communication is viewed as the most important soft skill for the future (see Fig. 1).

5 Lessons Learned and Conclusion

From deploying the CDB activity in four universities and studying students' perceptions on soft skills, we learned five lessons to increase the impact of this activity on soft skills.

- To limit absences, we may introduce the CDB activity as a class assignment. Only the output of the first step should be graded, as it is not dependent on the other students' work. Moreover, it will motivate students to lay strong foundations for the next step(s) in the chain.
- Students may become more engaged by realizing the relevance of soft skills. For that purpose, we may show recordings of industry professionals discussing the collaborative nature of the software design process or invite an industry partner to give a brief talk about it.
- Students, especially those who are early in their curriculum, should be better prepared to the CDB activity. To make them feeling more comfortable with practicing soft skills, we suggest a 15-minute seminar providing guidelines for mobilizing them, especially those that students initially rated lower (i.e., critical thinking and communication).
- During the first session of the CDB activity, more supervision is actually needed to keep the chain going. In fact, many students tend to write some code directly, even though they were asked to contribute to the solution design first. They are not familiar with the isolated analysis tasks assigned to them. Instructors should have as part of their teaching pedagogy the habit of making students think about the problem and how to solve it before coding a possible solution.
- Our results suggest that soft skills acquisition is a long-term journey. Therefore, the CDB activity should be conducted more than once over a course, in order to make students better at leveraging them.

In conclusion, we believe that ***Collaborative Design & Build*** (CDB) activity, along with the lessons learned, will inspire new instructors to include it in their courses, as it has already done on three continents. Initially designed for beginners, the CDB activity involves solving small problems in chains, mobilizing so both hard and soft skills. The goal is to familiarize students as early as possible with industry-team based design. In an educational context, this activity is a good preparation for Capstone Projects [32], which are closer to real-life projects, but therefore more complex and not accessible to beginners. Unlike Capstone Projects, the CDB activity involves solving different problems in parallel within a few hours, with solutions passing from one team to another. This beginner-friendly learning method is a pedagogical innovation in CS education.

Acknowledgments

This work is supported by the CyberExcellence project funded by the Walloon Region, under number 2110186.

References

[1] D. Alt, L. Naamati-Schneider, and D. J. N. Weishut. 2023. Competency-Based Learning and Formative Assessment Feedback as Precursors of College Students' Soft Skills Acquisition. *Studies in Higher Education* 48, 12 (May 2023), 1901–1917.

[2] S. Bharuthram and M. Patel. 2017. Co-Constructing a Rubrick Checklist with First Year University Students: A Self-Assessment Tool. *Journal of Applied Language Studies (Apples)* 11, 4 (2017), 35–55.

[3] G. Brieven, L. Leduc, and B. Donnet. 2022. Collaborative Design and Build Activity in a CS1 Course: A Practical Experience Report. In *Proc. 8th International Conference on Higher Education Advances (HEAd)*.

[4] G. Brieven, L. Leduc, and B. Donnet. 2023. How Students Manage Peer Feedback Through a Collaborative Activity in a CS1 Course. In *Proc. 9th International Conference on Higher Education Advances (HEAd)*.

[5] G. Brieven, S. Liénardy, L. Malcev, and B. Donnet. 2023. Graphical Loop Invariant Based Programming. In *Proc. Formal Method Teaching Workshop (FMTea)*.

[6] J. Chase and E. Okie. 2000. Combining Cooperative Learning and Peer Instruction in Introductory Computer Science. *ACM SIGCSE Bulletin* 32 (March 2000), 372–376. Issue 1.

[7] K. Ciani, J. Summers, M. Easter, and K. Sheldon. 2008. Collaborative Learning and Positive Experiences: Does Letting Students Choose Their Own Groups Matter? *Educational Psychology* 28, 6 (October 2008), 627–641.

[8] M. Danao. 2024. *11 Essential Soft Skills In 2024 (With Examples)*. https://www.forbes.com/advisor/business/soft-skills-examples/ [Last Accessed: July 2nd, 2024].

[9] M. Daneva, A. Herrmann, N. Condori-Fernández, and C. Wang. 2019. Understanding the Most In-demand Soft Skills in Requirements Engineering Practice: Insights from Two Focus Groups. In *Proc. International Conference on Evaluation and Assessment in Software Engineering (EASE)*.

[10] F. Emmert-Streib. 2024. Importance of Critical Thinking to Understand ChatGPT. *European Journal of Human Genetics* 32 (April 2024), 377–378.

[11] M. Galster, A. Mitrovic, S. Malinen, and J. Holland. 2022. What Soft Skills Does the Software Industry *Really* Want? An Exploratory Study of Software Positions in New Zealand. In *Proc. ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*.

[12] O. Glazunova, O. Parkhomenko, V. Korolchuk, and T. Voloshyna. 2022. Building the Professional Competence of Future Programmers Using Methods and Tools of Flexible Development of Software Applications. *Information Technologies and Learning Tools* 89 (June 2022), 48–63.

[13] W. Groeneveld, B. A. Becker, and J. Vennekens. 2020. Soft Skills: What do Computing Program Syllabi Reveal About Non-Technical Expectations of Undergraduate Students?. In *Proc. ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*.

[14] P. Ø. Haavold and B. Sriraman. 2022. Creativity in Problem Solving: Integrating Two Different Views of Insight. *ZDM International Journal on Mathematics Education* 54 (April 2022), 83–96.

[15] M. Hertz. 2010. What Do CS1 and CS2 Mean?: Investigating Differences in the Early Courses. In *Proc. ACM Technical Symposium on Computer Science Education (SIGCSE)*.

[16] I. Holik, I. D. Sanda, and G. Molnar. 2023. The Necessity of Developing Soft Skills in STEM Areas in Higher Education, with Special Focus on Engineering Training. *Athens Journal of Technology and Engineering* 10, 4 (December 2023), 199–214.

[17] V. Kannan and S. P. Gouripeddi. 2018. Enhancement in Critical Thinking Skills Using the Peer Instruction Methodology. In *Proc. IEEE International Conference on Advanced Learning Technologies (ICALT)*.

[18] E.-S. A. Lee, L. Danukarjanto, S. Sharmin, S.-Y. Hung, S. Huang, and T. Su. 2024. Exploring Student Motivation in Integration of Soft Skills Training within Three Levels of Computer Science Programs. In *Proc. ACM Technical Symposium on Computer Science Education (SIGCSE)*.

[19] G. Matturro, F. Raschetti, and C. Fontán. 2019. A Systematic Mapping Study on Soft Skills in Software Engineering. *Journal of Universal Computer Science* 25, 1 (January 2019), 16–41.

[20] Oxford University Press. 2024. *Oxford Learner's Dictionaries*. https://www.oxfordlearnersdictionaries.com [Last Accessed: June 9th, 2024].

[21] F. F. Patacsil and C. L. S. Tablatin. 2017. Exploring the Importance of Soft and Hard Skills as Perceived by IT Internship Students and Industry: A Gap Analysis. *Journal of Technology and Science Education (JOTSE)* 7, 3 (August 2017), 347–368.

[22] B. J. Polo, P. A. Silva, and M. E. Crosby. 2018. Applying Studio-Based Learning Methodology in Computer Science Education to Improve 21st Century Skills. In *Proc. International Conference on Learning and Collaboration Technologies (LCT)*.

[23] J. Rosario, D. Lebowitz, A. Webb, L. Ganti, A. Vera, T. Macintosh, A. Walker, and J. Rubero. 2020. Assembly Line Education: A Novel Educational Technique for Today's Learners. *Cureus* 12, 2 (February 2020).

[24] N. Salleh, E. Mendes, and J. Grundy. 2011. Empirical Studies of Pair Programming for CS/SE Teaching in Higher Education: A Systematic Literature Review. *IEEE Transactions on Software Engineering* 37, 4 (July/August 2011), 509–525.

[25] C. Scaffidi. 2018. Employers' Needs for Computer Science, Information Technology and Software Engineering Skills Among New Graduates. *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)* 8, 1 (February 2018), 1–12.

[26] M. Stevens and R. Norman. 2016. Industry Expectations of Soft Skills in IT Graduates: A Regional Survey. In *Proc. Australasian Computer Science Week Multiconference (ACSW)*.

[27] H. Tadjer, Y. Lafifi, H. Seridi-Bouchelaghem, and S. Gülseçen. 2020. Improving Soft Skills Based on Students' Traces in Problem-Based Learning Environments. *Interactive Learning Environments* 30, May (2020), 1879–1896.

[28] P. te Braak, F. Van Droogenbroeck, J. Minnen, T. P. van Tienoven, and I. Glorieux. 2022. Teacher's Working Time for Time-Use Data: Consequences of the Invalidity of Survey Questions for Teachers, Researchers, and Policy. *Teaching and Teacher Education* 109 (January 2022), 103536.

[29] G. Thompson, S. Creagh, M. Stacey, A. Hogan, and N. Mockler. 2024. Researching Teachers' Time Use: Complexity, Challenges, and a Possible Way Forward. *The Australian Educational Researcher* 51 (September 2024), 1647–1670.

[30] T. Towip, I. Widiastuti, and C. Budiyanto. 2022. Students' Perceptions and Experiences of Online Cooperative Problem-Based Learning: Developing 21st Century Skills. *International Journal of Pedagogy and Teacher Education* 6, 1 (April 2022), 37.

[31] A. Yadav, C. Mayfield, S. Moudgalya, C. Kussmaul, and H. Hu. 2021. Collaborative Learning, Self-Efficacy, and Student Performance in CS1 POGIL. In *Proc. ACM Technical Symposium on Computer Science Education (SIGCSE)*.

[32] J. Zheng, Z. Li, and L. Li. 2024. *Go beyond the Technology: Emphasizing Soft Skills in IT/IS Capstone Classes*. Elgaronline, Chapter 3, 44–67.