10th International Conference on Higher Education Advances (HEAd'24)
Universitat Politècnica de València, València, 2024
DOI: https://doi.org/10.4995/HEAd24.2024.17182

Identifying Potential for Improvement in Computer Science Degree Programs from the Student Perspective

Axel Böttcher, Veronika Thurner, Sabine Hammer
HM Munich University of Applied Sciences, Department of Computer Science and Mathematics, Lothstraße 64, D-80335 München, Germany.

How to cite: Böttcher, A.; Thurner, V.; Hammer, S. 2024. Identifying Potential for Improvement in Computer Science Degree Programs – the Student Perspective. In: 10th International Conference on Higher Education Advances (HEAd'24). Valencia, 18-21 June 2024. https://doi.org/10.4995/HEAd24.2024.17182

Abstract

We want to pick up students where they are – but we don't even know where they are since we may not know enough about the status of our students and their needs and wishes.

In this paper we share experiences on a workshop that we performed together with students to let them reflect on their situation and thus gain insights into what students need and want to study successful – and to foster their self-reflection competences accordingly. A student-centered workshop can help other faculties and teachers learn about their students' perspectives on their organizational processes and their teaching. In addition, such a workshop helps students to identify deficits in their metacognitive skills.

Based on the results of the workshop, we were able to identify various starting points for improving teaching as well as administrative and organizational processes.

Keywords: *Student expectations; student perspective; self reflection; heterogeneity.*

1. Introduction and Motivation

As many technical degree programs worldwide, our program on Computer Science suffers from high dropout rates of around 50% (Böttcher et al., 2020). Even so, we are convinced that we could contribute to preventing many students from dropping out, if we were able to timely identify those students who are at risk of dropping out and provide them with the specific support that they individually need at their time of doubt. To achieve this, we need to better understand why our students struggle or even fail. Final exam grades and failure rates in introductory courses are only an observable symptom. However, the *causes* of failure are

manifold and include, e. g., a lack of mathematical and technical knowledge as well as underdeveloped meta-cognitive skills.

A hypothesis about another cause is that several of our students already possess considerable programming skills, while several others do not, and that we do not appropriately meet the different needs of this highly heterogeneous cohort. This hypothesis is supported by Figure 1, illustrating the results of a survey of two subsequent cohorts, performed three to four weeks after the start of the first semester in the course on Software Development.

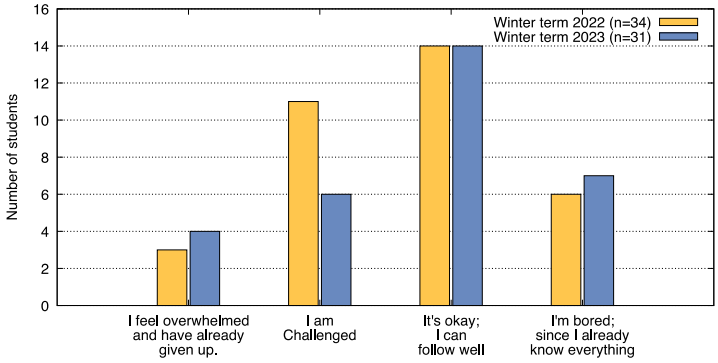


Figure 1. Students' perceived degree of challenge about three to four weeks after start of the semester.

For us teachers, the heterogeneous initial skill set is a severe challenge, since we are in danger of loosing one third of the cohort while still boring a major part. Yet even worse: teachers might feel obliged to increase the pace of the course, because some of the students already have some experience (Petersen, 2016). Experience shows that this non-majority students rather loudly request more input, or even tend to ridicule the perceived "snail's pace" at which concepts are introduced that they already have down their belt, e. g. from previous studies or apprenticeships, while they conveniently block out their memories of how they themselves struggled when they initially learned these skills. As a consequence, they risk to drown out the rather quiet albeit greater group of students that struggles – usually silently, as they are often too ashamed or even scared to explicitly voice their struggles.

Obviously, the goal of our teaching is to increase our students' skill sets, in all parts of the cohort. Therefore, to keep on board the significant share of students that are already challenged or even overwhelmed, we must properly understand the causes of their struggle, and which kind of support they really need. Specifically, if we design interventions to address and hopefully resolve existing difficulties, we must be sure that our students really benefit from these offerings, i. e., we must know and understand what they want – or make them want what they need.

In this paper, we report on our experience with a workshop format that we have created to provide a safe and protected environment for eliciting student wishes and needs. This workshop format requires moderate effort in terms of preparation, implementation, and evaluation. It can help other faculties and teachers learn about their students' perspectives on their organizational processes and their teaching. Furthermore, it helps students to discover deficits in their metacognitive competences.

2. Related Work

Exam pass rates and drop-out rates in STEM and computer science degree programs are often reported as unsatisfactorily low and needing improvement, throughout different countries and institutions. A fundamental study with a broader geographic and institutional context is e. g. (Simon et al., 2019). The authors claim that overall exam pass rates are at around 75%. Corresponding to these observations, dropout rates from our degree program are usually even higher.

On their way to writing and passing exams especially in the entry phase of a degree program, students must master a multitude of cognitive and metacognitive skills. Underdeveloped meta-cognitive skills need to be developed (Falkner, 2014). Investigations on deficits in metacognitive skills are often done with questionnaires and interviews, e. g. (Petersen, 2016).

3. Goals

Our goals are twofold. Firstly, in our roles as teachers, we want to recognize and understand what our students need to effectively learn. In addition, in our roles as faculty staff, we want to identify general potential for structural improvement of our institution and our offerings. Secondly, students need to enter into a process of self reflection that helps them recognize their own development needs, and they should become aware of improvement potential in their self-organization. Gaining insight into our students' needs in all these aspects will help us to develop interventions that effectively support our students. Carnegie et al. (2014) report positive experiences in developing an engineering program by addressing more than just academic issues. We also have such measures in mind, including a kind of "pastoral care".

4. Approach, Design, and Results

To identify the students' perspective, we specifically designed a workshop format which we embedded into our first semester course on software development. We dedicated one afternoon in the week just before Christmas to work with our students on a meta-level, to help students detect their needs, and to simultaneously offer them a chance to recognize how they can better organize their course of study and improve their self-organization. For this workshop, we

designed four stages that addressed "organizational aspects", "personal issues", "self-competences", and "professional topics related to software development", respectively. Each stage was accompanied by a teacher or a coach. In order to prevent any personal biases in the future student-teacher-relationship, it is important to accompany those stages where students reflect on their personal or even private situations by staff who are not (and never will be) directly involved in teaching and grading of the participating students.

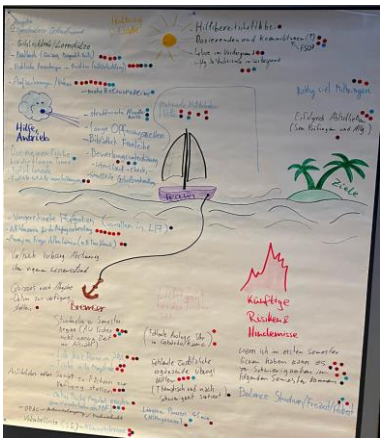
For the workshop, students were divided into groups of four to five persons each, which stayed together throughout the entire workshop, rotating from stage to stage as a group, similar to a circuit training. Stages were distributed around the entrance hall of our building, which was unusually uncrowded due to the approaching holidays, thus offering both space and sufficient privacy. Each stage was equipped with flip charts or pin boards, sticky dots and pens. Thus, students were actively engaged and documented their issues and findings themselves and in their own words. Every 30 minutes, each student group moved on to their next stage.

We started the workshop with a short plenary session where we elicited our intentions and goals and gave an overview of the general structure. At the end of the workshop, we gathered all groups back into a plenary, to do a final reflection on lessons learned and open issues. After that, we invited all students to a Christmas snack, to say thank your for their time and effort. Overall, 14 students participated, corresponding to hardly half of the active cohort, which was notably lower than expected. Nevertheless, all participating students were highly engaged throughout the entire workshop, so that we still were able to gather respectable results.

4.1. Organizational Aspects

Inspired by Luke Homann's Innovation Game "Speed Boat" (Homann, 2006, pp. 118-125), we designed thie stage addressing organizational aspects of the study process along six questions shown in Table 1. After having collected all issues that concern students' situation, they used adhesive dots to opt for which issues they deem to be most important to them.

Table 1. Oranizational issues and final result on chart generated by students.

Symbol	Issue	Resulting chart
	Who is an essential part of my study team?	
	What helps me during my studies?	
	What has slowed me down so far?	
	What goals do I want to achieve with my studies?	
	What has helped me emotionally so far?	
	What obstacles do I expect in the future?	

The most important findings out of this stage are:

- Most students see kind and motivated fellow students as an essential part of their team. *This is a positive insight; coming back to the physical classroom after the pandemic can be considered as helpful and should be maintained.*
- Nine students think that having video recordings of the lectures would help them. *Doing this would involve some additional effort on the teacher's side, or preferably some staff support (which unfortunately our institution does not offer). However, we have some experience with recordings. As teachers, we agree that the availability of recordings might help students in case of sickness, and when teachers demonstrate programming details in an integrated development environment, thus enabling them to reproduce the teacher's actions step by step and in one's own speed.*
- Several organizational details were named that brake and slow students down, e.g. issues regarding timetables, equipment in lecture rooms, accessibility of material on the e-learning platform, and problems with registering for electives.
  We will take these to the relevant committees of our faculty and the university.
- Unsurprisingly, the unanimous goal is to pass the first semester finals and then the entire program.
- Students receive emotional support from other students and also from lecturers.
- Eight students have concerns that knowledge gaps from the first semester are obstacles that will lead to problems and additional pressure in later semesters.
  At our institution. exams in the mandatory courses of the first semester must be taken at latest in the second semester. In case of failure, they must be repeated in the semester immediately following the fail. We are aware of these issues. However, removing this pressure makes students push difficult exams further and further to later semesters,

thus incurring the risk to fail the entire program after several years due to a fundamentals class. Nevertheless, this issue is an indication for us to provide more support during the study entry phase.

4.2. Self Competences

At the stage on self competences, students were guided to reflect on 21 items. The results of seven of them are shown in Table 2. We see that guidance is required with respect to self competences such as self-organization. According to these results, students obviously are not sufficiently well prepared by school in these competences.

Table 2. Students findings on their self-competences (n=14, not all answered every question).

	never	rarely	often	always
I do prioritize my daily tasks	2	9	1	1
I consciously start into my free evening	9	3	1	1
I can concentrate on my work	0	4	7	2
I make a note of when I do which tasks	7	4	2	2
I know from which activities I draw my energy	0	3	4	6
I am aware of my competences.	0	4	5	6
I use failures to learn from them.	0	4	9	1

4.3. Personal Issues

The most important findings for us teachers at this stage are that students …

- often do not feel that they are taken seriously enough by teachers..
  This is a serious problem that we need to address at the next lecturers' retreat.
- need sufficient time for themselves e. g. during weekends – which they often do not have.
  This is a tricky issue because the ECTS credit system imposes a certain workload. Especially students with no previous experience in the topics at hand (especially programming skills) need a significant amount of time for their studies.

4.4. Professional Topics (Related to Software Development)

At the stage on technical topics and the content related to software development (the topic of our class), we let students reflect on which technical topics they feel good at, and on which topics the feel less good. As we knew that the cohort is extremely heterogeneous with respect to prior knowledge in programming, we were happily surprised that *all* students claimed to have gained competences in some areas. (So our efforts as teachers were not in vain – every student learned something new and content-related.) To visualize our students' increase in technical competences, we let them draw arrows on a white board representing their estimated learning

progress for three main subject areas of the class. The starting point of each individual arrow represents the corresponding student's initial knowledge prior to the class; correspondingly, the head of the arrow represents the individual student's current skill set with respect to the communicated learning objectives of the course.

Figure 2 illustrates that all students have the impression that they have learned something during the course. The smallest initial skill set and highest increase of skills during the class seems is located in the subject area of testing. Specifically, this indicates that even those students with some initial programming skills prior to the class are not well founded in unit testing – which indicates some improvement potential in the established apprenticeship programs.

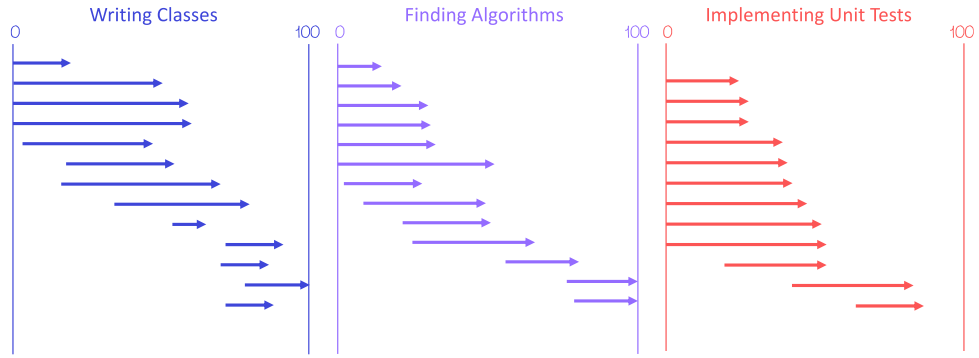


Figure 2. Subject-specific learning progress differentiated by three subject areas (in terms of estimated percentage of learning objective fulfillment).

On the other hand, a couple of students claimed in the second part of this stage that they need to practice more (see Table 3). Therefore, we conclude that we need to increase our efforts to take better account of the differences in prior programming knowledge between students in our cohort. As one active measure to resolve this issue, we already offered additional tutorial lessons for those students that identified themselves as challenged or overwhelmed at the beginning of the semester (as was depicted in Figure 1).

Table 3. Students' self-assessment of professional skills.

	Here, I am good at	**It's okay for me**	**I need more practice**	**Is difficult to me**
Building classes	11	1	1	0
Writing methods	12	0	1	0
Formulating test cases	3	5	6	0
Finding algorithms	4	4	4	1
Detecting bugs	5	4	2	1

5. Conclusions

We introduced a concept of a workshop that enquires into the causes why first-year students struggle with their work in a holistic way and from a student's perspective. In addition to professional issues that students struggle with, we have also included reflection on self-competences and personal issues, and we addressed organizational aspects.

The workshop described in this paper is only one activity to gain insight into student perspective. We accompanied this by interviews with students in first semester and with alumni.

Based on these insights, we are now better able to adjust both our programs and our support offerings to better dedicated support our heterogeneous students in their respective needs.

Acknowledgement

This work was supported by the German Stiftung Innovation in der Hochschullehre (StIL), https://stiftung-hochschullehre.de/, grant no. FRFMM-455/2022. Thank you for your support.

References

Böttcher, A., Thurner, V., Häfner, T. (2020). Applying Data Analysis to Identify Early Indicators for Potential Risk of Dropout in CS Students. *Proc. IEEE Global Engineering Education Conference (EDUCON '20)*, pp. 827–836. url: https://doi.org/10.1109/EDUCON45650.2020.9125378

Carnegie, D. A., Browne, W. N. (2014). Factors that affect the design of a successful engineering programme: A case study, *IEEE Global Engineering Education Conference (EDUCON '14)*, pp. 62-68, doi: 10.1109/EDUCON.2014.6826069.

Falkner, K., Vivian, R., & Falkner, N. J. G. (2014). Identifying computer science self-regulated learning strategies. *Proc. of the 2014 conference on Innovation & technology in computer science education (ITiCSE '14).* ACM, 291–296. https://doi.org/10.1145/2591708.2591715

Hohmann, L. (2006). Innovation Games: Creating Breakthrough Products Through Collaborative Play. Addison-Wesley Professional.

Petersen, A., Craig, M., Campbell, J., Tafliovich, A. (2016). Revisiting why students drop CS1. *Proc. of the 16th Koli Calling International Conference on Computing Education Research (Koli Calling '16)*. ACM, 71–80. https://doi.org/10.1145/2999541.2999552

Simon, Luxton-Reilly, A., Ajanovski, V. V., Fouh, E., Gonsalvez, C., Leinonen, J., Parkinson, J., Poole, M., & Thota, N. (2019). Pass rates in introductory programming and in other STEM disciplines. *Proc. of the Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE-WGR '19*, 53–71. https://doi.org/10.1145/3344429.3372502