Teaching Assistants' Experiences of Tutoring and Assessing in Computer Science Education

Emma Riese
Theoretical Computer Science
KTH Royal Institute of Technology
Stockholm, Sweden
https://orcid.org/0000-0002-4525-3568

Viggo Kann
Theoretical Computer Science
KTH Royal Institute of Technology
Stockholm, Sweden
https://orcid.org/0000-0003-3199-8953

***Abstract*— This Research, Full Paper presents a study on the experiences of teaching assistants (TAs) in Computer Science (CS) courses. TAs are widely used in CS education, to reduce the workload of faculty. A way to allow smaller groups of students in tutorials and individual guidance in lab sessions, even though the numbers of students in the classes are increasing. The aim of this study is to describe the TAs, own experiences of their role, which work tasks they have and how they are carried out.**

Fifteen semi-structured interviews with TAs, who were themselves studying towards a bachelor's or master's degree, were carried out and analyzed using thematic analysis. The results indicate that the TAs typically work alone with preparing the material for the tutorials and that they are given a high amount of freedom both regarding which examples to explain and which pedagogical approaches to use. During the lab sessions the students conduct lots of different tasks, related to assessment and tutoring. In the paper, we present a framework with the identified work tasks, with descriptions on how they were carried out.

The thematic analysis resulted in six role descriptions; as a teacher, friend, tutor, grader, mentor and apprentice. Combining the roles as tutors and graders, can be experienced as tricky, especially if the instructions and grading criteria are lacking details. The TAs can also view themselves as both a teacher and a friend to their students, which could cause conflicts of interest to arise. The TAs could experience to be an apprentice to the course coordinator and senior TAs, but also to be a mentor for the students and more novice TA colleagues. The implications of this study could be of use for faculty who employ TAs in their courses or are conducting or planning TA training sessions. Based on the result, we have made some recommendations for TA training.

Keywords— Assessment, Teaching Assistants, TAs

I. INTRODUCTION

Teaching assistants (TAs), students who are employed by the university to assist the faculty in the courses, are widely used in CS education across many institutions [1, 2]. This is mainly motivated by the fact that using TAs can reduce the workload of faculty and by the same time increase the teacher/student ratio, allowing for smaller groups of students in tutorials and individual guidance in lab sessions [1, 2]. That would be impossible for a single instructor to offer in a large class size [1, 3, 4]. Multiple institutions have reported about their programs that employ TAs [1, 5, 6] but few from the perspective of the TAs.

TAs in CS courses have, however, been studied before and this is a continuation of the study presented in [7, 8, 9] where one of the findings was that our TAs had been offered and participated in little or no pedagogical training. This has been reported to also be the case in other studies [10, 11], even though the TAs also view training as a success factor [12].

At the university where this study took place, a vast majority of the TAs in the CS courses are bachelor's students or master's students. In the more advanced courses also doctoral students could be working as TAs, however, in this study, we only targeted TAs who were bachelor's or master's students.

Employing TAs can reduce the workload for the course coordinator, the faculty member responsible for the course. On the other hand, the use of TAs also implies that course coordinators cannot always be present during all tutorials or lab sessions, especially not when these are run in parallel. Since, the course coordinator is not always present during the learning actives conducted by the TAs, they might not know exactly what is happening during the sessions and how their TAs carry out their work tasks. This research aims to fill this gap and to describe the TAs own experiences of their role and how they carry out the work tasks. The research questions (RQs) addressed in this paper are:

1. Which work tasks do the TAs have, in relation to lab sessions and tutorials, and how are they carrying out their work tasks?
2. Which roles do the TAs perceive themselves to have?

The implications of this study could be of use for faculty who employ TAs in their courses and also for faculty that offer, or plan to offer, a TA training course. By understanding the TA perspective, course coordinators could make more informed decisions when developing material for the TAs and when offering TA training.

II. RELATED RESEARCH

Multiple institutions have reported successful programs of hiring students as TAs within their CS courses [1, 2, 5, 6]. It has been shown that TAs could be a factor for student success [13] and that TAs, who are themselves also students, are more approachable for the novice students compared to the professors [14, 15]. The TAs were also themselves learners in the subject quite recently and therefore believed to be able to relate to students and understand and recognize their struggles and possible misconceptions [15]. On the other hand, previous studies have also shown that TAs are poorly prepared and lack pedagogical training [7, 11, 13]. Training was, however, shown to be a factor that TAs themselves view as important for their professional development [16, 17], which could also suggest that TAs would be willing to participate in more TA training, if offered. Formal mentorship has been suggested as a training method [12], and in one institution positive results with a team teaching approach, in which the TAs worked together in pairs, was reported [12].

It is of course desirable to recruit students with good teaching and communication skills as TAs, and some institutions report on a high interest among the students to become a TA [2, 18]. This is, however, not the case in all universities. Introductory TA training has been suggested as a way to help the new TAs develop skills and competencies needed [10]. Multiple institutions offer TA training [3, 5, 10, 19] in the form of a seminar, course or introduction day. It can also be beneficial for the TAs' development to reflect on their practice throughout the semester. One way to do that is by using a reflective diary [20]. Little is written about how and to what extent TAs communicate with each other and with the course coordinators, but a few studies have stated that it could be done through emails, meetings [1] and chatting tools [7]. A previous study showed benefits in terms of reliability aspects, when the TAs graded assignments in a group setting, rather than individually [21].

The assessments throughout a course allow us to measure and judge the students' skills and knowledge, which is known as summative assessment [22, 23]. The assessments can also be a way to guide and motivate our students towards achieving the intended learning outcomes and aim to provide the students with feedback and support, known as formative assessment [22, 23]. Feedback has also been shown as a factor important for student achievements [24]. Our TAs are typically both conducting summative and formative assessment throughout our courses.

The assessment throughout a course should also be clearly linked to the intended learning outcomes and learning activities of the course, which is referred to as constructive alignment [25]. In the studied courses, this is the course coordinators' responsibility, rather than the TAs. To also make adjustments in the content in such a way that the students experience it to be relevant for them, and to consider their need and motivation for learning how to program, can improve the retention rate in CS [26]. That is, the assessments and learning activities should be experienced as authentic [26].

The TA-student relationship has not been studied much before. However, in school settings the teacher-student relationship has been well studied and it is also recognized as an important factor for students' success [27]. Students in higher education are, on the other hand, not expected to interact with their TAs or faculty as continuously as pupils would interact with their teachers [27]. However, a theoretical model of student retention argued that students who are integrated academically as well as socially, are the ones who will be successful in their studies [28]. Empirical evidence has also showed that the social relationships, between students and between students and faculty, have an impact on the students' sense of belonging and retention rate [29]. How teachers treat their students is also consider as a factor in successful teaching [30].

One teaching approach that can be incorporated during lectures and tutorial in CS is to live-code [31]. To live-code (in this context) means that the instructor will write code, preferably together with students, broadcasting it on the projector and thinking aloud as the code is developed [32]. This approach allows the students to see programming as a process [33] and it has been shown to be an effective and appreciated approach [31, 34].

III. Research Setting

Most of our students are enrolled in some five-year engineering program, studying towards a Master of Science in engineering degree. Typically, all students, regardless of their major, have to enroll in a CS1 course, mainly using the programming language Python, given during their first or second semester. Each program has its own version of the CS1 course. This also allows for adaptations based on the skills and knowledge the students need in the following courses of their specific study program. In their second or third year, they can select a CS2 course and, if they wish, continue to select more advanced CS courses. Students majoring in CS, naturally, have more mandatory courses in CS and also start out using the language Java instead of Python.

In the introductory CS courses, most TAs are undergraduate students or students studying towards their Master's degree, and they are also the sample group in this study. In some more advanced courses, doctoral students could be TAs. For the doctoral students, it is mandatory to take part in a basic training course in teaching and learning, while for other TAs, little training has been offered. During 2018/2019, an initiative to offer voluntary training in assessment, in the form of a 90 minutes workshop, was carried out. All active TAs in CS1 and CS2 courses were invited to participate.

Most of our CS courses have the same course design with lectures, tutorials, and lab sessions, but naturally, the content differs. Each course has a course coordinator, a member of the faculty responsible for the course. The course coordinator is typically also the person responsible for all examinations in the course and all or most of the lectures. The course coordinator is also, typically, the one who recruits the TAs in the course. Currently, this is done on case bases and no standard matrix, such as the one suggested in [4] is used. Each course coordinator is instead free to decide which qualities and skills they deem necessary for their TAs to have and select who to hire for the position. How they recruit TAs can therefore differ, but some common approaches are to send emails to students who performed well in previous course offerings, or to ask students who explicitly expressed interest in the position. Some prospect TAs would also reach out to the course coordinators, applying for a position.

In the lab sessions and tutorials, the focus of this study, the TAs are typically responsible for conducting the learning activities according to the course coordinator's instructions but could also be quite free to construct their own material.

A. Lab Sessions

During a lab session, the students typically work on given assignments and the TAs offer help and guidance on demand.

Our TAs also assist in the grading. The most common approach is by assessing the students' solutions by oral presentations during the lab sessions. However, in some of the courses, the students would instead submit their code using the learning management system, github or via file sharing systems. The TAs would then review the submitted code, grade the submission and leave feedback in written form. Some of the more advanced CS courses use an online judge that runs the programs through a set number of test-cases, to check that the code fulfills the requirements of the functionality of the program [35]. The online judge system is, however, not used in any of the CS1 courses.

B. *Tutorials*

In addition to the lectures and lab sessions, the TAs give tutorials in smaller classes of 12-30 students. The tutorials are either 45 minutes or two times 45 minutes. In the tutorials, the concepts presented during the lectures are put into practice and additional examples are given, walked through step-by-step. The TAs are typically free to choose how they like to carry out their tutorials, only given the topics and some preparation material such as code examples, exercise tasks and presentation slides. In a majority of the courses, the students can choose which classroom to attend and change groups between each tutorial, but in some courses, the students are divided into set groups and have the same TA throughout the course.

IV. Method

In total, fifteen semi-structured interviews were performed with TAs that had been working in different CS courses, from introductory level for non-CS majors to advanced courses for CS majors. This qualitative approach with interviews as the data collection method [36] was chosen because it gives us detailed data consisting of descriptions and explanations from the TAs, without using any predefined frameworks or categorizations. The interviews were made in two rounds, the eight first in late 2017, which were analyzed and presented in [7, 8, 9], and the other seven in late 2019. TAs were recruited on a voluntary basis, where the researcher contacted TAs through email. TAs were recruited with the aim of interviewing a heterogeneous group with respect to the experience of the TAs. In the first round of interviews, we only considered time of experience. In the second round we also considered experience from courses at different levels. In the first round, all interviewees had been TAs in at least one CS1 course. By the time of the second round of interviews, TAs in our CS courses had been invited to participate in the TA training workshop about assessment, and we purposely contacted TAs from both the group of participants and non-participants. Four of the respondents had participated in the workshop.

It was completely voluntary for the TAs to take part in the study. All participants were given detailed information about the aim of the study before the interviews started. They were informed that participation was completely voluntary and that they at any time, without stating a reason, could decline to answer a question or discontinue their participation. The participants were also informed about who was going to have access to the raw data that was collected; solely the first author of the paper (since the second author was or had been a teacher of some of the TAs who were interviewed, their names and other identification were removed before it was shared, to protect their privacy). All participants gave their consent to be part of the study and agreed to have their interview audio recorded.

The set interview questions asked were:

1. *Would you like to start by describing what you do as a TA? Walk me through your work tasks.*
2. *How do you usually do when assisting and helping a student/student pair during a lab session? Can you describe such a situation?*
3. *If you assess students' solutions, can you describe such an assessment situation?*
    - *How do you decide if a solution is approved or not?*
    - *If a student does not pass, how do you tell the student?*
    - *Do you also give feedback to your students, if so in what way?*
    - *Which are your experiences and thoughts about plagiarism among your students?*
4. *How do you know what material to cover in a tutorial?*
5. *What freedom to design the exercises do you have?*
6. *How do you prepare?*
7. *What kind of material do you get from the course coordinator and what material do you construct?*
8. *What help and support do you usually get from the course coordinator?*
    - *Do you feel it is easy to get in touch with the course coordinator if needed?*
9. *What help, if any, do you usually get from other TAs?*
10. *How do you communicate with the other TAs and the course coordinator?*
11. *Have you received any training as a TA? In that case, regarding what?*
12. *If you attended the TA training workshop about assessment, did you learn anything from it, and what is your recollection of it? (Only asked in the second round of interviews).*
13. *Would you have liked to have more TA training? In that case, regarding what?*
14. *If you got to wish freely, is there anything you would wish for that could make your TA job easier or more fun?*

The interviews were transcribed and analyzed using a thematic analysis, following the six steps in [37]. The thematic analysis was conducted by the first author and validated by the second author of the paper. All interviews were conducted in Swedish, and the presented quotes have been translated to English by the first author.

V. Results

This section presents the results for each of the two research questions separately.

RQ1: How do the TAs carry out their work tasks during or in relation to lab sessions and tutorials?

Presented below is a framework with the identified themes of each of the work tasks from the thematic analysis. It is divided into two main categories: *lab sessions* and *tutorials*. An overview of the framework is presented in Table 1.

TABLE I. OVERVIEW OF IDENTIFIED WORK TASKS

TA's Work Tasks

A. Lab sessions
  1) Conducting assessments
    a) Prepare by reading up on or constructing requirements
    b) Check requirements
      - Test run code or use online judge
      - Review code for requirements on code structure
    c) Check that the students can explain their code
    d) Check for suspicious plagiarism
    e) Make a decision
    f) Communicate results to students and provide feedback
  2) Helping the students
    a) Prepare to become able to answer questions
    b) Find out what the students need help with
    c) Provide the right amount of help
    d) Adjust the feedback and help to the students

B. Tutorials
  1) Preparing material
  2) Conducting tutorials
    a) Give lectures/presentations
    b) Live-code
    c) Answer questions
    d) Supervise learning activities

A. *Lab sessions:* The work the TAs conduct in and around our lab sessions can be divided into two main sub themes: *conducting assessments* and *helping the students*. Under each category, the TAs view and description of their work tasks are described.

1) Conducting assessments: One of the overarching themes is that the TAs conduct assessments. This theme is deivided into six subthemes, each is presented below.

a) Prepare by reading up on or constructing requirements: To be able to grade the assignments, the TAs would typically read through the instructions given to the students. Most TAs tried to do this before the lab sessions started, but some also stated that they read up on the assignments on the go, when the students asked to present their solutions. When the TAs found the instructions to be lacking detail, which was quite often, they could also construct requirements on their own, based on their own perception of what a good solution would look like. Preparing for the lab sessions was something that the TAs typically did individually, and not together with other TAs.

b) Check requirements: This theme has two subthemes, presented below.

- *Test run code or use online judge:* An important part of the assessment is to make sure the program has the required functionality. This was typically done by testing the code with some input or by using an online judge system [35]. In some courses, the TAs could get examples on test cases that should be run, but more commonly, they came up with the input themselves.
- *Review code for requirements on code structure:* If the assignments had requirements on code structure (for instance that a class must be used) this would also be checked. The TAs stated that in some courses, the requirements on how the code should be written were not written in the instructions, but rather in between the lines. The TAs would in those cases make their own interpretations on what the requirements were, and also construct their own grading criteria.

c) Check that the students can explain their code: The TAs described that the largest part of the assessment was usually to get the students to explain the code they have written. To make sure that students know how the code works and that they could explain what happens when the code is executed. This part could also involve making the students explain their design choices and data structures used. If the assessment was conducted solely on submitted code, rather than through oral presentations, this step was skipped.

d) Check for suspicious plagiarism: The TAs explained that checking for plagiarism is also part of their work task. However, which strategies the TAs used could differ. Some TAs consider the fact that students could explain their code as sufficient to justify that they had also written the code themselves, while others also asked questions concerning the development process.

e) Make a decision: Ultimately, points b)-d) would result in a decision on whether or not the students would pass the assignment. The TAs would most of the time make this decision by themselves, but could also ask other TAs or the course coordinator for a second opinion, if needed. Typically, they would then try to ask a person (either a TA or the course coordinator) in the lab, but in some courses they could also use a live chat-tool. It was also described to be somewhat hard to find someone to ask, since typically everyone would be busy.

f) Communicate results to students and provide feedback: Once the decision has been made, the TAs would have to communicate that to students. The TAs could have different strategies to do that and experienced it to be a bit more uncomfortable when giving the students a failing grade, especially if the students got upset or even started crying. TAs would try to praise the things that were carried out well and be very specific in what was missing. If a student passed the assignment, the TAs typically also tried to find something that could be improved and give advice that could help the students with the next assignment or when programming in general.

2) Helping the students: The second overarching themes is that the TAs are helping the students. This theme is deivided into four subthemes, each is presented below.

a) Prepare to become able to answer questions: The TAs would prepare in a similar manner for a lab session with focus on helping the students, as for a lab session with focus on assessing the students' presentations. They would mainly go through the instruction of the assignments the students are working on. In addition, some TAs stated, they could try to solve the assignments for themselves in order to better foresee where the problems might arise for the students.

b) Find out what the students need help with: In order to be able to help the students, the TAs need to understand what the students need help with. This could be tricky, since the students do not always know exactly what is wrong or how to explain that to the TA. Typically, the TAs would let the students explain, ask questions, and let the students show them their code or error messages.

c) Provide the right amount of help: The TAs generally tried not to give away the entire answer to the problem but rather to guide the students to find the answer for themselves. Not ever *touching the keyboard* was a rule that was repeated throughout the interviews. However, this could also cause some frustration among the students, who might expect the TAs to be more helpful and on point with exactly how to solve the issue at hand.

d) Adjust the feedback and help to the students: The TAs stated that they tried to adapt to the situation in the lab. How busy the lab was, and how stressed they were could determine how long they could spend explaining something for a student. If time was short, the guidance could be to go back to the lecture notes, search the internet for an example or to just give away the answer after some explanation.

B. Tutorials

The following section presents the results from the thematic analysis regarding the TAs' work tasks during and in relation to tutorials. Under each theme, the TAs' view of their work task and description of how they carried out the work task is described.

1) Preparing material: In order to be able to conduct a tutorial, the TAs, quite naturally, needed to prepare. All the interviewed TAs stated that they would mainly do this alone. Depending on the content, previous experience and material given by the course coordinator, how and what material the TA prepared could be very different. Some TAs would construct one or two examples to live-code, some would prepare a slide-show, some would read through all lecture notes and build on the examples from there, and some would simply go through the instructions from the course coordinator to be prepared to improvise during the tutorial. The TAs experienced lots of freedom, both regarding which examples and material to use and which pedagogical approaches to use.

2) Conduting tutorials: Four subthemes regarding the execution of the tutorials were found in the thematic analysis, presented below.

a) Give lectures/presentations: The most common technique described was to give presentations or lectures, explain concepts and use either the black board to write notes or to show slides using the projector.

b) Live-code: To write code together with students, to live-code, was the second most common technique. The TAs would typically bring some notes with a key or general idea to the tutorial and start from there. Some of the more experienced TAs could also improvise and make up examples on the spot together with the students.

c) Answer questions: To let the students ask questions was something described as a natural part of any tutorial. Some TAs reflected that the questions asked could be on various levels and some well outside the scope of the course. Typically, the TAs would try to keep the most basic level and only cover more advanced topics if the time allowed towards the end of the tutorial or in the break for interested students.

d) Supervise learning activities: Some TAs described that they on some occasions had been given specific exercises to carry out with the students. It could for instance be analog Parson problems [38] or quizzes. The TAs would then be the facilitator of such learning activities by instructing the students, setting the time frames and concluding/summarizing the exercises.

RQ2: Which roles do the TAs perceive themselves to have?

The thematic analysis resulted in six themes, related to the TAs own perception of their roles: as a teacher, friend, grader, tutor, mentor and apprentice. An overview of the six role definitions and their relationship to each other is shown in Fig. 1. The mentor-apprentice roles should be viewed as a dimension on their own, where the TAs could be either a mentor or an apprentice, but also have both of the roles at the same time. The four other roles are related to each other and as shown in Fig. 1. The role of the teacher is most apparent, but it intersects with the three others. The teacher-friend roles are on the same axle, and the roles grader-tutor on a separate one. It is possible for the TAs to identify with all four roles, but also any subset of the four. Each role is described in more detail and with example quotes below.

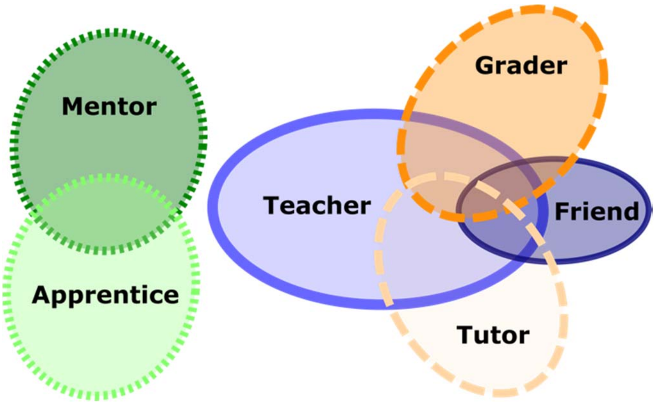

Fig. 1. Shows the six identified roles from the thematic analysis and their relationships to one another.

1) The Teacher: The strongest and most apparent role of the TAs is the one as a teacher. This was especially obvious when the TAs talked about their role in relation to the tutorials. Planning the content, giving lectures and trying their best to explain the key concepts. Feeling the responsibility to teach the "right" and relevant content, but also to do it in a way that the students really understood. Some of the more experienced TAs had also been given the opportunity to help develop the courses and to construct assignments and design exam questions, which further strengthened them in their role as a teacher. Conducting assessment, helping, and guiding students to pass the assignments were also tasks that would typically fall within this description.

> *"So it can exist two different ways to solve the same thing and then it is really enough that you know only one, well it is never wrong to learn something extra, but it is more [alarming] if it was something missed. So then they can sit there in the computer lab and have lots of trouble because you did not have the time to go through it."* - TA for 4.5 years, interviewed 2017.

2) The Friend: The TAs did not view themselves to be only teachers, but also friends to their students. Being an older student, who might just have taken the course one year prior to becoming a TA, they also consider the students to be their peers. The TAs reflected that this could also make them more approachable for the students. Multiple TAs described how students in their courses reached out on social media platforms or found them in the library when they were themselves studying. Some TAs stated that they also hang out together with some of their students or previous students outside class on activities not related to their TA role. The TAs would in some instances view themselves as primarily a friend to the students, rather than their teacher.

> *"Well, all of my old students usually greet me when I walk by and stop me and we talk for a while. So for me, yes I hang out with my students sometimes. But I always adhere to the requirements when assessing. I am very strict with that. Or well, not very strict but I always stick to them ... I have had to fail friends too".* - TA for 3 years, attended TA training workshop about assessment, interviewed 2019.

3) The Grader: Being a grader and having the power to decide if a student should pass or fail an assignment or a presentation was described to be a common role for the TAs. This role was experienced as more formal and strict compared to the role of the tutor. In the role of a grader, the TA would have to make decisions based on requirements and to handle the situation in a proper, correct manner, according to the practice. More restrictions applied to this role compared to any of the others.

> *"Yes, I assess presentation too and it is perhaps somewhat of a different mode. Then you are a little bit tougher as well. Then you expect, at least I expect a little bit more from the students."*- TA for 6 years, interviewed 2017.

The role of the grader was experienced to be tricky to combine with the role of being a friend to the students, and some of them also expressed that it could be uncomfortable to fail a student, especially one you know well. Some TAs also explicitly stated that if they were friends with a particular student in the class, they would not grade their assignments, while others said they would grade them but compensate by being even more strict (see example quote in previous section).

4) The Tutor: In contrast to the role of a grader, the TAs also expressed their role to be the one of a tutor; supporting and helping the students to pass the assignments and exams. They viewed themselves as someone who should guide the students in the right direction, and be there, both as a teacher and friend, and help them solve whichever problem at hand.

> *"If I immediately realize what is wrong, I try to work around the issue. So a lot of it is for them to understand the basics. Such as if they are going to create objects, they know that objects are created or they know such details. Because sometimes it's like they only get help and help and help, and they do not understand what has happened. Like people have coded for them or something like that. So it's important to know that they understand the basics. If they do not understand the foundation, then I explain that. Otherwise, I try to go into what they ask for and help them with that."* - TA for 4 years, attended TA training workshop about assessment, interviewed 2019.

5) The Mentor: The TAs experienced to be a mentor or even a role model for their students. The TAs expressed a sense of pride when they could see the students reviewing the material they developed and taught during the tutorials. This TA-student relationship was described as stronger and more apparent in the courses where the TAs were responsible for a set group of students and could more easily follow their progress.

> *"On the other hand, I'm happy when I see that the students use my material from tutorials. I also see that as a success. I see that they have started from my code skeleton. Like,*

> *when I have made an example that is based on their lab assignments. I can the see that they have not changed certain variables that they use, and then I think it has really been useful.*" - TA for 3 years, interviewed 2017.

The TAs, especially the senior ones, also experienced to be, or become, somewhat of a mentor for other TAs. Someone who could assist, answer questions and step in when the course coordinator was not present.

> *"Now there are other assistants who come and ask me [for help] and such."* - TA for 4 years attended the TA training workshop about assessment, interviewed 2019.

6) The Apprentice: The TAs expressed to be the apprentice to the course coordinator and carrying out the work tasks according to their instructions. They could also see themselves as an apprentice to the more experienced TAs, and sought their help and approval when feeling unsure of how to assess a particular solution or how to handle a situation. In some rare cases, the TAs also expressed that they could learn from their students who were even more experienced in programming than themselves, and in that way also become an apprentice to them.

> *"Yes but I think it can be good to bounce questions that you have and be able to ask, well, how do you think about this? Especially when you are new it feels like it's even more important to be able to ask questions. You learn a lot from it and to be able to see how others work and so on."*- TA for a semester, did not participate in the TA training workshop, interviewed 2019.

VI. Discussion

The purpose of this study was to describe the TAs' own experiences of their work task, how they were carried out and how the TAs perceive their role as a TA. From the results it is apparent that our TAs have lots of different work tasks (RQ1) and also perceive themselves to have different roles (RQ2).

In Table 1 we have presented an overview of TAs' work tasks, which were shown to be quite many, especially concerning the lab sessions. To the authors, it is quite surprising that the TAs stated to prepare for the lab sessions and tutorial individually, given the fact that they are typically a group of TAs facing the same challenges. Communication between TAs and between course coordinators and TAs, have, however, been found be lacking before [7] but we would have hoped for an improvement in this area in the second round of interviews. Another alarming finding is that the TAs also construct their own grading criteria regarding the code structure and that they come up with own test-input, when the instructions were lacking these details. This could be a threat to the reliability of the assessments. Previous research has also shown that grading in a group setting could improve the reliability of the grading [21], but almost all of the assessments described were conducted individually by a single TA. The results also indicate that our TAs mostly seem to stick to traditional lecturing when conducting tutorials, and few other approaches than lecturing and live-coding [34] were mentioned. This could be due to the fact that our TAs do not have to enroll in any TA training, and they might not be interested or know how to look up alternative approaches, and then only use the ones they are familiar with.

Regarding the lab sessions, one of the most challenging tasks was described to be determining how much help to provide a student, and how to give help in an appropriate way. This could also be quite a balance for our TAs, especially given the fact that they will also be the ones grading the assignments. If they give away the answers or too much guidance, it could be harder to assess what the students have actually written themselves and if they have the knowledge and skills to reproduce the code. This also goes hand-in-hand with the results of the second research question, that the TAs experience to be both tutors and graders, and that these roles are not mutual exclusive.

To further discuss our second research question, regarding which roles the TAs have, the most interesting result, to the authors, is that the TAs can view themselves as both a teacher and a friend to their students. It has previously been shown that the social relationships, between students and between students and faculty, have an impact on the students' sense of belonging and retention rate [29]. The TA-student relationship could possibly have similar effects, it has previously been shown that TAs could be a factor for student success [13] and also stated in prior research that the TAs are more approachable for the students than their professors [14, 15]. The findings in this study, strengthen this claim since the results show that TAs view themselves as closer to the students and more approachable to them. However, these closer friendships also open up for conflicts of interest situations to arise, especially if the TAs are also grading the students they primarily view as friends. Some of the interviewed TAs stated that they try to avoid these situations by not assessing their friends, but the results show that this was not the common praxis. Understanding the complexity of the different roles the TAs are experiencing is an important step to also be able to support the TAs in their professional development.

That our TAs are perceiving their role to include a mentorship-apprentice dimension, is not surprising and well aligned with previous research [7, 12].

A. Limitations

The participating TAs were recruited to the study on a voluntary basis. The found themes are not necessarily generalizable to the whole TA population, which is rarely the case with qualitative research [36]. The results of this study should be seen as a first step to further understand the TA perspective, both when it comes to how they are carrying out their work tasks and how they view their roles. Further research, in a quantitative manner [36] is necessary to further validate these results.

B. Lessons learned/implications - a training course for TAs

Based on the results, we recommend that all TAs should be offered and participate in an introductory TA training course, to support and communicate both expectations and best practices among the TAs. Based on the results, we recommend such course to consist of five modules:

1. Responsibilities, legislations and code of conduct
2. Techniques and tools when working during a computer lab session
3. Techniques and tools for conducting tutorials

4. Assessment – feedback and grading
5. Reflection and discussion with other TAs

The first module regarding responsibility, legislations and code of conduct should include clear instructions on what applies for a TAs whose friend is enrolled in the course they are teaching. It should also include the university's rules on plagiarisms and how such an errand is handled.

The second module should include practice and techniques used, or suitable to be used in a lab session. It could for instance include techniques on time-management, stress management, and guidelines on how to provide the appropriate amount of help to the students.

The third module should include practice and techniques used, or suitable to be used in tutorials. This could include, but not be restricted to live-coding [32], peer-instruction [39], and lecturing techniques.

The fourth module should include practice and techniques for assessment in CS, including exercises on how to interpret and use grading criteria [40]. Formative and summative assessment [22], and reliability and validity are also key concepts that TAs should be introduced to.

In the fifth module the TAs should have to reflect, discuss and test what they learned. This can for instance be done by providing some tricky and realistic cases to discuss, such as how to handle a very upset student who you just failed. The TAs should also have the opportunity to try to prepare a tutorial together, rather than on their own. This could be done by given the TAs a topic and some time to plan a student activity (not just a lecture/presentation), and then present their plan to the other TAs.

Furthermore, we would like to propose the following recommendations for course coordinators:

- In addition to the introductory training, the TAs should be given some custom training during the semester, or at least an opportunity to reflect on their practices and discuss together with their colleagues. It could be through keeping a reflective diary [20] and by hosting meeting. Suggestions for topics could be mentoring, diversity in CS and interpretations of grading rubrics/grading criteria used. This could facilitate professional development and sharing of best practices which could also benefit the experienced TAs.

- Give the TAs guidelines and/or grading rubrics for each lab they are assessing. If the TAs should assess more than the code, the TAs should be provided with questions to ask the students and guidelines on how to assess the students' answers. From the results it is clear that our TAs constructed criteria on their own, which can make the criteria differ between the TAs.

- Organize meetings with your TAs to go through the assessment criteria and lab instructions as well as following up previous lab sessions. Even the best grading rubrics can contain ambiguous phrasings [40], and to provide a safe space for your TAs to discuss them, can result in better reliability in the assessments.

- Organize formal mentorships for TAs (as also suggested in [12]), so that each new/junior TA have one senior TA to turn to for help. Mentoring a junior colleague should be a defined work task for senior TAs. Even though our more experienced TAs seemed happy to take on the role as a mentor to their junior colleagues without being appointed, formalizing this could be a way to both reward and acknowledge the senior TAs, and making the junior TAs more comfortable to ask for help and advice.

- Set up a communication channel for you to communicate with your TAs and for TAs to communicate with each other in real time, that can also be active during lab sessions.

C. *Future plans*

Since the spring semester 2020, it is mandatory for all our TAs to participate in a TA training course, using a blended learning format [41], in which face-to-face interactions and online modules are used, following the set-up and recommendations presented above. Module 1-3 are online modules while module 4 is in the form of a workshop, and module 5 is a face-to-face discussion seminar. After the semester, we will summarize the course evaluations and follow-up with the TAs who participated. This would also allow us to investigate how generalizable the identified themes are within our TA population and if we can see any effect of the introductory training. Another interesting follow-up study would be to interview course coordinators and compare their expectations of their TAs with how the TAs view their role and work tasks.

VII. Conclusions

Based on 15 interviews with TAs in computer science courses, we have presented a framework over the TA's work tasks related to lab sessions and tutorials (see Table 1), including descriptions on how the TAs carried out these work tasks. We have also identified six roles that the TAs experienced to have, as a: teacher, friend, grader, tutor, mentor, and apprentice. These findings provide us with a description of what is happing in the classrooms and computer labs when we as course coordinators are not present. These findings can also further guide how we as faculty and employers of TAs can support and train our TAs in their professional development in order to facilitate learning among our students, taking into account the different roles the TAs are experiencing to have. It is important to remember that TAs are themselves students, and not necessarily trained as teachers. Based on the results, we have outlined a proposal for an introductory TA training course, consisting of five modules: "Responsibilities, legislations and code of conduct", "Techniques and tools when working during a computer lab session", "Techniques and tools for conducting tutorials", "Assessment – feedback and grading" and "Reflection and discussion with other TAs". We have now, very recently, started to offer introductory TA training according to our proposed modules, to our TAs and plan to evaluate it after the semester.

References

[1] D. Mirza, P. T. Conrad, C. Lloyd, Z. Matni, and A. Gatin, "Undergraduate Teaching Assistants in Computer Science: A Systematic Literature Review," in *Proceedings of the 2019 ACM*

Conference on International Computing Education Research - ICER '19, 2019, pp. 31–40.

[2] E. Roberts, J. Lilly, and B. Rollins, “Using undergraduates as teaching assistants in introductory programming courses: an update on the Stanford experience,” *SIGCSE Bull.* 27, 1 (March 1995), pp. 48–52.

[3] J. Forbes, D. Malan, H. Pon-Barry, S. Reges, and M. Sahami, “Scaling introductory courses using undergraduate teaching assistants,” *Proceedings of the 2017 ACM SIGCSE Technical Symposium on computer science education*, 2017, pp. 657–658.

[4] D. Leyzberg, and J. Lumbroso, “Nailing the TA interview: using a rubric to hire teaching assistants,” in *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education,* 2017.

[5] R. Brent, J. Maners, D. Raubenheimer, and A. Craig, “Preparing undergraduates to teach computer applications to engineering freshmen,” in *2007 Frontiers in Education Conference (FIE)*, 2007.

[6] P. E. Dickson, T. Dragon, and A. Lee, “Using Undergraduate Teaching Assistants in Small Classes,” in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE ’17). Association for Computing Machinery, New York, NY, USA,* 2017 pp. 165–170.

[7] E. Riese, “Teaching Assistants’ Experiences of Lab Sessions in Introductory Computer Science Courses,” in *2018 IEEE Frontiers in Education Conference (FIE)*, 2018.

[8] E. Riese, and V. Kann, “Teaching assistants’ experience of their roles and responsibilities in relation to tutorials,” in *KTH SoTL 2019*, 2019.

[9] E. Riese, and V. Kann, ”Erfarna lärarassistenters upplevelser av övningar i inledande programmeringskurser,” (in Swedish) in *7:e utvecklingskonferensen för Sveriges ingenjörsutbildningar*, 2019.

[10] F. J. Estrada, and A. Tafliovich, “Bridging the gap between desired and actual qualifications of teaching assistants: An experience report,” *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education,* 2017.

[11] V. Muzaka, “The niche of graduate teaching assistants (GTAs): Perceptions and reflections,” *Teaching in Higher Education*, vol. 14, no. 1, 2009, pp. 1–12.

[12] E. Patitsas, “A case study of the development of CS teaching assistants and their experiences with team teaching,*”* in *Proceedings of the 13th Koli Calling International Conference on Computing Education Research*, 2013, pp. 115–124.

[13] S. E. Dechenne, N. Koziol, M. Needham, and L. Enochs, “Modeling Sources of Teaching Self-Efficacy for Science, Technology, Engineering, and Mathematics Graduate Teaching Assistants,” *Cbe-Life Sciences Education*, vol. 14, 2015 no. 3.

[14] A. Decker, P. Ventura, and C. Egert, “Through the Looking Glass: Reflections on Using Undergraduate Teaching Assistants in CS1,” in *Proceedings of the 37th SIGCSE Technical Symposium on Computer Sci-ence Education (SIGCSE ’06).* ACM, New York, NY, USA, 2006, pp. 46–50.

[15] Y. Ren, S. Krishnamurthi, and K. Fisler, “What Help Do Students Seek in TA Office Hours?,” in *Proceedings of the 2019 ACM Conference on International Computing Education Research (ICER ’19)*. Association for Computing Machinery, New York, NY, USA, 2019, pp. 41–49.

[16] F. Marbouti, K. Rodgers, H. Jung, A. Moon, and H. Diefes -Dux, “Factors that help and hinder teaching assistants’ ability to execute their responsibilities,” in *Proceedings of the 120th ASEE Annual Conference and Exposition*, 2013.

[17] A. C. Moon, H. Jung, F. Marbouti, K. J. Rodgers, and H. Diefes-Dux, “Undergraduate and graduate teaching assistants' perceptions of their responsibilities - Factors that help or hinder,” in *2013 IEEE Frontiers in Education Conference (FIE)*, 2013, pp. 1576-1578.

[18] A. Vihavainen, T. Vikberg, M. Luukkainen, and J. Kurhila, “Massive increase in eager TAs: experiences from extreme apprenticeship-based CS1,” in *Proceedings of the 18th ACM conference on Innovation and technology in computer science education* (*ITiCSE ’13*). Association for Computing Machinery, New York, NY, USA, 2013, pp.123–128.

[19] H. Danielsiek, J. Vahrenhold, P. Hubwieser, J. Krugel, J. Magenheim, L. Ohrndorf, D. Ossenschmidt, and N. Schaper, “Undergraduate teaching assistants in computer science: Teaching-related beliefs, tasks, and competences,” in *2017 IEEE Global Engineering Education Conference (EDUCON)* 2017, pp. 718–725.

[20] M. Ukrop, V. Švábenský, and J. Nehyba, “Reflective diary for professional development of novice teachers,” in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education - SIGCSE '19*, 2019, pp. 1088–1094.

[21] B. Harrington, M. Ahmadzadeh, N. Cheng, E. H. Wang, and V. Efimov, “TA marking parties: Worth the price of pizza?,” in *Proceedings of the 2018 ACM Conference on International Computing Education Research*, 2018, pp. 232–240.

[22] M. Taras, ”Assessment–summative and formative–some theoretical reflections,” *British journal of educational studies*, 2005 53(4), 466-478.

[23] D. D. Dixson, and F. C. Worrell, “Formative and Summative Assessment in the Classroom,” *Theory Into Practice*, 2016, 55(2), pp.153–159.

[24] J. Hattie, “Identifying the salient facets of a model of student learning: a synthesis of meta-analyses,” *International Journal of Educational Research*, 1987, vol. 11, pp. 187-212.

[25] J. Biggs, “Enhancing teaching through constructive alignment,” *Higher education*, 1996, 32(3), 347-364.

[26] M. Guzdial, *Learner-centered design of computing education : research on computing for everyone*. 2016.

[27] G. Hagenauer, and S. Volet, “Teacher-student relationship at university: An important yet under-researched field,” *Oxford Review of Education*, 2014, 40(3), 370—388.

[28] V. Tinto, “Dropout from higher education: a theoretical synthesis of recent research,” *Review of Higher Education*, 1975, 45, pp. 89-125.

[29] L. Strauss, and J. Volkwein, “Predictors of Student Commitment at Two-Year and Four-Year Institutions,” *The Journal of Higher Education*, 2004, 75(2), pp. 203-227.

[30] K. Bain, “What the Best College Teachers Do,” Cambridge: Harvard University Press, 2011.

[31] A. G. S. Raj, J. M. Patel, R. Halverson, and E. Rosenfeld Halverson, “Role of Live-coding in Learning Introductory Programming,“ in *Proceedings of 18th Koli Calling International Conferenceon Computing Education Research*, Koli, Finland, November 22–25, 2018 (KoliCalling ’18).

[32] J. Paxton, “Live programming as a lecture technique,” *Journal of Computing Sciences in Colleges* 18”, 2 2002, pp. 51–56.

[33] J. Bennedsen, and M. E Caspersen, “Revealing the programming process,” in ACM SIGCSE Bulletin, Vol. 37. ACM, 2005, pp. 186–190.

[34] M. J. Rubin, “The effectiveness of live-coding to teach introductory programming,” in *Proceeding of the 44th ACM technical symposium on Computerscience education*. ACM, 2013, pp. 651–656.

[35] E. Enstrom, G. Kreitz, F. Niemela, P. Soderman, and V. Kann, “Five years with kattis - Using an automated assessment system in teaching,” in *2011 Frontiers in Education Conference (FIE)*, 2011.

[36] M. Borrego, E. P. Douglas, and C. T. Amelink, “Quantitative, qualitative and mixed research methods in engineering education,” *Journal of Engineering Education*. 2009, 98(1), pp. 53–66.

[37] V. Braun, and V. Clarke, “Using thematic analysis in psychology,” *Qualitative research in psychology*, 3(2), 2006, pp. 77-101.

[38] D. Parsons, and P. Haden, “Parson’s programming puzzles: a fun and effective learning tool for first programming courses,” in *Proceedings of the 8th Australasian Conference on Computing Education-Volume 52*. Australian Computer Society, Inc., 2006, 157–163.

[39] L. Porter, C. Bailey Lee, B. Simon, Q. Cutts, and D. Zingaro, “Experience report: a multi-classroom report on the value of peer instruction,” in *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education*, pp. 138–142. ACM, 2011.

[40] Y. M. Reddy, and H. Andrade, “A Review of Rubric Use in Higher Education,” *Assessment & Evaluation in Higher Education*, vol. 35, no 4, pp. 435–448, 2010.

[41] N. D. Vaughan, M. Cleveland-Innes, and D. R. Garrison, *Teaching in blended learning environments : creating and sustaining communities of inquiry.* 2013.