Chapter 8

Evaluating Didactic Approaches used by Teaching Assistants for Software Analysis and Design using UML

Teaching assistants (TAs) are employed for supporting various activities of a course, from assisting with course material, grading and, predominantly, bridging the interaction between teachers and students. This role is even more significant in large courses with dozens of students, particularly during lab sessions with practical assignments. Therefore, here we explore how the role of the TA supports students to overcome the challenges they face when learning software design. During a bachelor course on Analysis and Design, we collected data from i) weekly interviews with TAs, ii) student's assignment hand-ins and iii) the corresponding feedback reports written by TAs to students. By analysing these three different sources we propose guidelines and practices for effective deployment of TAs.

This chapter is based on the following publication: Dave R. Stikkolorum, F. Gomes de Oliveira Neto, and Michel R.V. Chaudron. Evaluating didactic approaches used by teaching assistants for software analysis and design using uml. In *Proceedings of the 3rd European Conference of Software Engineering Education*, ECSEE'18, pages 122–131, New York, NY, USA, 2018. ACM

8.1 Introduction

Teaching assistants (TAs) support various tasks in a course, such as grading assignments, lecturing, supporting students during lab sessions, advising students, supervising projects, etc. Consequently, assigning TAs to these different activities inherently becomes part of the course planning activities performed by a course coordinator (usually faculty members at a university). When addressing student's needs related to feedback and guidance in courses with practical activities (such as programming and modelling), the responsibilities of a TA involve lecturer-like duties in order to guide students towards their deliverables.

In our practice we often use TAs to support students during practical group work. We believe that the use of teaching assistants in a software analysis and design (SAD) course can be of great value. We see the fact that their reasoning approach is close to the target student, hence bridging further the lecturers and the students to improve knowledge transfer. TAs are easy to approach and can relate to students' study experiences [95].

The aim of this report is to study the didactic approaches of TAs during the practical sessions in a software design analysis and design course. We aim to obtain insights into what particular elements of the TAs intervention help students further with the development of their software design skills. With this insight we would have a better understanding of students' reasoning and common difficulties during their learning process. In addition it can support the development of future training programs for TAs. Based on our experience from a software analysis and design course, we aim to answer the following research questions:

- RQ1: What are typical challenges in 'Software Analysis and Design' for which students seek TA support?
  - RQ1.1: Do students focus on different matters than the topics in the lectures?
- RQ2: Which didactic approaches do TAs use to support the challenges students have?
  - RQ2.1: Are there approaches that are specifically useful for Software Design contexts?
- RQ3: Which expectations do TAs have about students and vice versa?
- RQ4: Do feedback and guidance of the TAs focus on the same matters as the lecturers?

In this chapter we present and discuss the results of a case study that was conducted during an 'Analysis and Design' course in the second semester of the first bachelor study year. We collected data from student surveys, the TA feedback in the online learning system and from the weekly TA-teacher interviews during the course.

By answering the research questions we contribute to the understanding of the interaction between students and TAs. In addition, we contribute to the existing investigation of student's common difficulties in UML Analysis and Design assignments (i.e. [79] [132]). With our research we confirm and extend these studies. Moreover, this research also provides a basis for checking whether the TAs address the students' problems.

The remainder of this chapter is organised as follows: in Section 8.2 we discuss related work, in Section 8.3 we present the research method that we used, subsequently in Section 8.4 we present the results that are discussed in Section 8.5. Section 8.6 consists of recommendations for future deployment of TAs. Threats to validity are discussed in Section 8.7. We conclude and suggest future work in Section 8.8.

8.2 Related Work

In this section we discuss related work. In our research we aim to reveal the benefits of the intervention of a TA in a Software Engineering (SE) lab context. To our knowledge there is little published about the use of TAs during SE lab sessions. Specifically, research about what specific interventions are observed in the context of Software Analysis and Design, is not available to our knowledge.

To provide more background on the use of TAs we explore further in this section: i) general use of TAs and ii) use of TAs in a CS context.

8.2.1 General Application of TAs

Because of a growing employment of TAs in higher education in the UK, Park [99] distils lessons for future TA usage in the UK. He studies North American publications about the use of graduate students for teaching undergraduates. The literature explored different TA activities including lab demonstrations, practical classes, leading tutorials and seminar groups, and lecturing. Benefits that the use of TAs could bring include: reduction of teaching load, funding for postgraduate research students, teaching experience for the TA, apprenticeship model for future professors. The author presents a summary of lessons learned with a special focus on the following issues: i) selection and preparation, ii) training, iii) mentoring, iv) practical/personal issues, and v) professional development. Interesting to mention is that the summary also includes the remark that some students have difficulties with the ambiguity of their role – teacher versus student; employee versus apprentice.

Hardré et al. state that institutions need effective and efficient ways for the the development of TAs teaching skills, but that many TAs are not prepared enough for teaching responsibilities [44]. They propose the use of instructional design (ID) as a tool for the development of a TAs teaching expertise. ID should support TAs in thinking and planning like expert teachers, which would help them improve instructing undergraduates.

Hardré et al. explore what contributes to the development of a teaching assistant through a two-day professional development workshop [45]. The study involved 210 TAs and focused on general TA development. The study involved three clusters (hard sciences, social sciences, and arts & humanities). They found that TAs are aware that training can contribute to their learning and development of teaching. Features of the workshop that TAs value the most in relation to their development were: expertise of speakers, structural design of events, and quality of support materials. Eighty percent of TAs reported to be motivated to continue learning about instructional theory and practice. The authors suggest future work related to the degree of autonomy of the TA, the role, influence and effort of the supervisor and suggest to compare experience between different disciples.

8.2.2 TAs in Computer Science

Danielsiek et al. present a preliminary competence model for the training of undergraduate TAs (UTA) for computer science students [30]. Next to practical benefits, such as dealing with the increasing numbers of students, the authors point out the positive influence on student retention. The authors also point out there is little attention to UTA training in the community so far (2017). With the evaluation of their prototype training course they expose the different competences and perceptions UTAs have. In some of the observed courses more teacher-centered beliefs are observed whilst in others more student-centered are observed. Their analysis of applying their training course shows a positive change on the student-centred beliefs of the students.

Patitsas discusses a case study of TAs in computers science lab sessions [101]. The author comments there is not much literature on lab TAs (as of 2013). The author found that the following factors contribute to TA development: practice, teaching multiple courses, mentoring, staff meetings, team teaching and feedback. The team teaching is valued highly by the TAs, since it enabled them to learn from each other. For labs, the TAs most benefit from course-specific experience from others than general teaching experience.

8.3 Research Method

In this section we explain our overall research approach, how we collected the data, the course and assignments we used in our case study, the participants of our case study and how we analysed the data.

8.3.1 Overall Research Approach

In order to answer our research questions, we conducted a case study during a course for a bachelor programme in Software Engineering and Management (SEM) at the University of Gothenburg (Gothenburg, Sweden). The course has 63 registered students and is taught in the second term, for first-year students. The students have a basic knowledge about object-oriented programming from a previous course in the first term. The course covers object-oriented analysis and design, has two teachers and five TAs. A summary of the course activities is illustrated in Figure 8.1. Given that our main interest is related to TA activities, we rely on particular course elements to conduct the study, in particular: (i) feedback by the TAs on assignment submissions, (ii) supervision sessions with everyone involved in the course, and (iii) weekly meetings with TAs to verify course progression and students' performance. Besides the activities described above, we use two instruments to collect data: (i) interview sessions with the TAs, and (ii) questionnaires answered by students. Consequently, we perform qualitative analysis in our data to gather insights on the trade-off of using TAs in a software design course. Our data analysis and collection approach is based on triangulation. The collection and analysis of the data is discussed in Sections 8.3.2 and 8.3.5. An overview of the overall research approach is presented in Figure 8.1.

8.3.2 Data Collection

For our research we collected data from the following three sources: An *online survey* – four times, after every assignment, we asked the students about their experiences with the TAs. We focused the questions on particular topics of the specific assignment and on how the TA that was involved helped to understand the topic better. The survey consisted of a non-free text part and a free text part. The questions are shown in Table

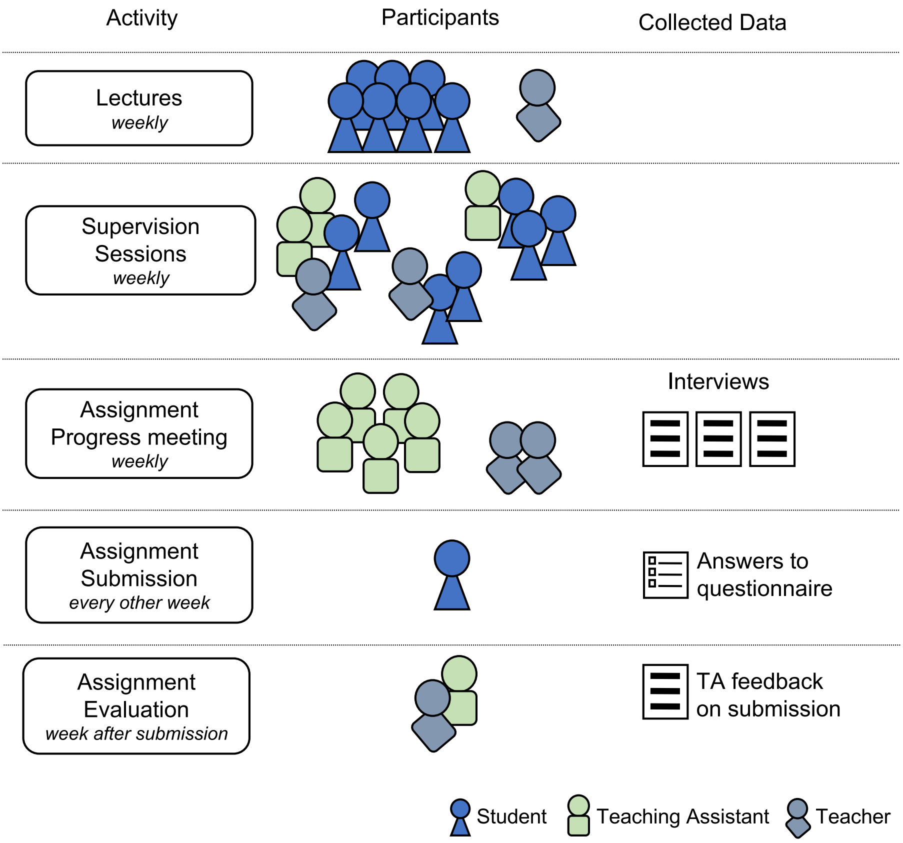

Figure 8.1: *Summary of course activities and data collection.*

8.1. Limesurvey[1], an online tool, was used for this purpose.

A *structured TA interview* – every week, during the course, the lecturers interviewed the TAs. This interview was structured targeting topics such as: i) problems with students, ii) observations and iii) discussions about the approach. We used an audio recording application on a mobile phone to record the interviews.

TA feedback text on hand-in – according to feedback guidelines provided and defined by the lecturer, each TA wrote down feedback on the hand-in for two groups per assignment. The online learning platform was used to give feedback on the hand-in's. This was written feedback and was given every other week.

[1] https://www.limesurvey.org

Table 8.1: *Overview of students' post assignment questions – non free and free text*

Non-free Text Questions

During the assignment

Participation in supervision sessions

> Q1: *Did you participate in supervision?* [Yes | No]

Students' opinion about the TA support

> Q2: *How do you feel about the support of the teaching assistant(s) during this assignment?* [Likert scale 1-5: not helpful - very helpful]

Difficulties with which the students were helped by the TAs.

> Q3: *With what difficulties did the teaching assistants help you with?* (for each assignment a more specific, task related, version) [list of choices, multiple answers allowed]

After handing in the assignment

Usage of the feedback by the students

> Q5: *Did you look at the feedback the teaching assistant gave with the grade?* [Yes | No]

Opinion about the learning fullness of the TA support

> *Q6: What do you think about the feedback of the teaching assistant after grading?* [Likert scale 1-5: did not learn - learned a lot]

Free Text Questions

During the assignment

> Q1: *Can you provide detail about how a teaching assistant helped you out with a specific topic?* (for each assignment a specific, task related, version) [free text]

After handing in the assignment

> Q2: *Can you give an example of feedback of the evaluation that really helped you to understand the topic better?* [free text]

8.3.3 Course and Assignments

The course that was used for the case study was a 1st year bachelor's object-oriented analysis and design course. The course targets analysis and design techniques using UML. The typical UML diagrams that were taught were: Use Case Diagram, Class Diagram, Sequence Diagram and State Machine Diagrams. The course consisted of lecture hours (2 hours per week) and supervision hours (2 hours per week). During supervision students can ask questions about the practical assignments they have to hand in and also about the previous lectures.

Three of the practical assignments were based on a software development project that was running in another course in the same teaching period. The fourth assignment was part of only the analysis and design course. The practical assignments had to be handed in every other week using an online learning platform for submission.

The case that was handled in the project course was a complex case (development of a robot car control system) that required a large amount of modelling. For the practical assignments the students delivered reports in which they presented their UML models for the different development phases of the software development project. In total there were four assignments:

- Assignment 1: Use Cases and Domain modelling with class diagrams – Students create an analysis report, using use case diagrams and class diagrams, that explained and motivated analysis decisions.

- Assignment 2: System Sequence Diagram and Use Case Realisation – Students create an initial design report, using sequence diagrams. Students motivated their design decisions.

- Assignment 3: State Machine Diagram and Sequence Diagram Simulation – Students create a report that focused on detailed behavioural design and testing by simulation.

- Assignment 4: Refactoring – Students delivered a report covering the refactoring of a game. The source code was given. The students had to redesign the reverse-engineered class diagram. The new design should be improved by using design patterns. Eventually the students implemented the code and delivered a running game.

8.3.4 Participants

Students – the students that participated were 1st year bachelor students following the 2nd semester of their SE program. In the 1st semester they learned about object-oriented programming in Java. The students worked in a group of 6 or 7 students during the whole course. During the course 10 groups were active.

TAs – Five teaching assistants were elected from the second year of the bachelor program. They were selected based on their grades for prior courses and information from the lecturers that experienced them in the classroom. TAs should apply for the position. The TAs task was to support the students during supervision hours and give feedback on the hand–in's that students uploaded on the online learning platform. They receive a small payment for the job.

Lecturers – 2 lecturers were involved, teaching the course and having the weekly interviews with the TAs.

8.3.5 Data Analysis

For analysing the data we used both a quantitative and a qualitative approach. First, for the survey questions, we used descriptive statistics for analysis. Second, the open ended answers on the survey, the transcriptions from the TA interview meetings and the written TA feedback on the student hand-ins were analysed by labelling the text using codes. The codes were distilled from proof reading the recorded text with the research questions in mind. The coding was done by one person. Examples of these codes were: *challenges using TA* (challenges identified during TA supervision), *observation* (remarkable observation) and: *approach* (approach a TA uses to help a student). After labelling we merged the labelled text from the different sources (survey,interview, TA feedback) into categories. The results of both approaches are presented in Section 8.4.

8.4 Results

In this section we present the results we obtained from the student survey, the written teaching assistant assignment feedback, and the interviews with the TAs. First we present the descriptive statistics. Second, we present the data for the open ended questions in the form of tables organised in categories.

8.4.1 Descriptive Statistics

In this section we show the statistics that were distilled from the non-free text questions in the student survey (Table 8.1). After each of the 4 assignments we conducted a post-assignment questionnaire. There were 3 moments in time. Due to scheduling challenges, the questions of assignment 3 and 4 were combined into 1 questionnaire. Each questionnaire consisted of a set of similar questions. A questionnaire consists of a part *during the assignment'* and a part *after grading*. We present the results of the 3 questionnaires below in order of the assignments. The number of responses (N) can differ. This is because of questions where multiple answers were allowed or where students skipped questions.

Assignment 1 – After assignment 1 36 students (N=38) responded that they participated in the supervision sessions. Most students (N=29) value the support of the TAs higher than 2 (neutral or helpful) in this assignment (see Figure 8.2).

Table 8.2 shows the challenges where students asked for help from a TA for the Use Cases tasks of assignment 1. Table 8.3 shows the challenges for the domain modelling with class diagrams task.

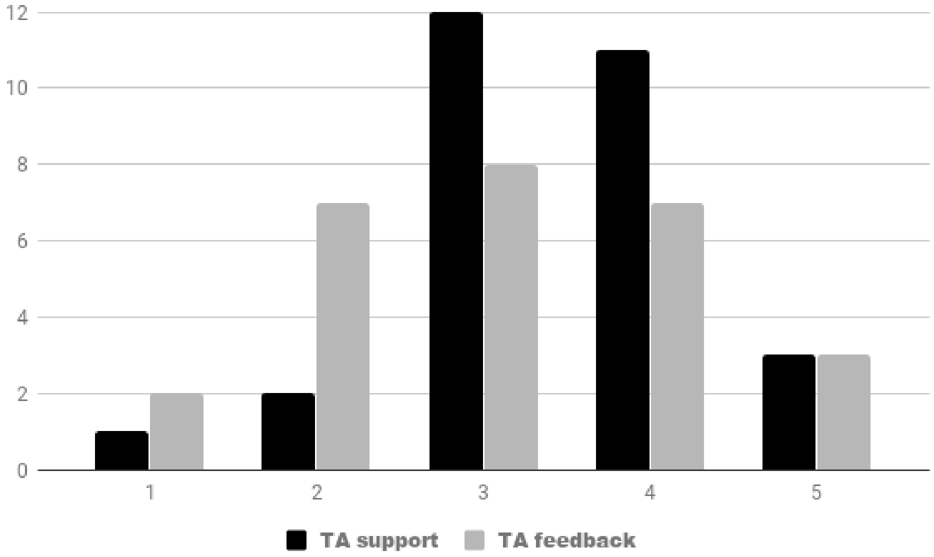

Figure 8.2: *Students' judgement about the quality of the TA support and quality of the TA feedback for assignment 1. A 1 indicates low quality, a 5 high quality.*

Table 8.2: *Difficulties During the Use Case task – assignment 1*

Difficulty	frequency
Apply Include or Extend	11
How to apply an include or extend relationship in a use case diagram	
Finding Use Cases	8
Finding use cases from the assignment's case text	
Finding Domain Concepts	4
Finding concept from the domain from the assignment's case text	
Finding Actors	4
Finding actors from the assignment's case text	
Using the Tool	3
The use of the UML modelling tool	
Other	10
Removing actors from the diagram	
Understanding the tasks and what to do/hand in (7x)	
Reviewing the diagram	
Deleting some extra actors from the diagram	

Table 8.3: *Difficulties During the Domain Modelling Task (Class Diagram) –assignment 1*

Difficulty	frequency
Finding the problem domains	6
Determining the problem domains in the assignment	
Respresent concepts with classes	5
Determining the main concepts and represent them in the diagram	
Finding Relationships	9
Finding relationships between concepts from the assignment's case text	
Choose Between Class or Attribute	5
Determining whether something is a main concept or property of a concept	
Finding attributes	1
Finding operations	1
Using the tool	2
The use of the UML modelling tool	
Other	4
Explained how thorough it was supposed to be	
Understanding the assignment	
Understanding that in this case they represented hardware connections	
The overall finished diagram needed help to be refined	

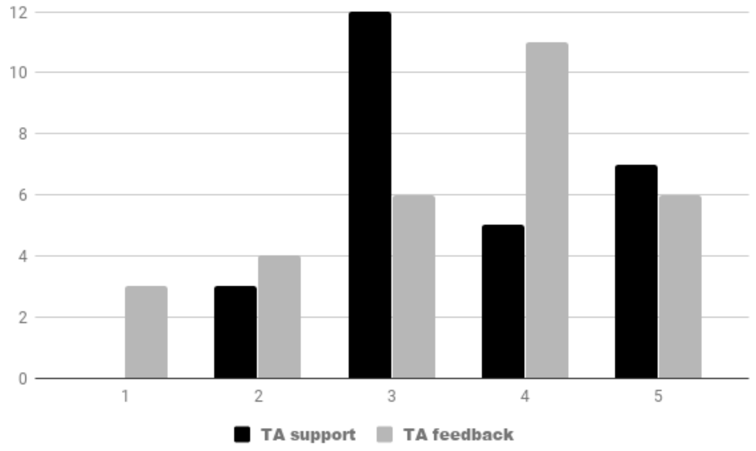

Figure 8.3: *Students' judgement about the quality of the TA support and quality of the TA feedback for assignment 2. A 1 indicates low quality, a 5 high quality*

Table 8.4: *Difficulties During the System Sequence Task – assignment 2*

Difficulty	frequency
Seeing the system as a black box	10
Making a top level diagram *High abstraction at system level without implementation detail*	9
Use case to sequence *Identify sequences from the use case that was distilled from the case*	8
Using the tool *The use of the UML modelling tool*	8
Finding methods	7
(A)synchronous messages *When to use (a)synchronous messages*	4
Apply reply messages *When to use reply messages*	4
Finding the central lifeline *Identify the central lifeline that represents the system*	3
Other *what was exactly required (2x)* *overall structure*	3

Table 8.5: *Difficulties During the Use Casse Realisation Task – assignment 2*

Difficulty	frequency
Finding the important objects that collaborate in a use case realisation	11
Identify concepts that play a role in the solution (design)	
Finding methods for internal system sequence	6
Assigning responsibilities to the concepts that were found	
Creating an Initial Design	6
Updating the Initial Design	6
Finding relationships	5
Having a white box view	5
Understanding how the internal system should work	
Apply Reply Messages	4
When to use reply messages	
Using The Tool	3
The use of the UML modelling tool	
(A)synchronous Messages	1
When to use (a)synchronous messages	
Other	3
Deciphering requirements	
Understanding the purpose of the diagram fully and its abstraction level	
Explaining what they were	

After grading, 25 of the students (N=28) responded that they looked at the feedback of the TA. In addition, 2 were informed about the feedback by a group member. They (N=27) seem to be more critical on judging that they learned from the TA feedback, compared to their judgement about TA the support. On average they judged neutral (see Figure 8.2).

Assignment 2 – After assignment 2, 37 students (N=42) responded that they participated in the supervision sessions. Most students (N=27) value the TAs neural or helpful (>2) in this assignment. (Figure 8.3).

Table 8.4 shows the challenges were students asked for help from a TA for the System Sequence Diagrams task of assignment 2. Table 8.5 shows this for the Use Case Realisation task.

27 students (N=30) stated that they read the feedback on the assignment. In addition, 3 were informed about the feedback by a group member.

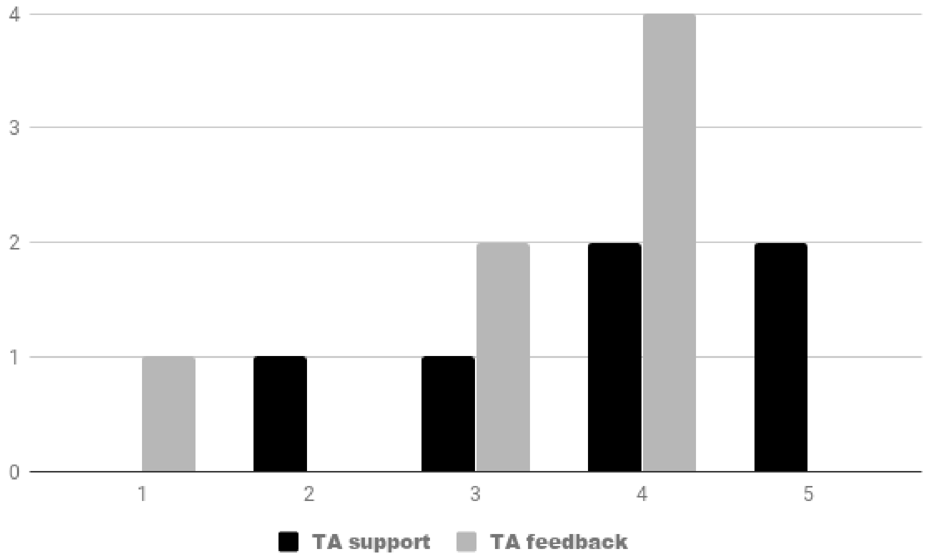

Figure 8.4: *Students' judgement about the quality of the TA support and quality of the TA feedback for assignment 3. A 1 indicates low quality, a 5 high quality*

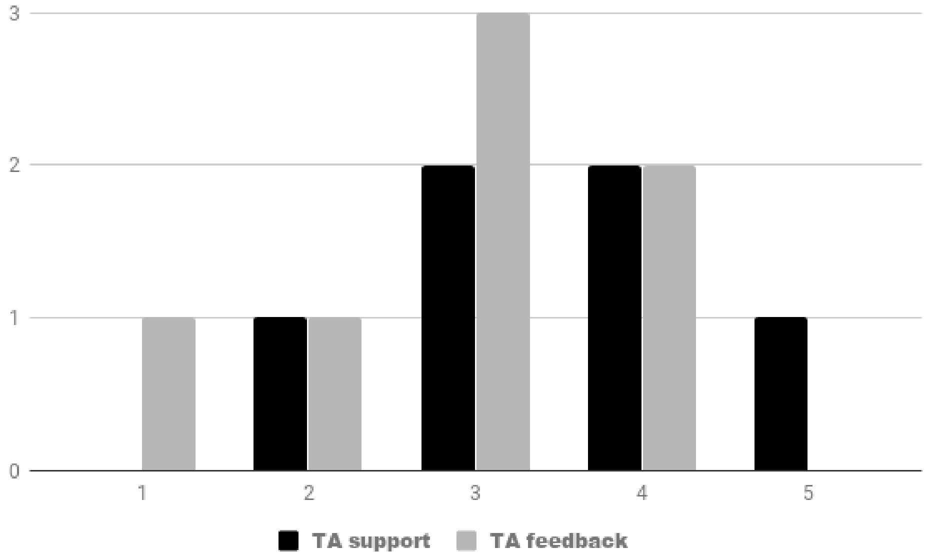

Figure 8.5: *Students' judgement about the quality of the TA support and quality of the TA feedback for assignment 4. A 1 indicates low quality, a 5 high quality*

Table 8.6: *Difficulties State Machine Diagram–assignment 3*

Difficulty	frequency
Identify State behaviour *Recognize state behaviour from the assignment text*	3
Identify events *Recognize events from the assignment text*	2
Identify States *Recognize states from the assignment text*	1
Identify Transisitions *Recognize transisitions from the assignment text*	1
Apply Guards *When to use guard notation (for conditions)*	1
Apply Proper Naming	1

Table 8.7: *Difficulties During the Sequence Diagram Simulation Task – assignment 3*

Difficulty	frequency
Using (A)synchronous Messages *When to use (a)synchronous messages*	3
Identify Important Methods *Recognize implementation responsibilites in the design*	2
Apply reply messages *When to use reply messages*	1
Identify Collaborating Objects *Recognize implementation concepts for the design*	1
Placing States on Lifelines *Adding state behaviour to a sequence diagram*	1

Most students (N=27) value the TAs' support as helpful or as very helpful (>3) in this assignment. (Figure 8.3).

Assignment 3 and 4 – The survey for assignment 3 and 4 were conducted combined. 10 students (N=14) confirmed they have participated in the supervision hours. 6 students judged the support during the supervison hours of assignment 3 or 4. Most judgements were neutral or positive. There was no very negative judgement (1) (Figures 8.4 and 8.5).

Table 8.6 shows the difficulties during the State Machine Diagram task. Table 8.7 for the Sequence Diagram Simulation task. Table 8.8 shows the challenges were students

Table 8.8: *Difficulties During Refactoring–assignment 4*

Difficulty	frequency
Apply a Class Design Pattern *How to apply a design pattern on class level (non-architecture level)*	3
Apply an Architecture Design Pattern	1
Getting Runnable Code *Update the source code based on design, compile and run*	1
Other *Suggestions for further patterns that could be applicable with our current one*	1

asked for help during the Refactoring task.

7 students (N=8) stated to have read the feedback on the assignment. Most students (N=7) value the TAs' support neutral or helpful (>2) in both assignment 3 and 4. The students judged that they have learned more from the TA feedback of assignment 3 (Figures 8.4 and 8.5).

8.4.2 Open Questions, TA Feedback and Interviews

This section presents the answers to the open questions in the students' post questionnaire, the written TA feedback on the hand-in's of the students and the data that was distilled from the interviews we had with the TAs. This section shows results on mainly two matters: i) an inventory of common difficulties were TA support was involved and ii) explore what type of approaches on guiding and feedback are used by the TAs.

Table 8.9 shows an overview of the difficulties that were mentioned by the students in the questionnaire (free text in Table 8.1). Next to what the students mentioned, we also list the topics that the TAs observed.

Table 8.10 shows a categorised list of the approaches that were used by the TAs during the 4 different assignments. An 'X' indicates the presence of that particular approach.

Table 8.9: *Overview of common difficulties mentioned in free text – an 'x' indicates that the students mention the difficulty in the text of the survey responses or that the TAs mention this as an issue in the interviews or the feedback*

	Student				TA			
Common topic	A1	A2	A3	A4	A1	A2	A3	A4
P Process / Motivation / Confidence								
P.1 Confirmation of direction	x							
P.2 Clarification of task and deliverable	x							
P.3 Individual participation					x	x		
S Software Development Related								
S.1 Analysis vs Design	x	x						
S.2 User Stories vs Use Cases	x				x			
U UML Related								
U.1 Abstraction / level of detail	x	x		x	x	x	x	x
U.2 Structure of diagram	x				x			
U.3 Responsibility (identify elements)	x	x	x		x	x	x	
U.4 Use of notation / concepts	x	x			x	x	x	x
U.5 Layout					x	x		
U.6 Pre / Post conditions					x	x		
U.7 (In)consistency			x			x	x	
D Deliverable / reporting								
D.1 Improve documentation	x				x	x	x	x
D.2 Clarification of task / deliverable	x				x	x	x	x

Student = from student questionnaire - TA = from TA interviews and written feedback
An = assignment n

8.4.3 Mapping Students' Difficulties and TA Approaches

In this section we relate the common difficulties that students have to the approaches TAs use to help students.

From the student questionnaires we identified the common difficulties (Table 8.9). The TA approaches were distilled from the interviews and the written feedback to the students (8.10). We present a mapping in Tables 8.11 and 8.12.

The rows in Tables 8.11 and 8.12 represent students' difficulties and the columns correspond to TAs approaches. An 'x' in a cell indicates that we see a potentially use for a TAs approach for a particular difficulty a student has. A 'ⓧ' indicates a difference between the students view and the TAs view. For example, when a 'ⓧ' occurs in the TA view this means that a certain TA approach is not mentioned by the student, while

Table 8.10: *Overview of the different approaches TAs use*

TA Approach Related	A1	A2	A3	A4
I Informative				
I.1 Invite students for next activity	X			
E Examples (comparison/clarification)				
E.1 Explain problem with the use of an example	X	X	X	X
E.2 Showing study material (slides, book, etc.)		X		
E.3 Suggest to read literature, slides and instructions		X		
E.4 Compare with programming		X		
E.5 Rephrase the assignment grading criteria		X	X	
E.6 Let the student think aloud			X	
G Guiding (not directly give the answer)				
G.1 Suggest direction for improvement	X	X	X	X
G.2 Point out errors/omissions	X	X	X	X
G.3 Point out particular notation features	X	X		
G.5 Ask questions to get student in right direction	X	X	X	
G.6 Give confirmation to student when on the right track		X		
G.7 Discuss the problem in an intense way		X		X
G.8 Suggest to use the richness of UML to improve readability			X	
G.9 Tips on how to approach the assignment				X
A Active / passive support				
A.1 Approach student actively by asking questions	X			
A.2 Invite student to contact a TA or lecturer when insecure		X	X	
A.3 Send extra feedback after group meeting		X		
R Revealing the student's problem				
R.1: Backup with other TA	X			
M Medium used				
M.1 In person communication		X		
M.2 Written feedback		X	X	X
C Character of TA				
C.1 Always willing to help out		X	X	X
C.2 Pointing out the TA's job is support not to solve the problem		X		
C.3 Thorough Explanation			X	
C.4 Make assignment him/herself for understanding the student				X
T Time				
T.1 Suggest follow up meeting after evaluation feedback	X	X	X	X
T.2 Look at deliverable before actual hand-in	X			

An = assignment n

Table 8.11: *Mapping of difficulties and possible approaches – student view*

	E1	E2	E3	E4	E5	E6	G1	G2	G3	G5	G6	G7	G8	G9
P1	ⓧ						ⓧ	ⓧ	ⓧ	ⓧ				
P2	ⓧ									ⓧ				
P3														
S1	ⓧ	ⓧ	ⓧ	ⓧ			ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ		
S2	x						x	x	x	x				
U1	x	x	x	x			x	x	x	x	x	x		
U2	x						x	x	x	x				
U3	x	x	x	x		x	x	x		x	x	x		
U4	x	x	x	x			x	x	x	x	x	x		
U5														
U6														
U7	x					x	x	x		x			ⓧ	
D1	x						x	x	x	x				
D2	x						ⓧ	ⓧ	ⓧ	x				

Table 8.12: *Mapping of difficulties and possible approaches – teaching assistant view*

	E1	E2	E3	E4	E5	E6	G1	G2	G3	G5	G6	G7	G8	G9
P1														
P2														
P3	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ		ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ		
S1														
S2	x						x	x	x	x				
U1	x	x	x	x		ⓧ	x	x	x	x	x	x		
U2	x						x	x	x	x				
U3	x	x	x	x		x	x	x		x	x	x		
U4	x	x	x	x		ⓧ	x	x	x	x	x	x	ⓧ	
U5	ⓧ	ⓧ	ⓧ				ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ		
U6	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ		ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ		
U7	x	ⓧ	ⓧ	ⓧ		x	x	x	ⓧ	x	ⓧ	ⓧ		
D1	x	ⓧ	ⓧ				x	x	x	x	ⓧ		ⓧ	
D2	x	ⓧ	ⓧ		ⓧ					x		ⓧ		ⓧ

from the TA's perspective the approach is actually identified.

We mapped common difficulties with the approaches of categories E (Examples) and G (Guiding) because those categories are more specific about how the approach was performed.

When looking at difficulty D2 across tables, we notice that students seem to have a preference to behave passively, while the TA is concrete suggestions of next steps. Examples of these approaches are G1 and G2. In these cases students aim to finish the task without to much mental effort. Instead, the TAs are expected to literally state the next step. In contrast, the TAs do not mention approaches were they provide concrete solutions to be helpful.

When looking at difficulties D1 and D2 across tables, we find that TA's use approaches in which they try to motivate and activate students, such as E2 and E3. For example, they provide suggestions for reading or studying a particular topic rather than concrete solutions. The TAs also mention that discussing the problem helps. During a discussion the TA and the students elaborate a certain topic. Instead of leaving the group after giving them guidance, the TA stays for discussing a particular topic.

When comparing tables on difficulty U7, a technical oriented challenge, it can be noticed that from a students view the TA approaches E2, E3 and E4 are not mentioned as helpful. This is contrast to what the TAs have used or discussed in the interviews to be helpful. E2, E3 and E4 are TA approaches in which examples are used to guide the student further in the assignment. The students don't mention this as helpful support or wanted to have more concrete answers.

Also across tables, we notice that at difficulties P3, U5 and U6 students did not made remarks about any supporting TA approach. This appears to be in contrast to what the TAs intended.

8.5 Discussion

In this section we discuss with the aim to answer the research questions listed in the introduction of this chapter.

8.5.1 What are typical challenges in ‘Software Analysis and Design’ for which students seek TA support? (RQ1)

Based on the results from the questionnaire we identified the challenges where students asked for TA support. On one hand, it seems that the most dominant topics are the ones that require more abstract reasoning, such as ‘Finding relationships’ or ‘Seeing the system as a black box’. On the other hand, it is worthwhile to mention two other matters. First, in the ‘other’ option students often mention that they needed help to ‘understand the task’ or ‘what we were supposed to do’. The students, who are novice in software designing, need guidance and confirmation. They expect the TAs to pick up this (lecturer) role. Second, we noticed that the ‘using the tool’ option was not chosen that much. From research [2] we know tools can frustrate students a lot (installing, configuring, crashes etc.). We did not put special attention to the use of (industrial) tools [83]. Therefore we expected to have collected more tool-related issues. It could be that the TAs, that were walking around during the supervision sessions, helped with small tool use problems at a time and it was not noticed as a TA support challenge in the questionnaire.

From the analysis of the open ended questions we found similar challenges. Although these challenges were more described in an abstract way than specific. For example: ‘Use of notation’ versus ‘Apply guards’.

We noticed differences between which challenges the TAs noticed and which ones the students noticed (RQ1.1). Table 8.9 shows, that for some of the topics, the TAs kept noticing the need for support. At the same time the students do not notice this in the open questions in the questionnaire. Although students do not mention this in their feedback, this does not mean they did not receive support for this topic. It still gives the impression that TAs try to continue focusing on important matters such as ‘Improve documentation’ and ‘Use of notation’, while at the same time the focus of the students seem to shift.

We did not find any unexpected challenges or reasoning, such as matters that wouldn’t arise in a lecturer-student conversation. The identified challenges and the fact that students seek for confirmation, confirm and extend our previous research [132]

8.5.2 What kind of didactic approaches do TAs use to support the challenges students have? (RQ2)

From the free text answers on the questionnaires and the transcribed interviews with the TAs we were able to identify and categorise approaches TAs use when helping out students during the supervision sessions as presented in Table 8.10. We see that the guiding approaches are quite dominant in every assignment. Especially 'suggestion of direction for improvement' and simply 'point out errors' are mentioned in every assignment as helpful. This is in line with findings from Hattie et al. They describe feedback to be most useful when one combines direct task feedback with feedback that is focused on the process [49]. As most of the approaches could be expected in different types of programs (i.e. non IT), we see that typically 'explaining with the use of an example' is useful for SE students (RQ2.1). This also counts for the approaches that help the students 'in the right direction' We see software design as an iterative activity. It is important that we, and in this case the TAs, help students out by reflecting on their intermediate results. Especially novices are insecure about their solutions and need some or more guidance in the design process.

There seems to be a mismatch between the activating approach the TA most of the times chooses and the more passive attitude that a student can have. While TAs try to encourage and activate students by showing examples and pointing to lecture materials, the student wants to have a 'yes, this is good' or 'no, this is wrong' answer. This is probably due to the novice stage of the student. This means that a student is not yet capable of associating the fresh material with examples or other sources. The TAs themselves give a suggestion to cope with this problem. They mentioned discussion helped the students to discover bits of the solutions. Maybe a brief discussion prior to suggestions for further reading or looking at examples could be a good approach. We believe that the TAs' active learning approaches could benefit students' learning. Freeman et al. state active learning increases exam performance [39].

8.5.3 Which expectations do TAs have about students and vice versa?(RQ3)

From the results that summarise the attitude of the TAs and the students we identified dominant issues that are addressed from both the TAs and the students.

First, from the TAs perspective we noticed that TAs mention different matters about the study behaviour of the students. They noticed the students do not use the scheduled supervision time to it's full extent. They expect students to take part in all the supervision hours. They explain it by the fact that "students underestimate the time" and that some students "are more into programming at the moment". The TAs were critical on the students by noticing some students had questions that should have been addressed in the lecture and stated they did not visit that lecture. TAs expect students to elaborate more on possible solutions before saying something is good or bad. In some cases TAs also had difficulties stating something was good or bad because of the variety in possible answers. With respect to the time spend, the TAs noticed students allocated more time half way the course.

Second, from the students' perspective, the students mention that they have difficulties with the different answers TAs give when asking for help. For the students the TAs sometimes could have different interpretations of the assignment goals. This seems to be typical for software engineering. Also students are critical on the feedback of the TA. They have a need that TAs say if it is good or bad. While at the same time the TA tries not to give them the answers right away.

8.5.4 Do feedback and guidance of the TAs focus on the same matters as the lecturers?(RQ4)

The students' difficulties that the TAs act on (Table 8.9) and approaches TAs use (Table 8.10) in this study reflect on what was discussed in the weekly meetings and instructions. There seems to be a shared focus with the lecturers. Interestingly for some topics there was no particular instruction, but they seem to be important for the TAs. We also find these topics of importance. The TAs mention the following:
Layout TAs value to spend time on the quality of the layout of a diagram. For them to give proper feedback, it is necessary to have a layout that is easy to read.

UML Syntax Students seem to not look into syntax / read about UML (because of less questions about it).

Prior programming knowledge TAs explained the root of some student struggles:"Probably students make mistakes because of knowledge of Java classes." In our previous research [132] we also found evidence that prior programming knowledge also can lead to difficulties in understanding software design.

Quality of Reporting although instructed properly and discussed during supervision TAs notice the quality of the reports.

Responsibility TAs mention different matters that worry them in relationships to the students' progress: the difference in individual participation; a coming deadline while students focus on - in their eyes - wrong priorities.

8.6 Recommendations for Future Deployment of TAs in SAD Courses

In this section we present recommendations for future use of TAs in courses similar to the 'Analysis and Design' course that was observed. Based on the analysis we performed in this research we list several recommendations.

8.6.1 Elicitation / TA Profile

When selecting the TAs lecturers should be aware of the different skills that are needed next to the technical (UML) skills. As Table 8.10 presents, TAs use different methods to support the different questions that come from the students. Having a mix of TAs that use different approaches is an ideal situation. Interview TA candidates and address the different approaches. For example TAs could react on challenges in a mini case.

8.6.2 Training and Support

We endorse the need for training TAs [30]. In our opinion, training depends heavily on the level of the course, the TAs and what (educational) task the TAs should fulfil. From a course content perspective one could use and extend our categorised challenges that are listed in the figures in Section 8.4 Table 8.9. From an approach perspective Table 8.10 can be extended and used to train student support approaches.

We recommend to support the TAs during the course by having weekly meetings and discuss student problems, solve organisational issues and discuss example solutions.

8.7 Threats to Validity

In this section we discuss the threats to validity of this research.

Construct Validity Closed ended questions. The questionnaire could be biased because of the pre-filled question answers. We tried to avoid this by adding an 'other' option amongst the answers.

Internal Validity Recorded free text and transcribed interviews. It is not proven that we recorded everything that was relevant. While answering the questionnaires, students may have forgotten some details of the TA interactions that could have been significant to report. Those issues can possible be missed. Also there could be an interpretation validity threat when we were analysing the data because of incomplete or unclear student statements.

External Validity Matters such as cultural background, student level (TA and course student) and the university's methods, could influence the generalisability of this study.

We plan to address these threats as we refine the study framework in future studies.

Conclusion Validity We base our conclusion on the evidence that was distilled from the questionnaire (closed ended and open ended questions) and interviews. There is a risk of not having recorded all the relevant information. We tried to avoid this by having multiple sources that contain relevant information. Most of the observations were found in multiple sources.

8.8 Conclusion

This chapter reflected on the use of TAs in order to: i) reveal interventions (approaches) TAs commonly use ii) extend our knowledge about common difficulties students have with analysis and design topics and iii) what believes are shared and valued useful by TAs and lecturers.

By continuously observing the TAs and students during an analysis and design course we were able to distil and categorise the common challenges students have recorded where TAs supported them. Further we have an overview of the typical interventions TAs use when helping students during their assignment tasks.

With this research we now have an approach that can be used to further research the use of TAs. What is learned can be used for improving training programs for TAs. Future work comprises further studies with other instances of the courses, and correlate with connected courses that use OO analysis and design knowledge. Moreover, we plan to investigate whether the same findings can be applied to TAs of different courses in Software Engineering.