

WS 2012/2013

Advanced Topics in Software Construction

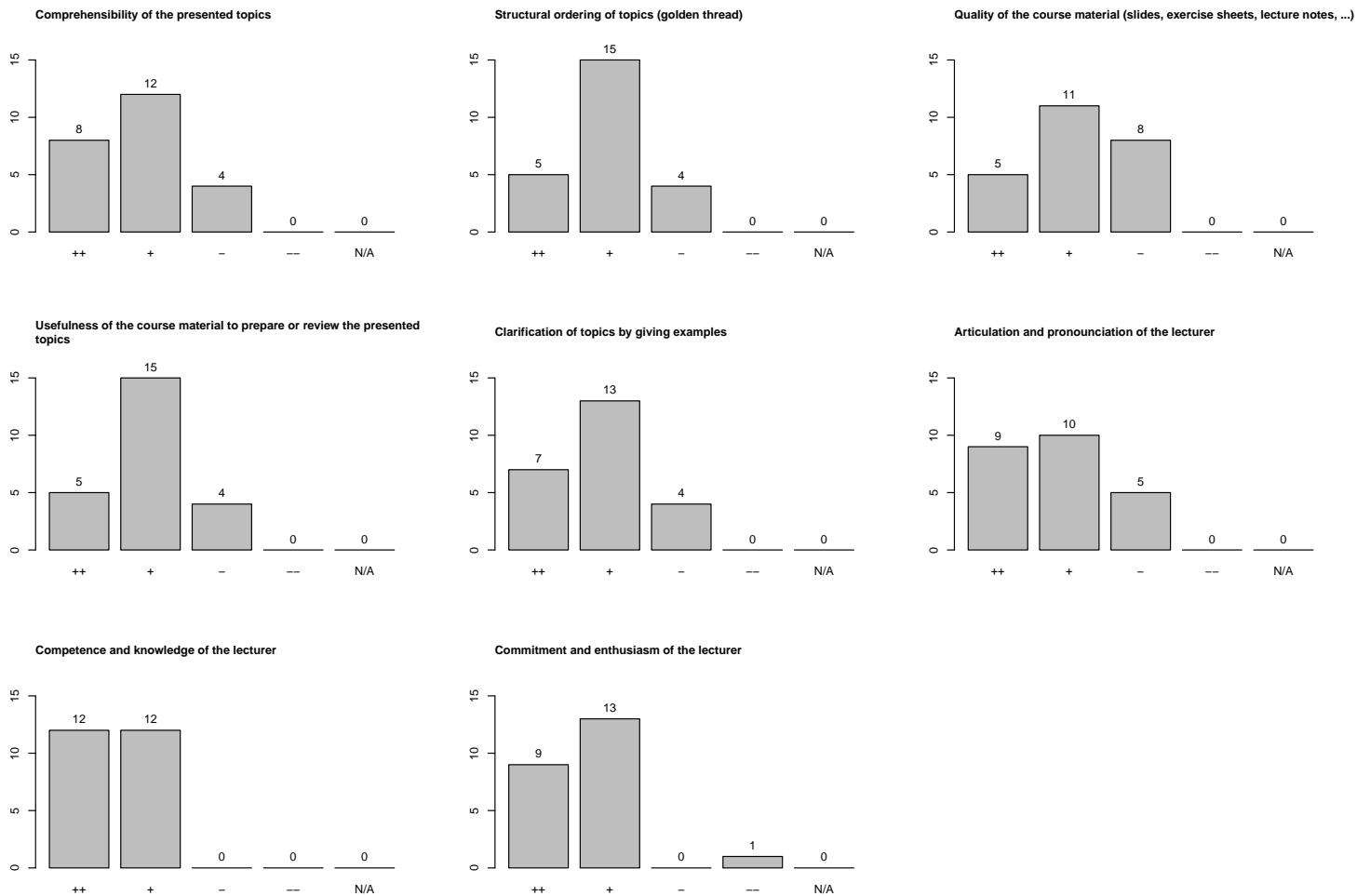
PROF. DR. ARMIN B. CREMERS, DANIEL SPEICHER, JAN PATRICK NONNEN

Average grade: 2.2

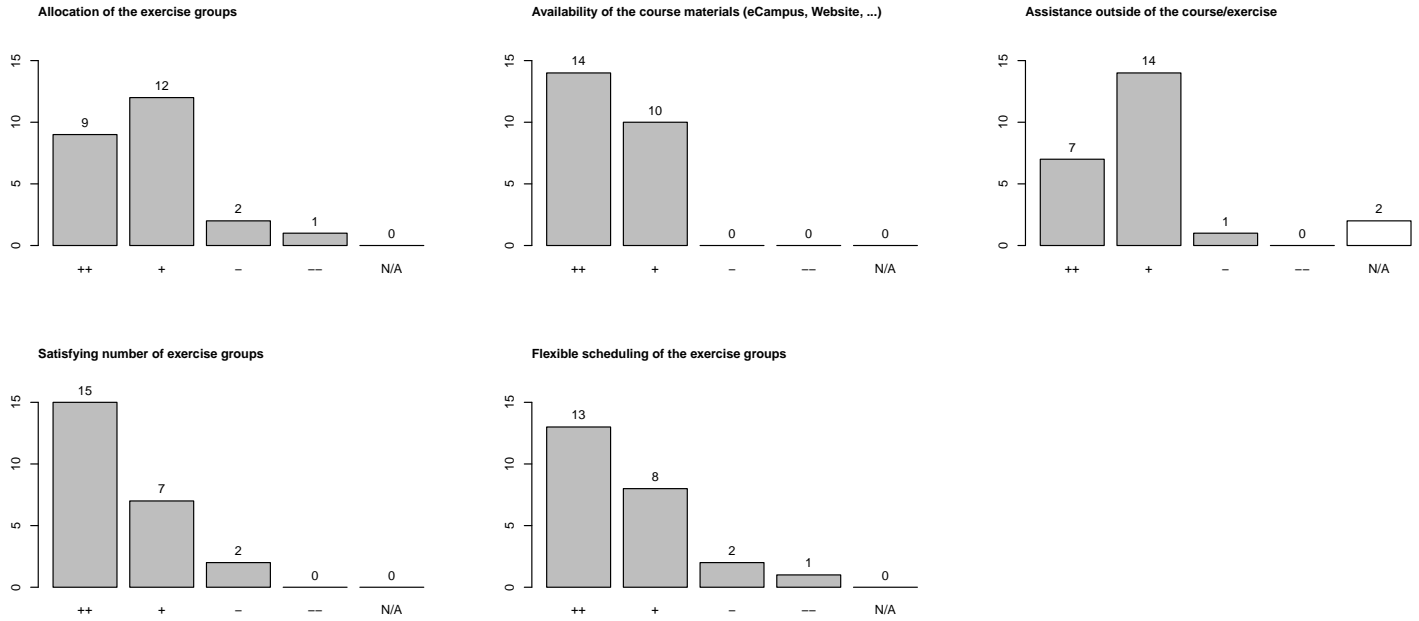
Participants (evaluated survey sheets): 24

- Bachelor: 0
- Master: 24
- Diploma: 0
- Lectureship: 0
- Minor subject: 0
- FFF: 0

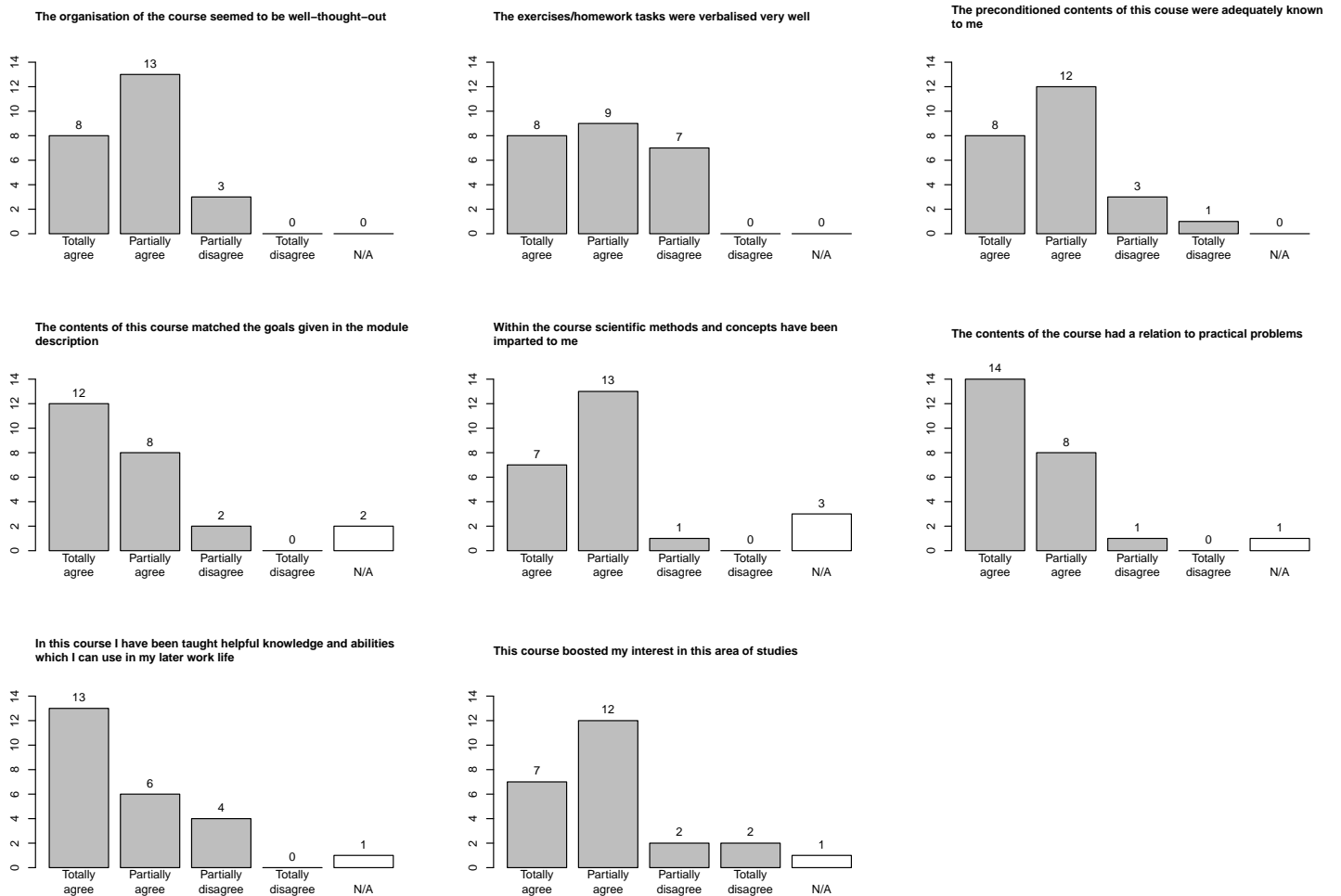
1 Please rate the quality of the lecturer's teaching.



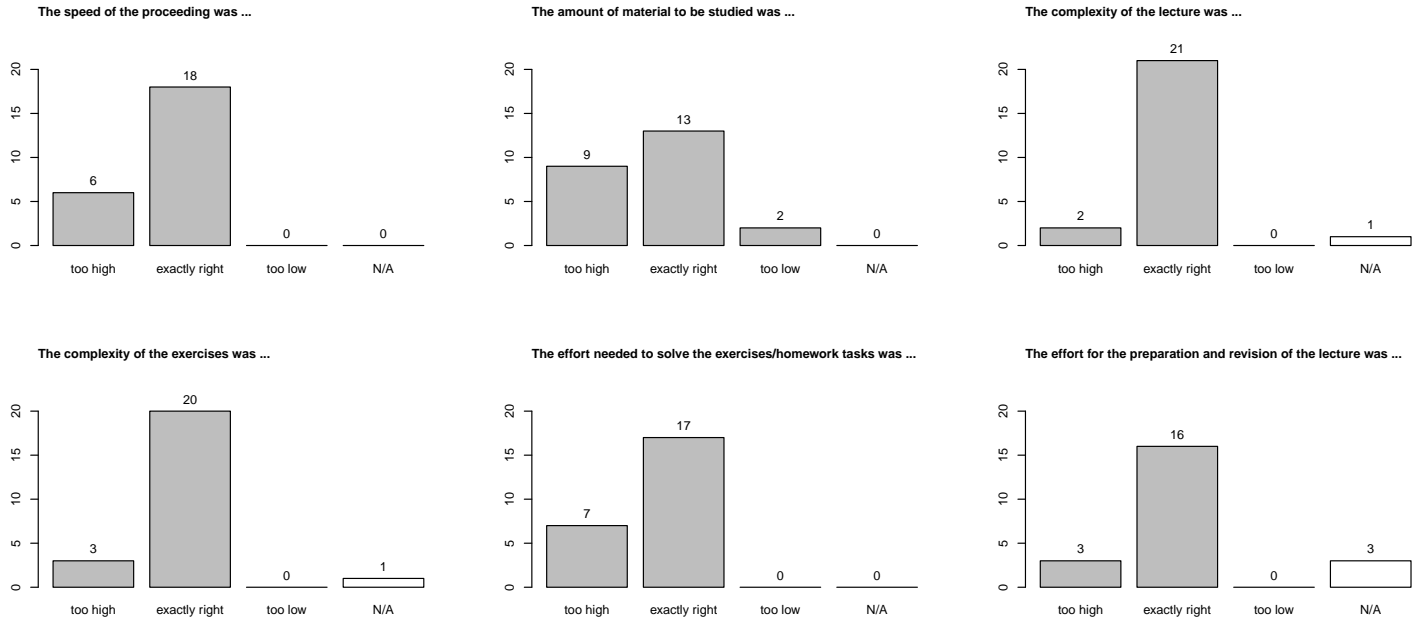
2 Please rate the organisation of the course.



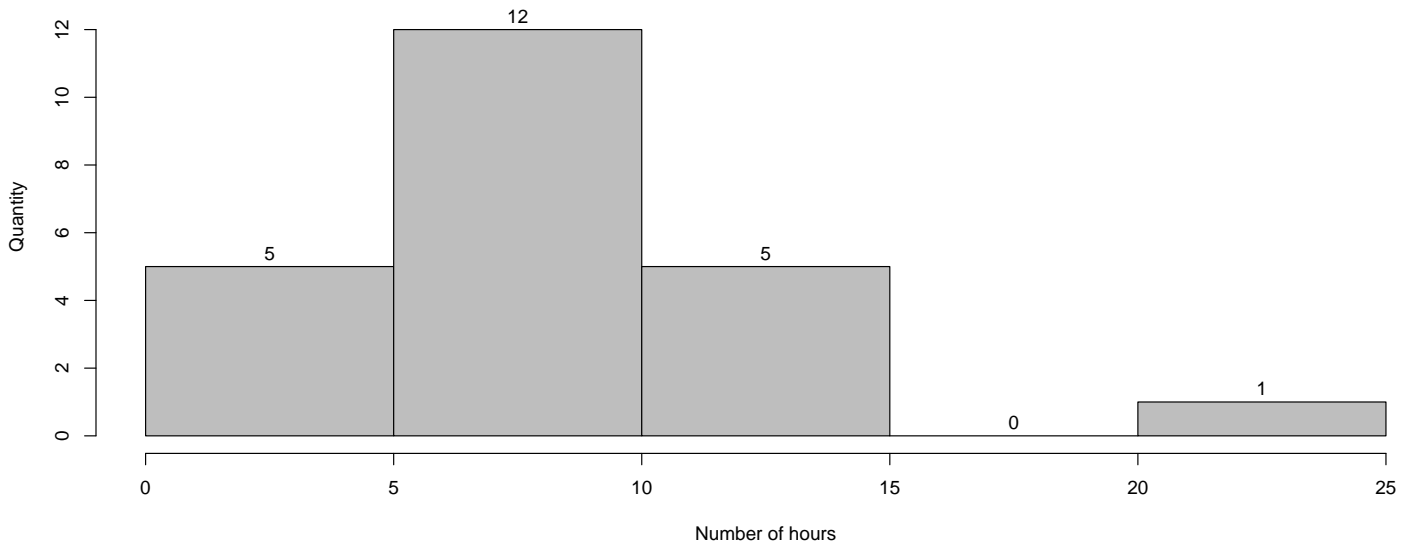
3 Please rate how the following statements fit your opinion.



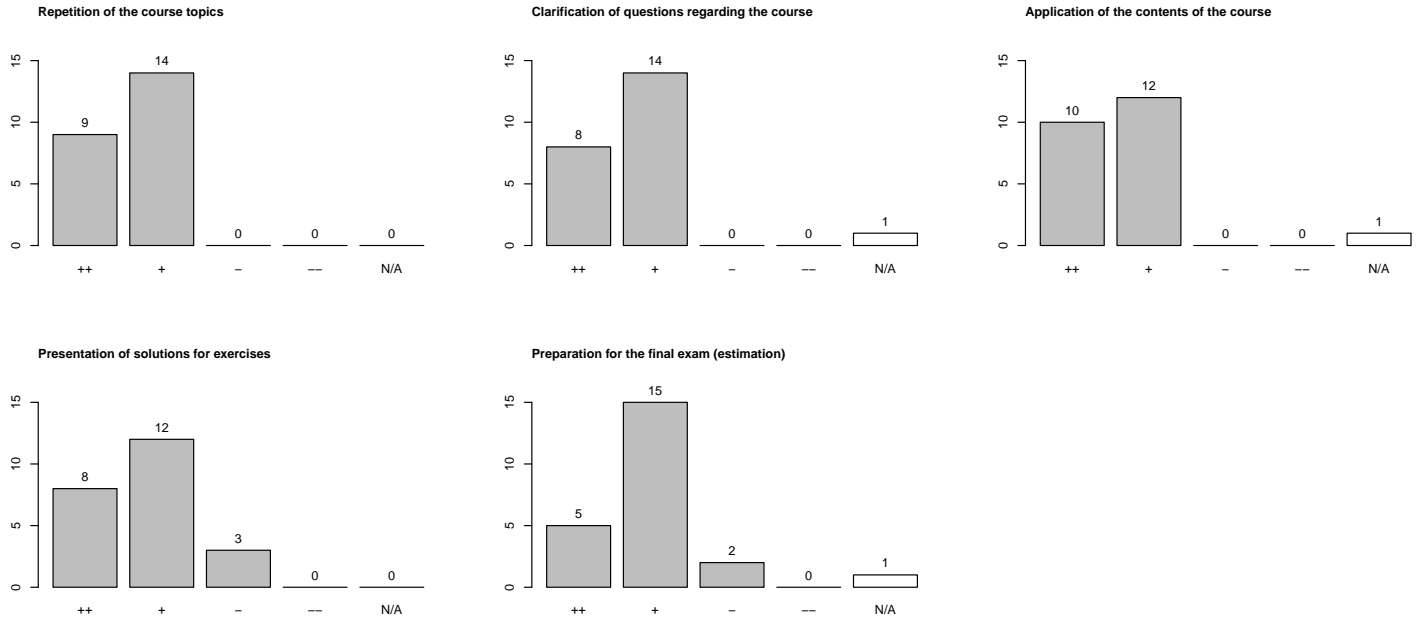
4 Please estimate the effort and complexity of this course.



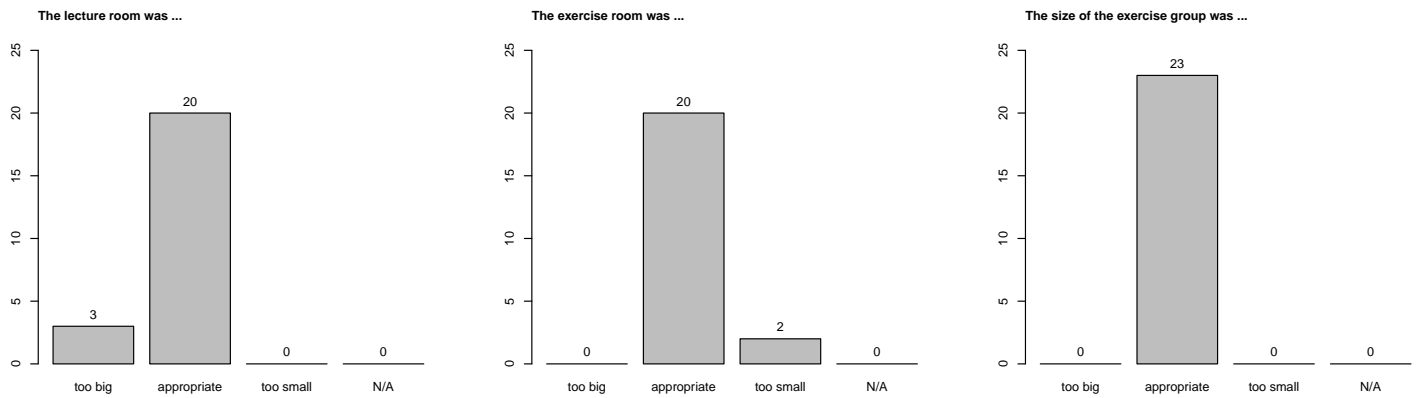
5 How many hours per week did you spend on this lecture (including the visit of the lecture and exercise groups) on average?



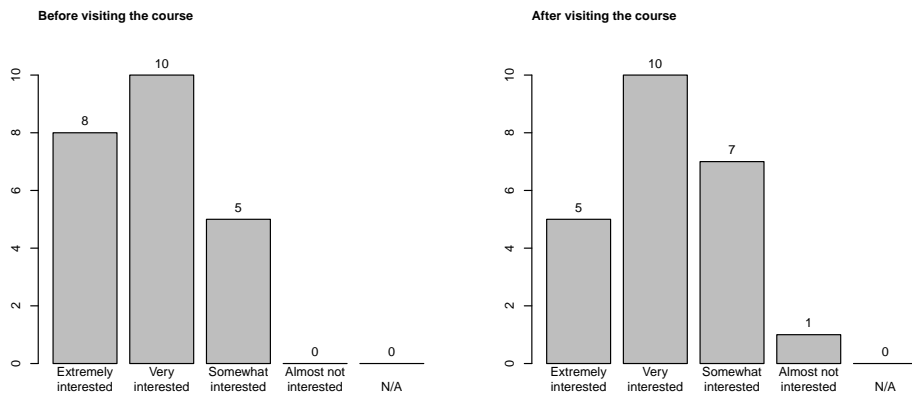
6 Please assess the value of the exercise groups to help understanding the presented topics.



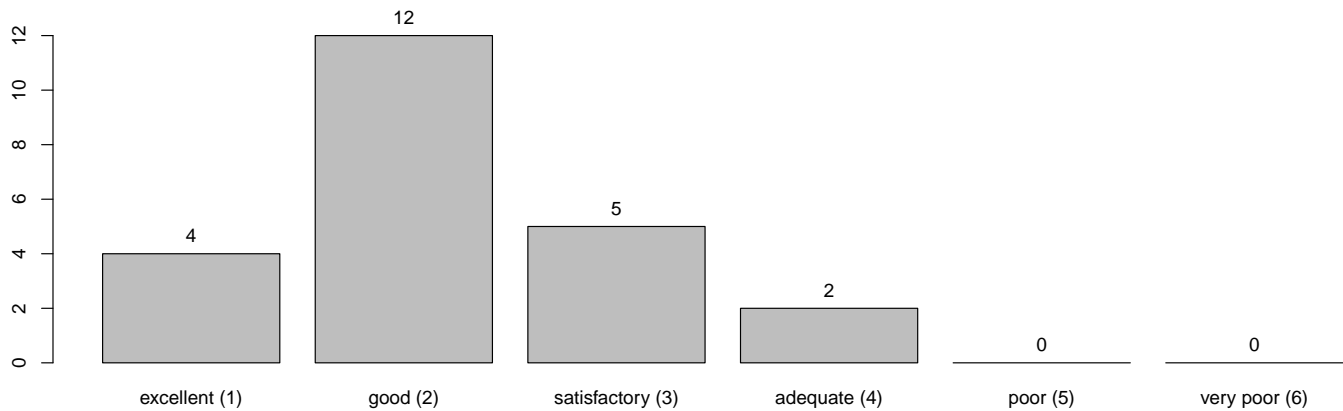
7 Please rank the size of the rooms and exercise groups.



8 Please compare your interest in the topics of the course before and after visiting the course.



9 Please give an overall rating of the course on a scale from excellent (1) to very poor (6).



10 Comments

Which things of the course did you like?	What could be improved?	You can leave remarks and feedback to our survey here.
	Contents of the course could emphasize more on real-world problems, rather than theoretical knowledge. Number of topics studies could be decreased to give deeper knowledge on more important ones.	
	if there were more practical examples with notes on slides this would be helpful to learn taught concepts. Some Slides look confusing since the concepts are almost without explanation	
Practical exercises where we were asked to write a code were very interesting		
Especially i liked lectures. And i learned more things during this course, not only about this field, also to fight with problems :)	In Exercise, some questions were not understandable somehow, may be change or ask questions in easy way :) And in the beginning we were three person in exercise group, but one of us dropped course and we were two, so it was really difficult in exercise, maybe it could solved by change or join some groups :)	Firstlay, thanks all of you :) And i wish good luck

Overall getting to know more about theories behind software construction, the anecdotes of Daniel Speicher. Housam as a tutor usually managed to explain things well and clarify what we didn't understand in the lectures	Often, after visiting the lectures, i still didn't know how to solve the exercises. The slides were only partly helpful, some additional material or examples would be good. Alternative solutions could be more encouraged. It's important to learn the concepts, but perhaps omitting certain constraints ("there are no 1:1-Relations") when it makes sense should be possible. Jan Nonnen seems to have very good knowledge, but he could improve rhetorically in the lectures	it's pretty good as it is. Questions 10 and 11 are the most important ones in my opinion, so perhaps there should be a bit more space for them
it was fun doing the exercises. All three lecturers did their job very good		
The practical use of the topics presented is clear and thus it is very motivating. The lecturer and the helpers are very competent, friendly and helpful. Thank you!	There should be a bigger connection of the lecture and the practical lecture by pointing out clearer and which way theory and practical work are related. It was hard to have just one day to prepare the theoretical part for the practical lecture	It was very interesting. I'm sorry that the Agile Programming Lab is in the semester holidays because it's impossible for me to attend it then.
lecture hall. graphical presentation of some example	well-explain the slides, clarify the slides, more details. provide some materials on the website	
The ranking. The small groups	not almost every week a different tutor (in the beginning of the course)	
The exercise competition: ranking the students (anonymously) was very motivating. Exercises were hands on and practical	there were too many slides per session. Most topics (like AOP) were discussed too briefly (leave some topics out and handle the others more in depth). Tutors for the exercises changed frequently. They were all good though	
Lecture about aspect oriented programming. Some Methods for software development	interesting topics were translated to boring Lectures. Don't show a fast overview about everything. show the general idea and a few concepts in more detail	
Exercises were the best part of the courses. The discussion method for exercises was really helpful in understanding the subject more	the lecture room is too big. sometimes it's hard to hear what lecturer is teaching. so it can be shifted to some other room.	
Student assignment competition. very practical topics. Small exercise groups. manageable assignment sheets	More concepts and methods that are really applicable to real-world software engineering. Less fancy and purely theoretical methods that have no relation to everyday software development. More flexible exercise group slots. less focus on Java/Eclipse-only technologies	
1. Aspect (AOSD) 2. MOF Lecture 3. Model Driven Approach	Every Lecture to be of same as it was. So, Nothing much to improve	1. Good Outline given on Topics 2. Flexible Timings
Aspect Oriented. Agile Development. Visualization	More Practical Problems in the exercise. Attachment of lecture with practical Problems	I believe that somewhere in exercise problems were focused mainly on JAVA. This could be improved. I was looking forward for Design Patterns. But didn't take place :(Rest i believe the course is well structured

New topics in this area		
-------------------------	--	--