PUBLISHED COURSE ANALYSIS

Publishing date: 2018-11-22

A course analysis has been carried out and published by the course convener.

The Karlstad University evaluation tool is owned by the Professional Development Unit and is managed by the systems group for educational administration, Student Centre.

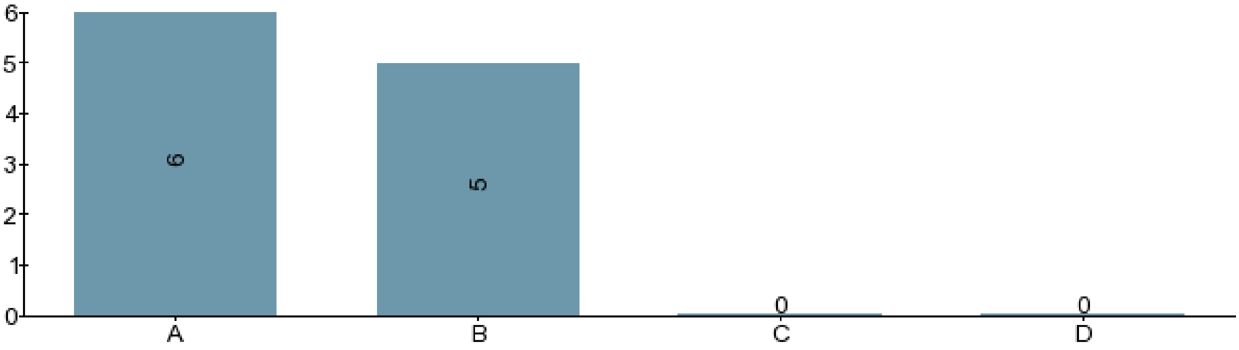
Software Engineering, 7.5 ETCS cr. (DVGC22)
Course convener: Martin Blom

Basic LADOK data	
Course Code:	DVGC22
Application Code:	31710
Semester:	HT-18
Start Week:	201835
End Week:	201844
Pace of Study:	50%
Form of Study:	Campus

Course Data	
Number of questionnaires answered:	11
Number of first registrations[1]:	43

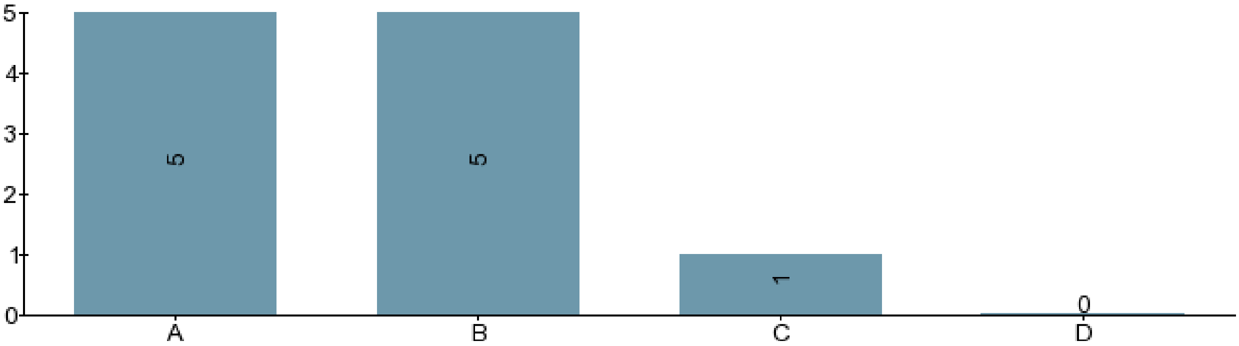
Changes suggested in the course analysis of the previous course date:
--

1. During the course I developed the knowledge, skills and other competencies described in the learning outcomes.



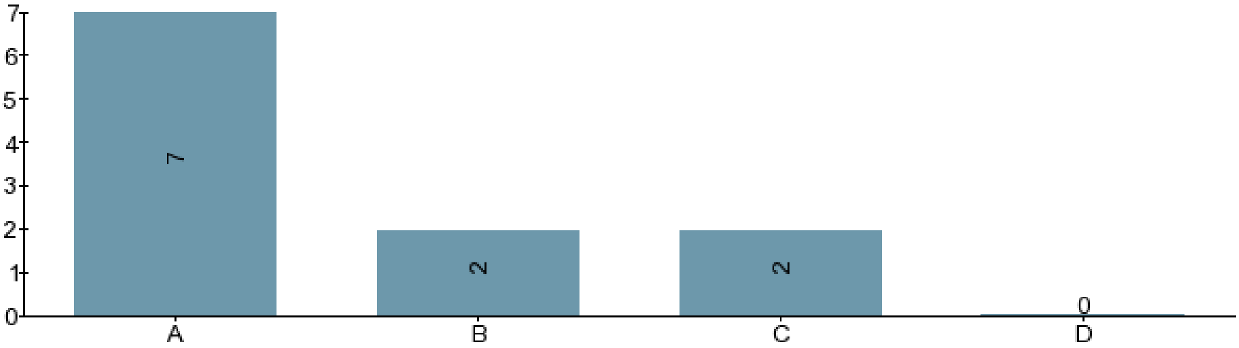
A) To a very great extent
B) To a great extent
C) To a certain extent
D) To a very little extent/Not at all

2. In the examinations, I had the opportunity to demonstrate if I have acquired the knowledge, skills and other competencies described in the learning outcomes.



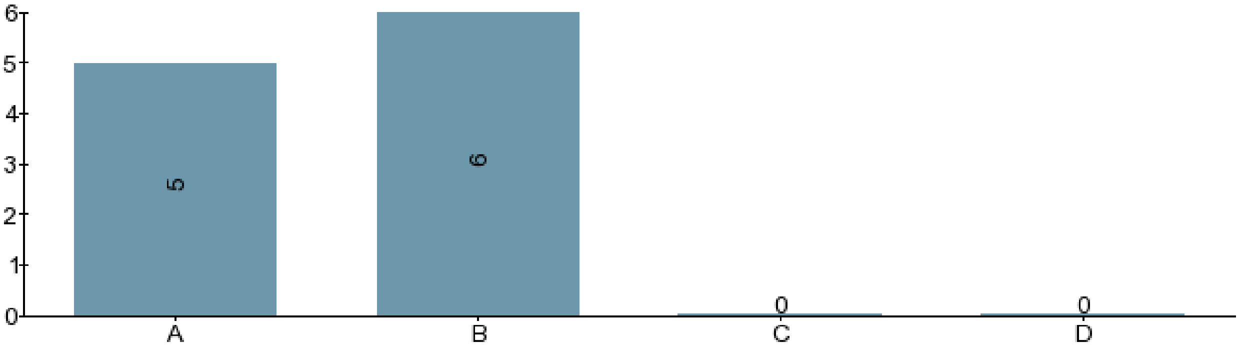
A) To a very great extent
B) To a great extent
C) To a certain extent
D) To a very little extent/Not at all

3. On average, I spent the following number of hours on coursework per week:



A) More than 40 hours (or more than 20 hrs at 50% study pace, more than 10 hrs at 25% study pace)
B) Between 30-39 hours (or between 15-19 at 50% study pace, between 8-10 at 25% study pace)
C) Between 20-29 hours (or between 10-14 at 50% study pace, between 5-7 at 25% study pace)
D) Less than 20 hours (or less than 10 at 50% study pace, less than 5 at 25% study pace)

4. During the course, I have found that teachers and other staff have been:



A) Professional and very accommodating
B) Professional and accommodating
C) Professional
D) Deficient

Analysis based on course evaluation, including comments fields. If information has been collected in other ways, it

should also be analysed here. Any effect of joint courses should be commented on.

Based on the first questions, the students generally seem to have been happy with the course, a sentiment shared by me (the teacher) as well. The course went reasonably well. There are a number of relevant comments from the students in the free-text answers that are worth both highlighting and acting on:

* Two schedules. The scheduling system at Kan, using TImeEdit with limited editing capabilities makes it difficult to specify all minor parts of the course and hence a separate schedule was produced. This is not optimal, but given the cumbersomeness of TimeEdit I think it would be difficult to get rid of but I will try.

* Bad planning for sprint planning and demo. The sprint planning meetings take quite some time, and with seven groups in the course the wait could be hours long. It would of course have been much better had the customer/product owner planned this and assigned time slots to each group. As for the demo, there are both pros and cons about the fact that all groups saw all other groups' presentations. The good thing is that the audience becomes bigger and that groups can be inspired by other groups. The downside is that the demos take a long time, the room is quite crowded and that groups might be less creative and unique because the all get too inspired by each other. I am not sure which way to go here and would need to think and discuss this more before making a decision.

* Too little focus on code. Even though the code itself isn't the most important part of the course, it is still the output from the groups and should as such be more important in the course nevertheless. There are actually more reasons than the ones specified by the students (wanting to learn more about using principles and patterns) related to grading the project work. Project work was in this course instance graded by individual weekly reports plus praise from other group members. This is not enough to fairly be able to judge a students contribution and value within the group. Adding code-related measures to the group work grade would be good and these measures can also be used as feedback to students during the course.

* Too short exam time. Judging by the exams, there seems to have been enough time to write the exam, despite the comment to the contrary. A short time also forces students to focus their effort and not writing long essays of free prose, but rather to concentrate and distill the knowledge they have gained in the course. The exam time will not change.

Suggestions for changes to the next course date.

* Put all scheduling in TimeEdit
* Plan sprint meeting per group
* Think about having separate demos for groups.
* Introduce code measures both at part of feedback to students, but also as part of examination of group work

---

1. **Number of first registrations for a course:** First registration = the first time a student registers for a specific course.