Summary results of the Technische Universiteit Eindhoven (TU/e)

Course

The course Programming (2IP90) is the first programming course of the bachelor degree Computer Science. It introduces imperative and basic object-oriented programming with Java. It comprises 14 lectures and 14 practical sessions. The study load is 5 ECTS.

The topics covered are: assignments, control structures, methods, objects, inheritance including abstract classes and interfaces, exceptions, GUI programming.

In the practical sessions, students make ungraded exercises concerning the new concepts in the course timeline. Furthermore, students make six graded homework assignments, in pairs. For students who have experience in programming, there is an optional challenge program, where they make more advanced assignments, requiring more effort and (algorithmic) insight. The assignment results contribute 40% to the grade. The rest of the grade is determined by an individual written exam, with the extra requirement that at least half of the points have to be achieved to pass the course.

Baseline measurement

The baseline concerned the instantiation of the course in the academic year 2021/2022. It ran from September 2021 to October 2021. About 400 students registered for the course.

Validation measurement

The validation concerned the course as it has been given in the academic year 2022/23, in the period September 2022 to October 2022. Again, about 400 students registered for the course. The course has been adjusted compared to the baseline. Some procedural guidance has been applied in the exercises of the practical sessions and it is explained what testing is and how to apply it. Students learn this in the practical sessions and apply it in the assignments, where they are asked to write test cases and test their programs. An exercise has been designed that runs through the first five weeks of the course and includes the newly taught concepts in the course, supported by the appropriate testing techniques.

Rubric

A rubric is used for grading and feedback. The rubric has categories for assignment specific aspects, such as the functionality of the solution and also general aspects of quality, such as adherence to coding standards. The categories of the rubric may have different numbers of points; these are normalized to a scale of 1 to 4 for project measurements. The rubrics are used during the baseline as well as the validation phase. Three assignments of the validation phase are identical to assignments of the baseline case. The other three assignments are different, in order to include topics of the intervention, such as testing.

Questionnaire

The standard QPED questionnaire is used. Questions are about previous programming experience, self-efficacy, perception of the fundamental elements that define programming and software quality, programming habits (like the use of meaningful variable names, naming conventions and coding styles) and the use of tools (like style checkers and testing tools). The Questionnaire is used during the baseline as well as the validation phase.

Main results

Rubric baseline

In the baseline measurement in the period September-October 2021, 73 students were involved. Students usually score well on readability, DRY principle, and output correctness (mostly level 4). Students score lower on modularity (about half scores 1). This topic is not so prominently taught yet at this moment in the course and hence shows a much greater variation in the scorings. Program robustness gives a mixed view with very good scores in assignment 1 (where this topic was easier) and lower scores (spread out over the levels) for assignment 3.

Rubric Validation

In the validation measurement in the period September-October 2022, 69 students were involved. We see again good scores on readability, DRY principle, and output correctness and lower (but improved, see below) scores on program robustness and modularity.

Rubric comparison

The results obtained from 73 students in the baseline phase were compared with those of 69 students in the validation phase. The comparison of results was carried out through univariate analysis techniques that made it possible to detect trends – in terms of differences and similarities.

We see positive trends in all categories, although often not that much, since the scores were already high. Output correctness and modularity show the highest increase in score. The first can be understood by the positive influence of the testing activities on the learning process. Modularity, however, was not part of the intervention as such, but it should be noted that modularity in this assignment is mainly about the proper handling of methods and the attention to (unit) testing may have contributed to the understanding of the concepts of methods and parameters.

Questionnaire

In most of the items the students in the validation phase consider themselves to be more skilled than those in the baseline. The use of graphical notations is low in both phases, which is not surprising, since these are not taught and hardly used in this first course of the programme. The results suggest the positive impact of the learning tools implemented in the project on the skills acquired by the students.

Conclusion

This study shows a positive effect of the use of specifications on the quality of the functional code and completeness of the tests.