Summary results of the Phillipsuniversität Marburg (UMR)

Course

The Phillips-Universität Marburg (UMR) is a classical full university. For the pilot study in the QPED project we have chosen the course Object-Oriented Programming (OOP, 9 ECTS), which is part of several study programs. It is mandatory in the first year of the Bachelor programs Computer Science, Business Informatics and Data Science. The majority of these students start their studies in the winter term, in which OOP is taught. A small number of Computer Science students start in summer and then have OOP in their second term. Additionally, this course can be chosen as an elective in further study programs, including Mathematics, Business Mathematics and Physics.

The topics covered are: Syntax of Java, algorithms, recursion, data types, memory layout, inheritance including abstract classes and interfaces, packages and libraries, introduction to software engineering including debugging, testing and step-wise refinement, exceptions, generic and enumeration types, lambdas, stream processing and I/O.

Students have weekly supervised exercise classes in which they solve assignments in self-study. Additionally, they have to submit solutions to assignments each week, which are checked and assessed by tutors.

The course has been given by the same teacher as in the pilot study since winter 2018, with an exception in winter 2020. Before 2018 and in 2020 the course has been given by a second teacher. The course runs from October to February and has between approx. 170 and 200 students taking the exam each year.

Baseline measurement

The baseline measurement with the diagnostic test was performed in the run of 2020/21 with about 180 students. As the other QPED measurement tools – the questionnaire and the rubric – were not ready by then, the baseline measurement with the remaining two tools was performed in the winter term 2021/22 with about 200 students taking the exam.

Validation measurement

The validation measurement was made in the winter term 2022/23. About 170 students took the exam in this run. After the first baseline measurement, already small adjustments have been made to the course (winter 21/22) in terms of initially introducing the TILE approach for some exercises. Thereafter (winter 22/23) the course has been adjusted further compared to the baseline measurement: The TILE approach has been consistently implemented in the lecture as well as the weekly assignments. Furthermore, for at least one assignment per week students could use the MASS system to get automatically generated feedback on their programming solutions. Some of the assignments were also rewritten to stipulate a step-wise solution approach, following a simplified form of the procedural guidance.

Rubric

A rubric is used for assessment and feedback. The rubric distinguishes between an assignment-specific part and a general part. For the evaluation at UMR we only used the general part of the rubric, which consists of the concepts: modularity, datatypes, readability, dry principle, flow, API documentation, correctness, robustness, test traceability and test completeness. A scale of 1 to 4 is used. The rubric is used during the baseline as well as the validation phase. The rubric was, in both runs, filled in by the tutors (student assistants) of the course.

Questionnaire

The standard QPED questionnaire is used at the end of the course. 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. The students were asked to fill in the questionnaire after the first exam took place but before the repeat exam.

Diagnostic Test

The QPED diagnostic test has been used, which is a specific assignment in the final, written exam of the course. This assignment consists of two parts, first students should identify quality flaws within a code snippet and, second, they should write unit-tests to check the implementation given in a code snippet. No further instructions are given w.r.t. criteria for the tests. The assignment was graded by experienced teaching assistants with a rigid grading scheme. Over the years, only the code snippets were exchanged. The diagnostic test was used during the baseline as well as the validation phase. Additionally, it was used once before the actual baseline phase.

Main results

Rubric

The rubric was used by the tutors to assess a subset of the assignments. Four of the assignments used the rubric in the baseline as well as the validation run, with between 43 and 57 student solutions (students work on the assignments in teams of 3). For the fourth assignment, which was in the last week of the lecture, there were only 5 submissions. Since the cohort in the validation run was smaller, there are only between 25 and 36 solutions assess using the rubric. For the last assignment, again, there is only a very small number of four submissions.

In comparison, the rubric-based ratings for the different concepts are similar for all assignments in both the baseline and validation run. Consistently, the concept “correctness” was rated higher in the validation. The rating of readability also improved in the validation run. However, the test-related concepts relevant for the QPED project, do not show conclusive results. The concept “test traceability” usually has a positive trend, while “test completeness” usually has a negative trend. Since the assessment is done by different tutors each year, who have different levels of experience, we expect that this is partly due to differences in interpreting the rubric. Since these exercises have mainly formative character, variations in the tutors’ assessments are tolerable for the course, but make the results less useful for a comparative study.

Questionnaire

Since filling-in the questionnaire was voluntary, only a small number of 46 students (baseline) and 11 students (validation) filled it in. Due to the small number of participants, we cannot be certain whether the results are really representative. Nevertheless, they indicate a positive trend: In the validation phase, students consider themselves to be more skilled than those in the baseline. While there is some variation in the exact classification (i.e., whether they rate their understanding as “good” or “very good”), when combining the two positive and the two negative ratings, there is a clear positive trend. These results suggest the positive impact of the learning tools implemented in the project on the skills acquired by the students.

Diagnostic Test

The diagnostic test was completed over three years by 180, 200 and 170 students, respectively. Here, we found a clear positive trend: With every year, the score in the diagnostic task was more than 10% higher. This effect was statistically significant as well with a very low p-value at $p<0.001$.

Conclusion

The objective diagnostic test shows a clear and statistically significant positive trend. The other evaluations do not show such clear results with statistical significance. We expect that, in particular, the use of the rubric suffered from a large variance among the testers who fill in the rubric. Likewise, the small number of participants in the second questionnaire will distort the results. Nevertheless, also the rubric and the questionnaire indicate a positive development, which further underlines the already strong positive results of the diagnostic test.