SUCCESS AND FAILURE OF 1000 FIRST SEMESTER CS STUDENTS

Markus Pizka and Manfred Broy
Institut für Informatik
Technische Universität München
Germany – 85748 Garching
{broy, pizka}@in.tum.de

ABSTRACT

In the fall of 2001 we faced the challenge to teach nearly 1000 first semester computer science students. Apart from the enormous class size our goal was to deliver high quality CS education. To achieve this, we introduced a new process supported by a technical infrastructure allowing continuous monitoring of activities in the lecture and the complementary lab. The data collected by this monitoring system for 1000 students of a single semester represents an unique in-depth view on the reasons for successes and failures of freshmen CS students. The analysis of this material delivers valuable insights for future improvements of mass as well as non-mass CS education.

We present the organizational framework that was needed to teach 1000 students, sketch technical aspects of the monitoring system and relate the results of the final exam with the monitoring data collected during the term.

KEYWORDS
Mass CS Education, Empirical Study

1 GIANT CS CLASSES

The ".com" caused enthusiasm for IT related jobs in 2000, and 2001 entailed large numbers of computer science (CS) students. Because of the strong demand for skilled IT personnel in industry this increase was most welcome. In Germany, this market situation, combined with the liberal and public German educational system, lead to extremely large CS classes. Almost every applicant for CS who fulfilled only general requirements was accepted regardless of the number of students already enrolled.

In the fall of 2001 we faced the challenge to teach 953 freshmen CS students. This situation caused numerous problems, such as how to interact with nearly 1000 students. However, our goal was to deliver high quality CS education; i.e. a high percentage of students passing the final exam while teaching demanding contents. At the same time, we viewed the immense number of students as an unparalleled chance to perform a detailed study of the reasons for successes and failures of CS students.

The theoretical introductory CS lecture was guided by a tutorial (or lab) conducted in small groups. This setting is somewhat comparable with a CS1 [1] course as the tutorial consists of structured, supervised exercises and a programming lab. A web-based database system was used to continuously monitor participation and the deliverance of homework of each individual student. While some of the analyses of these data confirm common expectations – e.g. doing homework increases the chance for success – we also gained surprising and important new insights.

Outline

Section 2 briefly introduces our first semester CS major curriculum. Section 3 presents the organization of the mass tutorial and technical aspects of the monitoring system. After these prerequisites, we are able to discuss the results of the final exam and factors for success and failure in section 4.

2 CS MAJOR CURRICULUM

At the Technische Universität München, CS major-oriented freshmen are required to take the following courses:

- Introduction to CS
- Technical Foundations of CS
- Advanced Mathematics

In addition to these core CS courses each student has to take further courses in his/her minor field of study.

2.1 Introduction to CS (ICS)

In the fall of 2001 we taught ICS [2]. ICS is a 15 week long and 4 hour per week lecture with an exam at the end of the term. ICS covers the following topics:

- Foundations of algorithms (text rewriting, etc.)
- Abstract computational structures, abstract types
- Prepositional and first order logic
- Syntactic structures (reg. expressions & BNF)
- Functional and state-based programming
- Mathematical semantics and correctness

2.2 ICS Tutorial

The ICS lecture is guided by a 13 week and 3 hour per week tutorial. The goal of the tutorial is to deepen the understanding of the contents of the lecture using structured and supervised exercises, and to teach and practice programming skills. Hence, the tutorial covers some elements of a CS1 course. We started with the functional language Gofer [3] and had moved on to Pascal by the end of the term. Besides functional and recursive programming the tutorial also emphasized proving properties of algorithms by computational and structural induction.

To be effective, the tutorial was conducted in 49 separate groups of at most 30 students. A total of 39 Ph.D.'s, graduate, and senior students were employed as tutors for the tutorial groups.

Every week, we developed 4-6 structured exercises for the tutorial. These exercises were distributed to the tutors for supervision in their tutorial group. Each week, one or two of these exercises were marked as homework and were to be accomplished by each student independently and delivered to the tutor within one week. Every tutor had the duty to mark the homework received from their students and to discuss potential difficulties with the group.

2.3 80/60 Requirements and Final Exam

The written exam at the end of the semester consisted of 5 exercises testing propositional logic (6 pts), text rewriting (10 pts), BNF (9 pts), recursive functional programming (8 pts), and induction (7 pts). The maximum number of points achievable was 40 points and at least 40% (16.5 points) were needed for passing the test.

The problems posed to the students in the final exam were variations of exercises used in the tutorial. Thus, students who participated in the tutorial and delivered their homework on a regular basis should not have felt serious difficulties passing the exam.

Previous semesters have shown that few students participate in the tutorials and do their homework regularly without any additional stimulation. Therefore, we told them, that they would not be admitted to the final exam if they did not attend at least 80% (10 out of 13 weeks) of the tutorial or failed to deliver at least 60% (8 out of 13 weeks) of their homework; and not being admitted would be equal to failing the exam. The official examination regulations do not include such requirements, but by the time the first freshmen found this out, the semester was already coming to an end. We succeeded in achieving a high level of participation and, in fact, admitted every student who was willing to take the exam.

3 ORGANIZATION OF THE TUTORIAL

The 49 tutorial groups were scheduled into 5 different timeslots A to E from noon on Tuesday to late Wednesday afternoon. Students were able to specify preferences for up to 3 timeslots when registering for the tutorial. Furthermore, they were able to request joint assignment with up to 2 other students. Based on these preferences the 953 students were spilt into 49 groups. The tutors then picked groups according to timeslot and geographic (distance from office to the tutorial room) preferences. This means, neither students nor tutors were able to choose a certain tutor or group of students. The numbers presented below should therefore not be biased by predetermined combinations of tutors and students.

The tutors were informed about the progress of the lecture and were introduced to the exercises by the ICS board (ICS lecturer and tutorial coordinators) in weekly meetings. Similar to students, not all tutors attend such meetings regularly without external stimulation. Unfortunately, there is no way to force tutors to attend these meetings or into the preparation of their teaching in general.

3.1 Technical Infrastructure

In contrast to a standard lecture with perhaps 50 students, communicating with and responding to the needs of 953 students in the course of the lecture was extremely difficult and required breaking new grounds. Our approach to collecting feedback from the students was to continuously track participation, homework and further activities, such as dropouts and movements of students between groups. Figure 1 illustrates the infrastructure used for this purpose.

We built a web-based database system using MS-Access and CGI/Perl to collect and store up-to-date information about the situations within the tutorial groups. The tutors were requested to provide weekly reports on attendance and delivery of homework for each individual student using a web form. The ICS board used this information to steer the lecture and the tutorial.

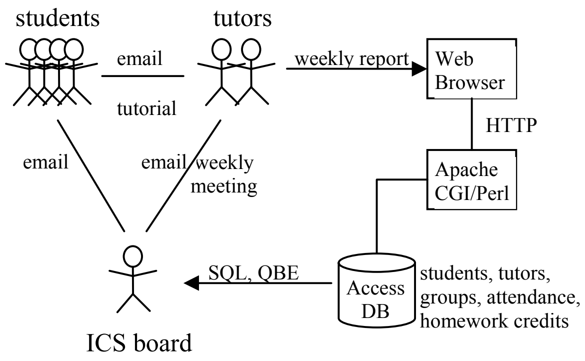

figure 1: controlling infrastructure

Apart form this database system, the weekly tutorials, and regular meetings, email was the primary instrument for communication between students, tutors and the ICS board. Within the 13-week period, the board received and replied to approx. 1500 emails from students, tutors and other board members. Each tutor handled about 200 emails from students per group supervised.

3.2 Unexpected Acceptance

The simple web-based controlling system proved to be of unexpectedly high value. First, it was rapidly accepted among tutors as they valued the possibility of printing out the evaluation form and showing it to their students. Consequentially, most tutors also willingly delivered their weekly reports.

In addition to this, students appreciated being informed about their performances with personalized notifications at mid-term and the end of the semester. They felt less anonymous and being more important to us. We received surprising responses like:

"this is an excellent service ... thank you very much for reminding me that I have to work harder ... I will…"

Most of all, this technically supported process noticeably increased discipline among all participants because of its high level of transparency especially to the members of the board. Since tutors needed to deliver their report by the end of the week, they were gently forced to mark the homework of their students regularly.

4 RESULTS

To be able to discuss the performances of our 1000 freshmen students, we start with the presentation of typical results achieved by previous non-mass ICS classes conducted without continuous monitoring. These previous experiences will afterwards be used as a baseline for comparison.

4.1 Previous Experiences

4.1.1 Final Exam

In Germany, failure and drop out rates of up to 40% at the freshman level are quite common. Amongst others, there are two explanations for this to keep in mind.

First, higher level education is provided free of charge and besides standard requirements, such as an advanced degree from high school, admission to universities is unrestricted. Therefore, a significant number of students starts studying CS without sufficient preparation and occasionally without adequate interest. Using the first semester for orientation entails high drop out rates.

Second, there are two ways to pursue professional training. First, universities and second, apprenticeships, i.e. part-time employment combined with off-the-job tuition over the period of 1-3 years. To a certain extent, it is accepted that some students fail at university and switch to apprenticeships.

4.1.2 Characteristics of Previous Tutorials

Tutorials that were conducted without an 80/60 requirement and without constant monitoring experienced a continuous decrease of attendance. The homework went from roughly 80% at the beginning down to 20% at the end of the term. In addition to this, students frequently switched groups. Some groups starved while a few other groups, conducted by motivated and skilled tutors, became congested with up to 50 students. In turn, the tutors of these large groups became overloaded and unable to mark homework and to respond to the individual needs of their students.

It can be supposed that this behaviour contributes to weak results in the final exam. Hence, a major intent of our administrative framework was to change this situation while studying the reasons for these effects.

4.2 Qualitative Experiences

The effort invested into the monitoring system quickly proved to be rewarding.

4.2.1 Reduced Rotation

After just 3 weeks there were no further rotations of students between groups. Interestingly, the main reasons for this were the access restrictions enforced by the web form for the weekly tutor report. Tutors were only able to access data of students belonging to one of their groups.

Therefore, to receive credits, students had to visit the group they were assigned to.

Switching groups was still possible, but it required valid reasons as well as approval by the ICS board. With this method the students delivered valuable feed-back on their tutorial groups to the board instead of silently escaping from unpopular tutors.

4.2.2 Integration of Tutors

The application of a web-based monitoring system improved the integration of the tutors and the ICS board into an effective team. On the one hand, our weekly statistics increased the interest of the tutors in the tutorial as a whole. As tutors were able to watch current trends in numbers they became motivated to achieve better results and sometimes even started fruitful competitions. On the other hand, tutors were quickly aware that the data entered into the weekly report not only exposed the behavior of the students, but also reflected their own capabilities. Poor attendance may indicate weak abilities to motivate students. This encouraged most tutors to become more dedicated to their teaching.

4.3 Quantitative - Attendance and Homework

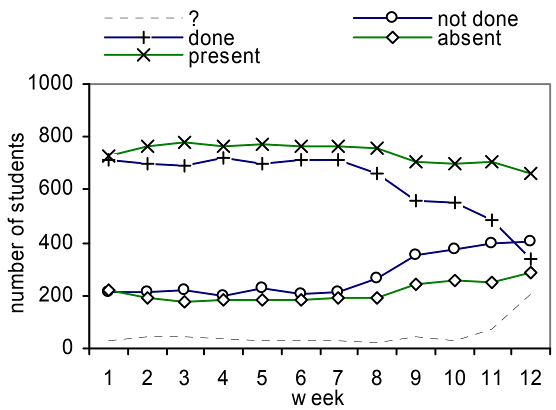

figure 2: attendance and homework

Figure 2 presents the numbers of attendees (absent/present), homework accomplishment (done/not done), and missing reports (?) during the semester. Week 13 was left out because of too many missing reports.

It shows that nearly 80% of the students attended the tutorial on a regular basis. Participation even increased during the first 3 weeks when the students realized that we were seriously monitoring the achievement of the 80/60 requirement. Homework was delivered in great numbers. But, as one can see, motivation for doing homework immediately dropped after week 8, when most students had achieved the required 60% homework credits.

We were also surprised by the cooperation of the 49 tutors. The number of missing reports remained low until week 11.

Besides these encouraging results, a surprisingly high number of about 200 students (~20%) did not attend the tutorial from its beginning. We analyzed that these 200 absences were produced by almost the same 200 students each week. Presumably these students signed up for CS but changed their minds meanwhile. Obviously, these 20% contribute significantly to the high drop out and failure rates at the end.

4.4 Results of the Final Exam

The ICS exam is mandatory for CS majors and some CS minors, such as mathematics. Students are automatically granted up to 2 retries in the following 2 semesters before being dismissed. Meanwhile they may either repeat the first semester or proceed regularly.

559 out of the 953 students registered for the tutorial decided to take the final exam.

	passed	**failed**
first semester	229	225
higher semesters	21	84
Total	250	309

table 1: final exam

Clearly the results of the final exam were far below our expectations and hopes. More than 50% failed the exam and only 250 students passed. Table 1 also exposes that a vast 80% of the students in higher semesters, retaking this exam for the 2nd or 3rd time, failed whereas 55% of the 454 freshmen, taking this exam for the 1st time, passed. It seems that several retries are hardly promising.

4.4.1 Language

For a considerable number of 40% of the students, the German language used in the lecture and the tutorial was not their first language. Table 2 compares native German speaking students with others. As one can see, only 26% of the non-German native students passed the exam whereas 57% of the native German-speaking students passed.

There are several different possibilities to interpret this observation. Obviously, understanding the language used in the course is important. In addition to this, these numbers also reflect differences between the educational systems of 21 different countries as well as their qualification for this specific CS curriculum. Further analysis, which we do not present here, indicates a correlation between the performance of the students and their geographic origination.

native tongue	result	num
Yes	passed	190
Yes	failed	142
No	passed	60
No	failed	167

table 2: first language

4.4.2 Gender

Female students performed significantly weaker than males (see table 3). Males reached an average of 42% of the maximum 40 points of the exam while females averaged only 30%.

Gender	avg. points	num
Male	42%	467
Female	30%	92

table 3: gender

4.4.3 Majors and Minors

To eliminate the influence of different semesters, we will only consider the 454 freshmen for all further statistics.

CS majors are able to choose between different minors. Table 4 compares the performances of CS majors with different minors. Physics and math minors clearly outperformed those with other minors such as business administration.

Minor	avg	num
Physics	70%	9
Math	54%	37
Psychology	44%	8
electrical engineering	41%	119
business administration	39%	144
medical science	36%	16

table 4: CS majors with different minors

A similar observation is obtained by comparing the different majors with CS minor (table 5). In this case, math majors leave CS and business majors even further behind.

Major	avg	num
Math	61%	25
CS	42%	388
economics	37%	32

table 5: comparison of majors

Interpreting these numbers is difficult. Are students with an interest in math better qualified for CS or is math a particularly suitable supplement to CS studies? Most likely both speculations are true to some extent.

4.4.4 Attendance and Homework

Figure 3 relates the performances of the students with their attendance in the tutorial and homework exercises. The diagram supports the common assumption that attending the tutorial and doing homework leads to better results.

Interestingly enough, the diagram also shows a sharp drop in the homework graph at 10 exercises. Students who accomplished homework in 10 out of the 13 weeks achieved an average of 45% points. This means the majority of these students passed, whereas students with only 9 weeks homework averaged only 34% points meaning most of them failed!

This 80/60 requirement (8 out of 13 home exercises were needed to qualify for the exam) helps explain this phenomenon. It could be assumed that students that turned in 10 or more exercises are the ones that accomplished their homework primarily by themselves, whereas students with only 9 exercises tried to fulfill the requirements by partially copying from other students.

4.4.5 Impact of the Tutor

group	avg	num
35	69.6%	13
34	60.2%	7
29	51.8%	12
41	49.9%	11
22	47.8%	11
20	46.1%	11
1	44.9%	9
30	44.0%	11
27	42.5%	7

group	avg	num
23	39.6%	10
40	38.8%	3
2	38.4%	7
9	36.6%	12
28	35.6%	11
3	32.0%	5
24	29.8%	6
25	21.2%	9

table 6: group performance

Among the most interesting results is the discrepancy between the performances of different tutorial groups. Table 6 shows the result of sorting all tutorial groups by their performance in the exam[1]. While the best group achieved an average of 69.6% and everyone passed, the weakest group scored only 21.2% and almost all failed. But all students attended the same lecture and all groups were given the same structured exercises and homework. The only obvious difference between the different groups is the person of tutor teaching the tutorial. This indicates that the tutor has a tremendous and underestimated impact on the success and failure of students.

[1] here, the table was condensed by displaying every 3rd group.

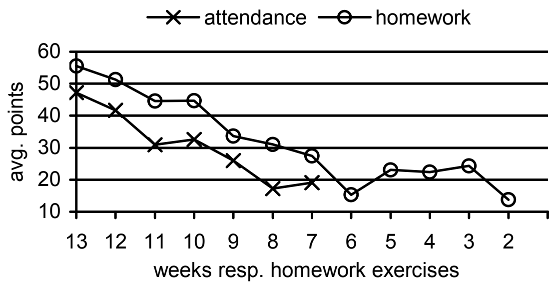

figure 3: attendance and homework

While most tutors only taught exactly one group, eight of the tutors had to teach two groups. To further investigate the impact of the tutor, we compared the performances of pairs of groups conducted by a single tutor.

Table 7 lists the id of the tutor teaching two groups and the performances of his chronologically (i.e. day of the week) 1st and 2nd group. With the exception of tutor 4, all 2nd groups performed significantly better than the 1st groups, sometimes with incredible differences, such as in case of tutor 6 – 32% versus 65%! One likely explanation for these enormous differences is the preparation of the tutor for the tutorial. The tutors themselves obtain a deeper understanding of the exercises with their 1st group thus being better prepared for the 2nd group.

tutor	avg 1st group	avg 2nd group
1	44.0%	47.1%
2	38.1%	38.7%
3	46.1%	54.3%
4	38.5%	35.6%
5	39.6%	60.9%
6	32.0%	65.0%
7	38.5%	51.8%
8	36.2%	48.6%

table 7: pairs of groups taught by the same tutor

5 CONCLUSION

Undoubtedly, the monitoring infrastructure introduced in this paper has proved its benefits as a tool for the continuous observation and improvement of CS education. The data collected represents unique experiences with mass CS education and delivers an in-depth view on successes and failures of CS students. By this, it opens new perspectives for future improvements of CS education.

It should be evident, that solely regarding failure rates is inadequate and differentiated analyses of the performances of both, students and teachers, are needed. This is especially important in situations with overall weak results, as described in this paper. Regarding only the results could easily lead to the misconception that the exam itself needs facilitation. This would decrease the quality of the curriculum instead of improving it. In contrast to this, our statistics enable more precise improvement steps.

Our numbers indicate that the tutors, who are in direct touch with the students, are crucial for the performances of students. Most of the tutors are at the same time junior researchers. Their careers usually depend on numbers of publications and research grants in one way or the other. Unfortunately, educational performances are not as important and are rarely considered. Therefore, we believe that transparent controlling systems as presented in this paper and incentives for educational achievements are basic prerequisites to increase the motivation for teaching tasks and thereby improve the quality of CS education. While such measures using similar systems are common in virtually any business environment, our numbers indicate the potential benefit of such systems for education.

Clearly, one has to be very careful with the interpretation of the raw numbers presented in this paper. A sound and detailed interpretation must be based on a thorough statistic analysis of these numbers. This analysis along with continued long-term observation will be our next steps.

6 REFERENCES

[1] ACM Curriculum Committee on Computer Science. Curriculum '78. *Communications of the ACM*, 22(3):147-166, March 1979.

[2] Broy, M.: *Informatik – Eine grundlegende Einführung*. vol 1, edition 2, Heidelberg, Germany: Springer Verlag 1998.

[3] Mark P. Jones: *Introduction to gofer 2.20*, ftp://nebula.cs.yale.edu/pub/haskell/gofer, September 1991.