FOSTERING WRITING HABITS IN COMPUTER SCIENCE: A ROAD TO MEANINGFUL LEARNING.

María Feldgen[1] and Osvaldo Clúa[2]

Abstract *¾ Writing activities have been reported as a uniquely valuable mode of learning. Writing forces one to think, and it is an effective way to introduce discussion on current topics into the classroom and to contribute to an effective teamwork. We created an active and collaborative learning environment in the first of two related courses. We required every student in a team to write technical reports as material for discussion for the in-class and out-of-class assignments. They learned to organise material and clarify concepts, to ask questions, to process the information as a group in the classroom and then met outside of the class to finish up the project tasks. In the second course we had students from the first course and from another traditional course. We asked them to search for material for the in-class assignments about one central concept. In this paper we describe how the writing assignments in the previous course developed working habits. We observed that students who took our first course voluntarily wrote informal reports linking and organising the information they searched for. In the discussions and questions they used the reports as experts do. They used the learned communication and interaction skills during teamwork, they did not require additional support for these soft skills. This was not the fact with the teams with students from the other course. Finally, we describe how they performed in the final exams and in the counselling meetings.*

Index Terms ¾ Cooperative Learning, Dissection, Teaming, Writing.

INTRODUCTION

Research has shown that the process of writing can encourage active learning, particularly when the goal of writing is to structure ideas, clarify meaning and generate logical arguments [1][2]. When designing problem-solving strategies, writing is a tool that can be used to clarify a problem, to monitor progress, to integrate ideas and to modify possible solutions. The design is a cyclic iteration procedure, goal directed and non-linear that utilises heuristic reasoning processes and strategies to gather and filter information about the problem, and to inform the revision of possible solutions [3]. The software design approach is top-down and research stated that teaching students the principles of top-down design is not sufficient to enable them to practice top-down design. [4][5].

To be able to apply this process approach, students need both to understand how the process works and to be able to do it. Writing helps students to do it. The writing activity focuses the students' attention on their current work. These activities could be as simple as documenting observations, or describing the manner in which they decided to perform a task. Whatever the content, it should require students to think through their cognitive processes and make plans on how to improve that process [6].

From this belief we introduced writing activities in two related courses in Software Engineering. The two subjects, File Organisation & Processing and Operating Systems are offspring of a former one, which was split when our institution decided to move from full year courses to one semester-courses. But the faculty of the two is almost the same, and the topics, views and evaluation methods are strongly related. In both courses we try to develop top-down design skills using deliberate practice on projects, supported by writing activities.

Writing about the material of the course, about decision making before and after every step in design and as discussion material during teamwork, promotes reflection about how they are doing the task. And active reflection on experiences during teamwork promotes the acquisition of more meaningful and persistent learning [6]. Furthermore, writing gives us the opportunity to provide students with feedback, as a result, it becomes a highly effective tool in helping them uncover and then wrestle with their misconceptions while the learning is taking place [7].

In this paper we present the experiences from our two courses, comparing the behaviour of two groups of students. One group comes from our first course with writing assignments and the other group from a more traditional course. We observed that the first group students with writing experience voluntarily wrote informal reports linking and organising gathered information. In the discussions and questions they used the reports as experts do.

WRITING ACTIVITIES

To provide a framework for practice not only of design and programming, but also of the software processes as they are practised in the software industry, we use active instructional

[1] Maria Feldgen, Facultad de Ingeniería, Universidad de Buenos Aires, mfeldgen@ieee.org.
[2] Osvaldo Clúa, Facultad de Ingeniería, Universidad de Buenos Aires, oclua@ieee.org.

methods in our large classes, with more than 150 students. We alternate lectures with in-class problem solving, troubleshooting, problem formulation or brainstorming exercises to prepare students for teamwork projects during our three hour-classes [8]. To involve students actively and 'to keep the class with us' during the lectures, we ask them to prepare hand-written reports. The reports addressed information related to:

- The essential points discussed in the last lecture
- Preparing for the next lecture
- Related to the current project.

We carefully select the topics students must cover in their reports in order to convince students to embrace the writing assignments from the start and perform them with commitment and enthusiasm.

First we address students' interest in writing based on their expectations on how engineers work. Although the primary training and interests of engineers lie in technical areas, in many software-engineering activities, the ultimate product of their work is a written document. Organisations know this, of course, and sometimes base hiring and promotion decisions on writing ability [10].

Second, if the requested work is interesting students will feel that they do not waste time on it while they are learning how to master a professional report and how to write as a professional. Students must know that the report is evaluated by the individual effort he/she does. Consequently mistakes and errors are part of the learning process of writing reports. And students will appreciate feedback with corrections and recommendations.

Third, fostering habits in writing will allow students to reorganise the material in their own way, with their preferred examples, graphs, diagrams, tables as summaries of the different concepts and their relationships.

We ask students to write individual reports about a topic turning the purpose, ideas and arguments they have generated into a text. Students know that we do not expect them to produce a finished text in one sitting, but it is the first approximation of what they want to say. The report is a prewriting and a preparation of discussion material for a project report. For this reason it must be presented as a professional report, well structured and referenced. We help students with outlines and different proposed layouts to address different learning styles.

Our report writing activity addresses three primary benefits. First, it allows students to deal with their personal interpretation regarding a specific topic or a set of topics [6]. Second, writing results in enhanced group discussions, promotes collaborative learning and group members can gain detailed access to each other's thoughts. Third, writing gives us a powerful assessment tool that is quite useful in gauging whether students really understand a given topic [9].

We arrange projects and writing activities using a combination of principles from learning-style theory, from human motor behaviour and from group dynamics [10][11]. It is our way to teach them soft skills.

OUR TWO COURSE PLAN

We make writing activities mandatory in the first course and in the second course they are recommended but not mandatory. According to the recommended practices when developing collaborative learning groups [12] [13], we divide the writing activities in three stages:

During the first stage, the writing activities are for building a sense of community among students in the classroom. The individual reports give students the opportunity to know how their classmates work, their preferred style of presentation of information and their behaviour in general.

In the second stage, a special project is developed with a lot of in-class work using reports and the TIDEE competencies [14]. This project is the building step for an effective team.

In the third stage teams are supposed to be ready for the software design real projects, applying the skills learned in the special project and writing technical documentation.

In the first course the activities are well structured and a follow-up scheme is used, asking students to hand in elaborated material weekly. None of the students has ever been given this kind of writing assignments before.

In the second course three projects are asked with the same objectives of knowing people, forming and building effective teams. In this course we recommend students to write informal reports about the topics that they must discuss during the in-class assignments.

In this course we conducted a pilot study, as there were students from the first course and from a more traditional course. The pilot study was initiated to allow the authors an opportunity to assess the potential effectiveness of the writing activities and to see if the habits acquired in the first course show up in the next semester. Also we wanted to see the reactions of the students who had no writing experience from the previous traditional course.

Class Outline

The mandatory/recommended reports are used during the in-class assignments using the following outline:

A homework report on some topic is asked individually. Students are able to choose one of a list of related topics (e.g. different sort methods) as long as the entire list is chosen.

During the next class we work on the reports in four steps. First we present a real case and ask students with the same topic to solve the case in small groups at their seats using their reports. They have to write a short report on how to solve the case. Second a lecture using the different topics is held, allowing enough time for active participation asking and replying questions. Third students are asked to form mixed groups, each member with a different topic. They are asked to compare their approaches and write the conclusions. Last overall conclusions are drawn and the next topic is introduced in order to repeat the outline in the next

class. The in-class reports and the individual reports are handed in.

When teams are formed, the reports are related to their projects and the groups remain stable.

We used this working plan during the 1999 and the 2000 courses.

FIRST COURSE

Pre 1999 Experience

Before we introduced the two-semester plan, we used to have courses with three software projects with no writing activities. As an average we had 150 students in each course. In the first project, a team project with an average of 35 teams, only 71% of the teams completed the work and 30% of them had to redo some parts in what we called a remedial assignment. The second project, an individual one, from an average of 106 students who began the work, only 80 finished it with a 40% with remedial work. The third project had an average of 18 groups and with some exceptions, finished it as asked (Figure 1).

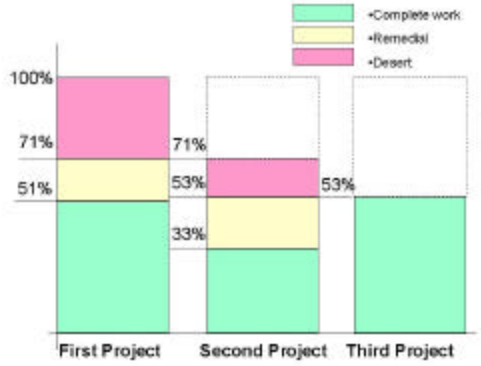

FIGURE 1

These results are usual in large classes with students from different previous courses in a commuter school. We can post only two conditions for students to join a group: common free time availability and place of residence. Most of our students are working as employees during the day and they come to class from 7 pm to 11 pm. Monday through Friday and during the day on Saturday. In addition, Buenos Aires is a big and congested city with high commuter time and most students live or work far away from the school building. In these classes, students do not know each other and they are accustomed to meeting in class only. In the last years we tried to change their habits and helped them to get to know their classmates and to create the need for out-of-class meetings.

1999 and 2000 Experiences

We spent three weeks in the first stage (six classes). We explained the syllabus and how we expected to go through it to the students. We explained and handed out Work Goals, Assessment Rules, Presentations and Writing Styles. Those are highly structured and well defined. But this was not the case with the way the work had to be done, in order to allow them to practice and hone a number of techniques and skills involved in problem solving. Also, we explained that teamwork is essential for their success in the course and in their career in general, and that class time would be used to build effective team skill [11]. For this reason, the class time schedule of three hours was divided into alternating activities of 45/60 minutes, lectures and in-class assignments as stated before.

In order to help them to get to know their classmates in class and to introduce them gradually to an active learning environment, we asked two reports on background known topics in the first stage. These topics were discussed in the first lectures. They knew that the reports provided them with an opportunity to observe "how their potential team members work", to gather personal information about individuals and to expose his/her preferred working and presentation styles.

When the class finished, students handed in their individual reports and the short in-class written reports. However a professor with large classes as we have cannot read all the reports. We selectively chose some of the reports to read based on student's behaviour during the in-class assignment. Then we made a general 'constructive criticising' and suggestions. Most of our students preferred that we did not mention the author of the mistakes. They came alone to ask how to do it correctly the next time, when they did not understand the general explanation.

At the beginning of the second stage, students were grouped in teams. The first steps to build a team are conclusive for the success of the projects and to actively involve every member in the teamwork. If we form an effective team, then we need to give them the right task. For this reason, in the first subject, we introduced a hardware dissection [15][16] in addition to our classical software projects. We expected students to draw the necessary analogies between the unfamiliar task of software design and the more familiar task of assembling a device. We found that the dissection activity is a key one in order to create the adequate background for meaningful software design process learning [17].

We divided the dissection project in different activities, identified by the TIDEE categories. TIDEE identifies the following categories in the design process: 1) Information gathering, 2) Problem definition, 3) Idea Generation, 4) Evaluation and Decision Making, 5) Implementation; and categories related to soft skills, 6) Communication, 7) Teamwork and 8) Process Improvement [14][18]. For each of the first four categories we asked for a written report, different for every member in a team.

In class they worked with their reports relating the steps within the process of design. The team had to make the work breakdown of the reports and the members of the team

decided how to distribute the different items within the group. The items of the first reports were of the same type, with the same difficulties in gathering information. The next reports had items with different levels of complexity, so the work breakdown could not rely only on a raw division of items into members. Members with a more difficult job needed the collaboration of the other members creating the need for out-of-class meetings to review the work breakdown. After the first category we wanted to have teamwork in all the individual and group assignments, and we got it.

The second and third projects were software projects. The second project had to be done individually. We made an individual assignment to ease independent learners integration for the last large project and to prevent students from having a free ride by going along with the rest of the team. Students had to apply the acquired soft skills.

The driving idea on these two projects is to allow for an exchange of information, asking the students to use the information gathered by other students. In the second project work is individually done using collectively elaborated ideas.

We followed the same steps as in the first project, a top down design with several feasible solutions. We also addressed the TIDEE competencies for the software projects and the reports and in-class assignments followed the same idea as in the first project, to reinforce the need for the steps to follow in a design.

In the 1999 course we had 180 students and in the 2000 course we had 150 students. The first stage was passed by a 90% of the students. Desertion was caused by personal reasons without relation to the courses. The first project was finished by everybody and the individually asked project was too. In the last project groups remained the same both years (Figure 2).

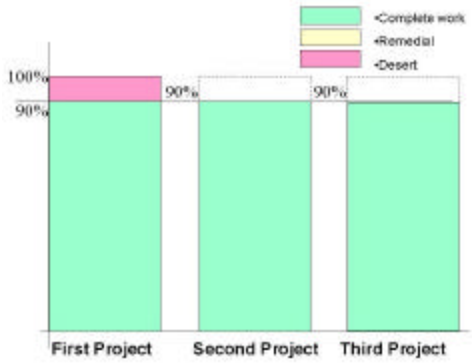

FIGURE 2

In 1999 the dissection project was asked to be done out of class. In 2000 we asked to do it in class to see how students used their plans and schedules. We found that our plan was a complete success, with no desertion after the first weeks and we did not need to ask for remedial work.

Students searched for information so filled with enthusiasm that assistants, Physics professors, advanced electronic students, librarians and we were literally annoyed by their persistence. They also "discovered" the library's magazines and journals section. They came to class with books, magazines and information gathered from different sources to support their arguments and convince their classmates about their references used in the reports. In the previous years we had only seen students and books together in our classroom before any examination. Time for in-class assignments was not enough, so they negotiated out of class time with the faculty in order to reach a general agreement.

SECOND COURSE

Operating Systems is the subject following File Organisation & Processing, so we could expect to have the same students in the following semester with the addition of the other File Organisation course students. However this is not the case. Depending on the different majors, several students wait for one or two semesters before taking the Operating Systems course.

Students have to develop three projects as mentioned above. The projects are chosen with the stages in mind:

- A general job control project to be done in UNIX,
- Some programming involving Operating Systems primitives and
- A set of Administration Scripts to accomplish some task.

We used this working plan, as a pilot study, during the 1999 and the 2000 courses.

Pre 1999 Experience

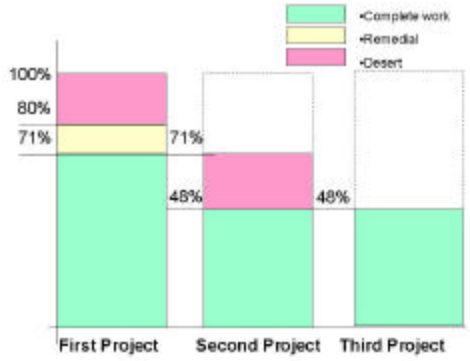

FIGURE 3

The pre 1999 experience was almost the same as in File Organisation: a high desertion when the project required more effort. From an average of 120 students, 80% passed the first project but with a 20% of remedial work asked. The second project was finished only by 60% of the students who passed the first one and there was no desertion in the third project. Though we did not ask for remedial work in

these two projects, grades were lowered according to their performance (Figure 3).

1999 and 2000 Experiences

In the 1999 and 2000 experiences we had the first mixed group of students: who worked with us the previous course and those who came from another one or from our pre 1999 courses. In order to see their interactions we scheduled more in-class time to do the work the projects needed.

In the first project, while our former students came with notes, the others came with a half-elaborated idea but not in written form. They were asked to draw a schema on how to do the job control assignment. We saw that students who wrote their ideas also wrote their references and they used them in the discussion with their team-mates. As a rule they took the leader place in the team and *asked* their team-mates to have some job done for the next class. During the next meeting they were asked to put pieces together and our former students exhibited more global knowledge of how each piece fitted in the whole.

During the second individual programming assignment we began to see some of the 'other' students asking questions with written material and working with it during a class that followed our four class outline.

For the last project the groups were not the same as in the first one, but with few desertions. Students grouped themselves following the guidelines we introduced in the previous course. Our former students remained in a sort of leading position because of their previous work breakdown experience, but the 'other' students discussed technical questions with notes and references provided by the 'old' students. There was no doubt that they learned some of the writing advantages from examples.

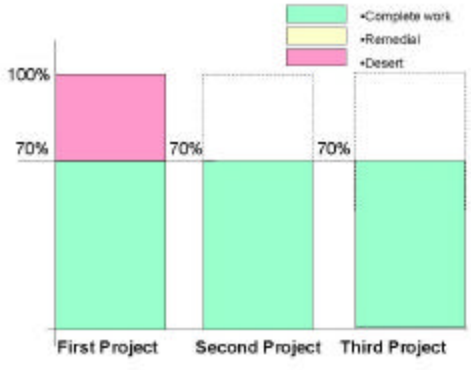

FIGURE 4

In the figures, desertion was concentrated in the first weeks. This can be an outcome of the tighter follow-up they felt during our in-class work (Figure 4).

CONCLUDING REMARKS

Engineering students seem to have a natural tendency to take in information and then assemble it for themselves into a useful structure. The planned activities purposely require them to role-play and interact in closely supervised groups. Writing is an explicit form of expressing this interaction, allowing a closer second look at it. This way of communication prevents the learner from assembling the information in his/her own convenient fashion, and provides a closely coached practice opportunity of demonstration and observation of each other's mistakes and suggested corrections [11]. As professors, we began to be actively involved in their learning acting as facilitators who provide personal feedback.

Students found the top down approach natural after the dissection project and the only way of arriving at a solution. In previous experiences, the top down design had appeared to be an imposed task. Their communication abilities increased, so that they tried to figure out the faculty's taste about sketches and representation graphics for the projects in order to get a good marks. In the individual projects we observed a lot of teamwork. Former teams met at the laboratory room, checking their classmates' work. When several different courses of action appeared, they asked the assistants for help. It was a forward step from the lone work we had seen in previous courses. The reports were complete and the programs worked as required. Each former team member implemented a different alternative and outcomes were compared. We did not ask for it, students did it by themselves finding some pleasure in doing so. After all programming is a substantial part of the career they have chosen to make a living out of.

We also noted that writing and referencing became part of the study habits and students found themselves comfortable with them. In some other courses, they asked one of the authors to postpone some counselling meeting until they could elaborate the material a little further, and when the date arrived, they came with the questions clearly stated, delimited and with some examples worked out. Previously we used to have counselling sessions with general and ambiguous questions, and with students expecting to get another lecture more focused on the test material.

Perhaps the more relevant habit change we saw in our students was in the way they asked for help. They brought us notes and schemes, and the references of where they found the difficulties. That eases our answering and counselling work a lot. They also improved in technical speaking a lot, using terms in a more precise and unambiguous way.

By the second semester of 2001 and the first of the next year some of these students will take another course with us. At this time we will be able to evaluate if the changes in their habits sustained in the long term.

REFERENCES

[1] Hein, T.L., "Writing: A Unique Strategy Designed to Bring Current Topics in Science and Engineering to Non-Majors", *IEEE Frontiers in Education, FIE 2000 Proceedings,* Session T2F, Kansas, MS, 2000.

[2] Sharp, J. E., Harb, J. N., Terry, R. E., "Combining Kolb Learning Styles and Writing to Learn in Engineering Classes", Journal in Engineering Education 86 93, 1997.

[3] Adams, R. S., Atman, C. J. ," Cognitive Processes in Iterative Design Behavior", *IEEE Frontiers in Education FIE'99 Proceedings*, Session 11a6, San Juan, Puerto Rico, 1999.

[4] Sims-Knight , J.E., Upchurch, R. L., "The Acquisition of Expertise in Software Engineering Education", *IEEE Frontiers in Education FIE'98 Proceedings*, Session S4C, Tempe, AZ, 1998.

[5] Gick, M., "Problem Solving Strategies" *Educational Psychologist*, 21, (1 and 2) New York: Laurence Erlbaum Ass. Inc., 1986, p 99-120.

[6] Sims-Knight, J. E., Upchurch, R. L., "Reflective Essays in Software Engineering", *IEEE Frontiers in Education FIE'99 Proceedings*, Session 13a6, San Juan, Puerto Rico, 1999.

[7] Hein, T.L., "Using Writing to Confront Student Misconceptions", *European Journal of Physics*, 20, 1999, pp. 137-141.

[8] Feldgen, M., Clúa, O.: "Introducing Software Design Abstractions using Hardware Dissection", *IEEE International Conference on Engineering and Computer Education ICECE 2000 Proceedings*, Sao Paulo, Brazil, 2000.

[9] Wheeler, E. D. ,Balazs, G. G., McDonald, R. L., "Writing as a Teaching and Learning Tool in Engineering Courses", *Proceedings of the IEEE Frontiers in Education FIE'97*, Session S4F, Pittsburgh, Pennsylvania, 1997.

[10] Huckin, T. N., Olsen, L., *Technical Writing and Professional Communication. For Nonnative Speakers of English*, 2nd Edition, Mc Graw Hill, 1991.

[11] Seat, E., Lord, S.M:, "Enabling Effective Engineering Teams: A Program for Teaching Interaction Skills", *Proceedings of the IEEE Frontiers in Education FIE'98*, Session T2H, Tempe, AZ, 1998.

[12] Jacobson, D. ,Davis, J, "Ten Myths of Cooperative Learning in Engineering Education", *Proceedings of the IEEE Frontiers in Education FIE'98*, Session F3B, Tempe, AZ, 1998.

[13] Johnson, D: W, Johnson, R.T., *Learning Together and Alone. Cooperative, Competitive and Individualistic Learning*, Allyn and Bacon, 1994.

[14] http://www.cea.wsu.edu/TIDEE

[15] Agonino, A., "A Hands-on Discussion of Dissection: Coalitions Lessons Learned", *Proceedings of the IEEE Frontiers in Education FIE'98*, Session F2E, Tempe, AZ, 1998.

[16] Sheppard, S. D., "Mechanical Dissection: An experience in How Things Work", *Proceedings of the Engineering Education: Curriculum Innovation & Integration*, Santa Barbara, CA, 1992.

[17] Feldgen, M.., Clúa, O., "Hardware Dissection in Computer Science as a Tool to improve Teamwork", *IEEE Frontiers in Education, FIE 2000 Proceedings*, Session S1C, Kansas, MS, 2000.

[18] Gentili, K. L., Hannan, J. , Crain, R. W. ,Davis, D. C. , Trevisan, M. S., et al ,"A Process Oriented Class in Engineering Design: How it Works", *Proceedings of the IEEE Frontiers in Education FIE'97*, Session F3H, Pittsburgh, Pennsylvania, 1997.