Proceedings of the NACCQ 2000
Wellington NZ www.naccq.ac.nz

Teaching Software Engineering in a Practical Way

Ken Surendran
Frank H. Young

Department of Computer Science
Rose-Hulman Institute of Technology
Terre Haute, IN 47803, USA
(UNITEC Institute of Technology,
Auckland, New Zealand.)
ken.surendran@rose-hulman.edu

ABSTRACT

Nowadays most Computer Science and Information Systems curricula include a Software Engineering (SE) course. The actual titles used for the course vary depending on the areas within SE that receive primary focus. Overall course content generally reflects current SE practices and includes coverage of the System Development Life Cycle (SDLC) and the ensuing work-flows (both core and supportive) in the system development process. One of the objectives of the SE course is to prepare the students for the real world challenges encountered in professional system development. For this reason system development projects are commonly used in the course facilitation process in the form of assessments.

The authors have been using client-sponsored system development projects in the facilitation of SE courses. In this paper, they discuss the strategies for designing the assignment and the facilitation techniques they have used for simulating real-world experience. Based on feedback from students, the authors discuss the merits of using client-sponsored projects, share their findings concerning the effectiveness of these techniques and suggest schemes to improve the facilitation process.

Keywords:
Client-sponsored Project, Real-world experience, Software Engineering, System Development Life Cycle, Work-flows.

1. INTRODUCTION

Software engineering (SE) is a young discipline. A detailed study (Ford, 1996) on the status of SE found that most elements of SE are still immature. Rapid changes in technology also seriously hamper the maturation of SE. In an attempt to improve this state, there are renewed efforts (Pour, 2000) to define and accredit new curricula that stress SE fundamentals and practices. Such curricula must provide students with experience working in teams and prepare students for lifelong learning.

Recently some universities have started offering curricula in SE *per se*. Many continue to offer SE as a course in their Computer Science curriculum. Globally, there is a very high demand for software engineers, far exceeding the supply of SE graduates. Most often, it is the CS graduates who fill this gap. Hence it is essential to enrich the SE courses in CS curricula by giving students opportunities to practice SE concepts and principles through project work. Different approaches have been reported for providing real-world experience in the facilitation process. In one of them the authors (Villarreal, 1998) combine SE with a Database course and use realistic projects in the course assignment. Another reported strategy (Polack-Wahl, 1999) uses projects to

let students experience the client side of system development through role-plays. (The role-play, however, is used in a course that follows a first course in SE.) Most of the reported strategies emphasize the need to use real world projects in assignments.

At Rose-Hulman Institute of Technology (RHIT) a two-term, compulsory course in Software Engineering is offered to the 3rd year students. (A term equals ten weeks.) Topics include SDLC and ensuing workflows in the system development process. In the first term, the focus is on the primary workflows leading up to the detailed design of a system. In the second term implementation issues and the supportive workflows such as Quality Assurance and Project Management are considered. This SE course is followed by a two-term Senior Project course in which students, in groups of three to five, carry out industry sponsored projects involving system development. The SE course must prepare the students for this comprehensive Senior Project.

At RHIT, the SE students work on various client-sponsored projects and, as part of their assignment, they produce system artifacts as they follow the various SDLC workflows. The selection of projects and the manner in which the assignment is organized influence the level of realism that is brought to the course. At Rose-Hulman, the variety and the design of project-based assessment help simulate real world situations encountered in system development. In this paper, the authors discuss the facilitation techniques they used in the SE course - in particular the issues pertaining to group projects - and share their findings concerning the effectiveness of these techniques.

We first present an overview of the two-term SE course, highlighting its scope and contents. We then review the approaches used over the years in selecting and assigning group projects. The strategies deployed for simulating realism are also explained. A review of the feedback collected from students during the two terms is discussed in order to evaluate the effectiveness of the strategies used. The conclusion discusses possible improvements to the facilitation techniques.

2. COURSE OVERVIEW

The aim of the two-term SE course at RHIT is to educate students about the skills necessary to produce reliable, cost effective and quality software in a systematic manner. Apart from tests, the main assessment consists of, in the first term, producing a feasibility report, a system requirement specification and a design specification for a client sponsored project, and in the second, developing a prototype and the essential accompanying documentation. These activities are all carried out by groups of two or three students.

One chronic problem in SE courses is that students do not understand the importance of written documents (as opposed to code). There is a “chicken and egg” problem here. Students need to have a technical documentation course before the SE course in order to learn how to write the necessary documents. But before they have taken the SE course, students have no awareness that documents are necessary and valuable in a software development environment. Each course should be a prerequisite for the other! At RHIT we have solved this problem by integrating technical writing and SE into one two course sequence (Young, 1991).

2.1. Class and Session Details

At Rose-Hulman, Software Engineering is a compulsory course for the CS major and there are now two classes, each with about 25 students. The students who take the SE course have a good grounding in programming, data structures and other foundational subject areas. As a result, the coding aspect of the SDLC is de-emphasized in the SE course discussions.

The sessions in the first term deal with core workflows and include the following topics: system concepts, software as a product, project proposal and feasibility analysis, software engineering process models, information gathering techniques, systems analysis, requirements specification, architectural design, and system design specification.

Along with the conceptual aspects of each workflow, the diagrams used in related system artifacts are discussed. For instance, in the sessions on analysis and design, all the major system modeling tools pertaining to the procedure-oriented paradigm, such as context diagrams, entity-relationship diagrams, network connectivity diagrams, structure charts, and database schema are discussed. In addition, in view of the growing interest in the object oriented paradigm, a few sessions are dedicated to UML, use-cases and class diagrams.

The sessions in the second term are concerned with the remaining core workflows and the supportive workflows. The specific topics discussed are: fundamentals of system testing, test case design, quality assurance, software metrics, system implementation, change management, risk analysis and management, estimation, scheduling, project management, people management (motivation, negotiation, delegation), installation, training, and maintenance. In addition, towards

the end of the second term, the students hear "lessons learned" presentations from the seniors who completed Senior Projects during the year.

2.2 Project-based Assignments

The course has several formative and summative assessments. The following are examples of formative assessments used in the course:

- Review of intermediary system artifacts (analysis and design),
- Analysis of a failed project (through own literature search)
- Preparation of an implementation plan
- List of ten project management tips (do's and don'ts) picked up from senior project presentations

In addition to three written tests, the summative assessments consist of project based assignments in which the students, in groups of three, carry out all the work-flows of system development pertaining to client-sponsored projects and also present the results of a project their group implemented. Outlines for these intermediary system artifacts are provided to the students. The students are strongly encouraged to use these templates. The following intermediate outputs are evaluated:

- Plan: project proposal and feasibility report
- Analysis: requirements specification
- Design: architectural and detailed design specifications
- Implementation: final project report consisting of system and user notes for the prototype.

3. CLIENT-SPONSORED PROJECTS

In the courses that lead up to SE students have solved well-defined problems assigned to them by course facilitators. These assignments rarely include design. When they enter the SE course, the students are competent coders, but they seldom realize the need to follow a system development process. It is in the SE course that they learn about developing marketable system products that are used by many people and maintained over a long period. In the SE course students need to deal with clients external to the course.

3.1 Project Proposals

Client-sponsored projects have been part of the RHIT SE course since the early 1990s. Potential clients are informed about the nature of the course and the client's role. They are also made aware of the possibility of not getting a finished product at the end of the course. Clients submit outline proposals that are often incomplete. The proposals received usually are from different application domains with their scopes varying widely.

Time critical projects are unacceptable because the SE course is an educational experience as opposed to a production experience. Many clients have some pet projects that do not have high enough priority to receive organizational resource allocations. These projects are ideal for students. The clients will be very happy with prototype solutions or a proof of concept. However, it is essential to ensure that all the selected projects have some analysis and design components and that a liaison person is available for each project.

The lack of adequate scope information turns out to be exceptionally useful in this course. Students must learn to modify their plans as they learn more about the desired product. Projects must be broken up and reassigned when their scope is discovered to be too large. Some proposals may be found to have much less work required for their solution. The course facilitator has to carefully evaluate all project proposals. In addition, the workload must be balanced at every stage of the project assignment.

3.2 Project Allocation

During the first week of the first term, students are briefed about the project-based assignment. They are given a list of all projects with proposal outlines and necessary contact details. By the end of the first week they are asked to form teams (three or two per team) and choose projects for the planning phase. Having several projects to choose from increases the students' motivation and commitment. Teams are given separate projects. At times, two teams may be allowed to work on the same project independently. The clients are informed about these selections and are given the names of students who are working on their projects.

Team formation has not been a problem in the SE course. The facilitator must occasionally assign students

who are unable or unwilling to form teams. Schedule changes may force team changes during the second term. The earlier team composition is maintained as much as possible.

3.3 Performance Evaluation

Evaluation of individual performance is an issue in team assignments. In the RHIT SE course the same grade is normally assigned to all members of the team. It is acknowledged that the contributions from individuals in a group may vary. As long as there are no specific problems of non-contribution by an individual, all members in the group are graded equally. In their final report, students indicate how well they performed as a team and how well they interacted with each other. In addition to the project assignment, there are tests to measure individual performance. Also, all presentations are peer evaluated - three teams selected randomly rate the performance of each team.

4. SIMULATION OF REAL-WORLD EXPERIENCE

The project-based assignment improves skills required for professional software development. Students need to experience the stresses and strains of working in a group, assigning and accepting roles and responsibilities, dealing with different groups (internal and external, such as clients) and using standards / templates. Students experience these when they work on a system development problem given by a client who is external to the course.

Considerable gaps between what is normally taught in a CS/SE curriculum and what is normally required in SE industry have been observed to exist in the following areas: people management (negotiation and leadership, in particular), user interface, configuration management, ethics and professionalism, requirements gathering and project management (Lethbridge, 2000). Client-based projects provide excellent opportunities for requirements gathering and user interface design. Group projects also provide opportunities for negotiation and management. External clients provide opportunities for increasing the professionalism of students.

A few issues relevant for real world experience are considered here. For each of the chosen issues, we discuss its relevance and the strategy used for simulating it.

4.1 Internal Customers

In most group assignments, the groups work on a single problem for most of the term. In the SE course we need opportunities for students to interact with different groups working on different aspects of the same project. There is need to experience the role of being an internal customer (i.e., using someone else's output for further development) and also having an internal customer (i.e., making artifacts for someone else's use).

Different strategies were used for simulating the internal customer concept. In the past, each group was asked to carry out the design for a system that was analyzed by a different group. This made the students realize the need to follow standards in documentation and to communicate with other groups. More recently, this approach has been modified. Teams, which were assigned to work on two projects (where possible from the same sponsor) were asked to switch documents and projects at the end of the following phases: Plan, Analysis and Design.

4.2 Information Gathering Approaches

Gathering information from clients is an important skill in the world of SE. However, client characteristics vary. Some are easy to approach while others require more formal approaches. Opportunities must be created to share the experiences of different groups in dealing with clients. Different projects offer opportunities for using different information gathering approaches.

Because the groups work on two projects and swap artifacts in between, they get to see the techniques used by the other group. At times, they may have to revisit the customer for more information before proceeding with design. This also reinforces the need to review the requirement specification with the client and users. In addition, they also learn about various techniques that were used in different projects during the project presentations.

4.3 Value Judgements

During the design phase of SE, many decisions are made that are not based solely on technical issues. Students must be made aware of, e.g., the business perspective, legal issues, ethical issues.

Client-based projects afford many opportunities to cultivate value judgement. Cost effectiveness is an important consideration. Reliability needs and liability

issues must be considered. Estimates must not be misleading.

4.4 Formal Reviews

Carrying out formal review at the end of each phase is essential. Making the groups exchange artifacts after each phase enforces the review process. In addition, a formal formative assignment is given to students to review all the system artifacts before proceeding to implementation phase. They also review the artifacts produced by other groups. The results of their findings are discussed in the class. For most students, this is the first time they have critically reviewed another student's work.

4.5 Project Size

In the real world, projects vary in size and the experiences may differ according to the size. The fifty students worked on some ten different projects - some small and some big. In order to maintain workload balance, projects were swapped and teams were combined to implement larger projects, and smaller projects were combined for assignment to a single team. Students learn the importance of crude estimates of project scope. Different experiences are also shared with other students through class presentations.

4.6 Planning

Software development requires discipline and planning. Client based projects help SE students realize their own inadequacies in these areas. Some clients use standard student work procedures (delivering needed information either "just in time" or a few days late). Students quickly come to realize both the consequences of this behavior and that they are guilty of the same behavior. Students must also take responsibility for organizing their groups, assigning and scheduling tasks appropriately. Because the projects are not clearly defined, students must do careful planning and revise their plans periodically.

5. EFFECTIVENESS

At the end of each term, feedback from students was collected to evaluate effectiveness of the course. In both of them, there were two broad sections: one dealing with their learning and the other on facilitation techniques. Consolidated responses pertaining to learning related responses for the first term (proposal, feasibility, requirements specification, and design) were shared with the students. The facilitation related responses pertaining to the project assignment are discussed in the following.

Among the responses received, about 75% of the students liked the idea of swapping projects after each phase and about 85% considered that swapping enhanced their learning. About 90% considered that it was helpful to deal with the same client for the two projects involved in swapping. Almost all of them said that swapping artifacts made them realize the importance of doing a professional job and of using templates for the system artifacts. Without exception, all agreed that swapping should be limited to two projects only.

Most agreed during the first term that the project-based assignment helped in learning SE skills. However, only 75% considered that project-based assignments helped them learn supportive work-flows like project management and also implementation (coding, testing). One reason might be that, being good coders already, implementation might not be challenging.

Students found it harder to coordinate team activities during implementation. For various reasons, only about 60% had realized in full what they had planned. Lack of continued client support was a major problem. (Two main clients changed employers in the middle of the project period and eight teams were implementing their projects.) Teams working on a single project were more successful. Teams working on two projects seemed to concentrate on just one. Teams working on large projects had varying results. Those who had elected a coordinator seemed to do a better job. Certainly the experience has made them appreciate the importance of communication. Almost all of them agree that the course as a whole has prepared them well for Senior Project.

The approach taken still needs refinement. At present, the projects chosen vary widely. Perhaps, the range may have to be narrowed so that not more than two teams work on a project and teams working on two projects are avoided. This can be achieved by re-examining the scope before implementation starts. In a few cases, the students had to learn new tools just for implementation. While this is not a bad idea, there may not be time to learn a new tool and apply it within a term. Perhaps, it might help to plan the learning of tools once the architectural design decisions are made and the tools are identified.

6. CONCLUSION

Looking at the performance in the Senior Project, the SE course seems has a very positive impact. SE course is a prelude to Senior Project and the preliminary planning for Senior Project is carried out in the SE course itself (Oexmann, 2000). Senior Project being their first actual (not simulated) real world system development encounter, it is important to ensure that their first experience is positive and pleasant rather than negative. The quality of documentation and the overall products delivered in the Senior Project have improved significantly over the past few years. The project-based assignment plays an important role in preparing the students for the Senior Project and in general improves the effectiveness of the SE course. The clients usually get higher than expected functionality and quality in the products. This is evidenced by the increase in the number of proposals received and the complexity of the projects they lead to.

Learning from failures (our own and others) is a strategy used in SE facilitation. Failure due to lack of communication with the client is a significant learning element that a project-based assignment offers. Individually, we do not have the time to make all the possible mistakes ourselves and learn from them. But the sharing of experiences that take place in presentations helps students learn from the mistakes of others and increases the learning that takes place in the course.

Teams (and the other students) also learn from disasters. A change of clients (or losing the client support altogether) in the middle of a project is an experience that requires maturity and skill to recover from. The frustrations the teams went through and the countermeasures they came up with to overcome their problems further enhanced their learning experience.

SE is still not a mature engineering profession. The gap between what is learned in the curricula and what is needed in the industry is rather wide in relation to other engineering disciplines. To improve the situation, one approach would be to eliminate some of the not so important topics to make way for new ones. However, for now, project-based assignments help develop, in addition to the primary technical skills, quite a few soft skills such as negotiation, inter personal skills. Such assignments address the higher levels of Bloom's taxonomy for educational objectives, viz., application, analysis, synthesis, and evaluation. Thus the project based assignments together with other value-adding formative assessments help prepare the CS graduates to perform SE activities more professionally.

7. REFERENCES

Ford, G. and Gibbs, N.E. (1996) "A Mature Profession of Software Engineering". Tech. Report CMU/SEI-96-TR004, SEI, Carnegie Mellon Univ., Pittsburgh., pp 19-20.

Lethbridge, T.C. (2000). "What Knowledge is Important to a Software Professional?". IEEE Computer, Vol.33, Number 5, May, pp 44-50.

Oexmann, D. and Surendran, K. (2000). "A Win-win Process Model for Industry-Sponsored Senior Projects". Paper presented at the Conference on 'Students and Industry: Enhancing Under-Graduate Project Work with External Clients', Rose-Hulman Institute of Technology, Terre Haute, IN, USA, April.

Polack-Wahl, J.A. (1999) "Incorporating the Client's Role in a Software Engineering Course". Proceedings of the 30th SIGCSE Technical Symposium on Computer Science Education, ACM Press, New Orleans, Louisiana, USA, pp 73-77.

Pour, G., Griss, M.L. and Lutz, M. (2000) "The Push to Make Software Engineering Respectable". IEEE Computer, Vol. 33, Number 5 May, pp 35-43.

Villarreal, E.E. and Butler, D. (1998) "Giving Computer Science Students a Real-World Experience". Proceedings of the 29th SIGCSE Technical Symposium on Computer Science Education, ACM Press, Atlanta, Georgia, USA, pp 40-44.

Young, F.H. (1991) "Software Engineering and Software Documentation: A Unified Long Course". Proceedings of the 21st annual Frontiers in Education Conference held at Purdue University, Indiana, U.S.A., IEEE Education Society,21-24 September, pp 593-595.