

An Attempt to Teach Professionalism in Engineering Education

Conny Johansson, University of Karlskrona/Ronneby, S-372 25 Ronneby, Sweden

Lennart Ohlsson, Utilia Consult AB, Jordvändevägen 18, S-236 35 Höllviken, Sweden

Abstract

This paper describes our experiences from creating a two year software engineering education program at Karlskrona/Ronneby University College where we have included professionalism as a major educational element. As the basis for teaching professionalism we use the concept of commitment culture, a working atmosphere built on voluntarily taking responsibility. A large proportion of the curriculum consists of project courses which are run as role playing games where teachers act as customers and consultants.

Keywords: Software Engineering, Professionalism, Commitment Culture, Role Playing.

1. Introduction

Education is sometimes seen as an activity that is somewhat separated from the rest of society, its role being to provide knowledge and skills in a number of specific areas. The integration of these pieces into an ability to act in society in a professional manner is, however, left to be learned in "real life". Since the purpose of an academic environment is to serve as training grounds for preparatory exercises, it is inevitable that this separation must exist to a certain extent. Many problems in the professional life can simply not occur in a protected environment. Nevertheless, it is worthwhile to consider whether it is possible to narrow the gap between university curricula and the outside world. In particular there are a number of difficult and general problems associated with working together in organized teams, for example managing mutual dependencies and establishing an effective communication between members. If we could make the educational situation better reflect the future professional situation, there are two potential benefits we may achieve. First of all, the students will enter professional life being better prepared in some additional areas. Secondly, the addition of some realism to the education may illustrate how the various conventional subjects fit into their future lives. Adding realism to the education does not have to be done at the expense of other subjects. Because it gene-

rally increases the students motivation for those subjects, it can even result in a net improvement.

The University College in Karlskrona/Ronneby has a profile which emphasizes information technology and close industry linkage. When the University College was founded in 1990, the education program for a two year University Certificate degree in software engineering was developed in close collaboration with the software house Elektronikcentrum AB. In designing this program our ambition has been to focus very clearly on the role of the professional software engineer by letting the students "get their hands dirty". Our ambition has also been that they, as a consequence, acquire a more thorough understanding of the traditional subjects than is obtained with conventional methods.

Software engineering is based on the principles of computer science. It also includes the application of these principles in combination with tools and methods. The most important software engineering skill is the ability to manage complexity. A characteristic which distinguishes a software system from other complex systems is that it is invisible and its structures are essentially conceptual. Since there are no physical laws which constrain the possible structures, there is a fundamental problem of comprehending a software system. Understanding the consequences of making modifications to the system creates a problem in particular. Adding to these difficulties is the fact that most software systems today are so large that they must be developed by several people who must be able to form a coherent view and communicate about their invisible product.

2. Commitment culture

A professional is expected to deliver products or services of good quality. A first requirement is thus that he has the necessary technical skills. He is furthermore expected to predict the time and cost required to complete the task in order to make a business agreement with a customer. In this context good quality should not be taken as an absolute measure. The most useful definition of quality in a business situation is customer satisfaction. This means that the two issues of quality and predictability are closely related since the satisfaction of a customer equally dependent on what is agreed as on what is delivered.

Delivering the specified product or service on time and to the agreed price, is the first requirement on professional behaviour. In many fields this requirement is sufficient. In a field like software engineering, however, the problem to accurately specify the system to be developed is often non-trivial. In this situation a professional must take active part in assuring that the specification adequately reflects the client's true requirements. Writing a very detailed specification is not always sufficient because software development is often a so called "wicked problem", i.e. the problem can not be defined until the solution is

known. Ways to overcome such obstacles typically involve prototype development, partial deliveries and tight loops of feedback from the users. A professional software engineer should master such techniques and take the responsibility for applying them when appropriate. During this learning process, the client is likely to discover that he requires modifications to the system. If they can not be made within the agreement, the professional takes the initiative to elaborate on the possible alternatives including their consequences in terms of budget and schedule. In this case the negotiations have a positive basis, aiming towards an improvement of the original agreement. There is also the opposite, more difficult situation. A professional must be able to deal with those unfortunate, but inevitable, situations when he is unable to keep to the agreement. Sometimes a written contract specifies the consequences, but sometimes it does not. Often an alternative way ahead can be found that minimizes the losses for both parties.

The foundation for a professional attitude is captured in the concept of commitment. If somebody is committed to do something you can rely on that what has been agreed will actually be done. Commitment involves both keeping promises and avoiding to make promises that can not be kept. Humphreys (1987) has given the following characteristics of a commitment:

- The person making the commitment does so willingly
- The commitment is not made lightly
- There is an agreement on what is to be done, by whom and when
- The commitment is openly and publicly stated
- The person responsible tries to meet the commitment, even if help is needed
- Prior to the committed date, if it is clear that it cannot be met, advance notice is given and a new commitment is negotiated

The voluntary aspect is particularly important, because it distinguishes a commitment from the more common situation of being assigned a task. When you are committed to do something, you actively want to do it. An assignment on the other hand is more based on "doing as you have been told". Taking an assignment is of course usually part of a voluntary agreement on a higher level, for example an employment. Nevertheless there is a clear distinction in the working culture.

In a commitment culture active responsibility is used as driving force in everyday work. Although a commitment culture is more based on the carrot than the stick, it does not require that rewards are tangible, like bonuses or extra vacation. One of the more important rewards is personal development. In a commitment culture, commitments are not only

made to external customers. Members of an organization also make commitments internally when they divide up work among themselves. In a commitment culture there is no difference between large and small commitments; they are all considered seriously.

A common stumbling block when you establish a commitment culture, is that it requires everybody to be sufficiently skilled negotiators. Many people are not used to publicly assert what they can do and what they can not do under given constraints. The principles are simple but what is often lacking is the habit to apply them. This, in turn, leads to insufficient amount of practice. A commitment culture requires an open atmosphere where each individual wants to interact, is unafraid of trying and does not risk a penalty if failing.

The second problem when working in a commitment culture is project control. Project control is the art of balancing the commitment triangle of quality, cost and time. It can be divided into estimation, planning, tracking and obtaining customer feedback. The project plan is the tool to make predictions about schedule and budget. Tracking, in combination with customer feedback, is used to decide whether corrective actions are needed.

When you elaborate the problem of project management further, it is useful to separate quality management from pure project management. Project management is concerned with how resources should be distributed on the various activities and deciding internal deadlines for intermediate results. Quality management is concerned with how to define progress, i.e. how to know that, when a certain amount of the resources are consumed, a corresponding distance have been covered towards the goal of customer satisfaction. All the well-known techniques for project and quality management are additional tools that can help meeting commitments. Examples of such supporting techniques are documentation structures as a manifestation of progress, a standardized development model as a generic plan, risk assessment to optimize the certainty of predictions, configuration and version management to allow development, bug-fixing, requirement changes to proceed in parallel, analysis- and design methods and rapid prototyping to obtain early knowledge of the customer's real requirements, test strategies, inspection procedures and associated metrics to measure progress.

3. Role Playing Projects

We are using the idea of commitment culture as the foundation for an attempt to teach professionalism in our education program. In order to give the student sufficient opportunities to develop such a culture, we have three project courses during the two years. Together they make up 40% of the whole program. The first project course, which takes place towards the end of the first year, the students do small individual projects lasting for five weeks. In the following two project courses the team size and the project size are

then gradually increased. In the second project course the team consists of four to five students and the project lasts for ten weeks. In the final project course the students work in teams of twelve to fifteen on a fifteen week project.

The voluntary aspect of a commitment can be hard to achieve in an educational framework with its long history of training the students to await instructions and obey the teacher. To add realism to these courses, they are run as role playing projects, similar to the approach described in Jaquot et al (1990), with teachers acting as clients. Rather than just being given an assignment and a deadline, as is common in conventional project courses, the students are in this setup encouraged to actively negotiate with their customers. In real-world problems, the task is often fairly fixed and what is negotiated is delivery time and price. In an academic setting on the other hand, the "price" is fixed in terms of the number of credit points given for the course. Similarly, the delivery time must also be fixed in order not to interfere with other courses. What remains to be negotiated is the contents of the task in terms of functionality, robustness, documentation, progress demonstration, etc.

In conventional project courses the students are often given only a little amount of feedback. Whether they will pass the course or not they have little knowledge of before they hand in their solutions. In simulating the real world, an important part of the negotiations is how to continuously assess the client's satisfaction. The developer must continuously be able to present proofs of progress in some form. The customer is then required to either commit to the approach taken and the results achieved so far, or to communicate the modifications he sees necessary.

In addition to maintaining a good customer relation, the first project course is mainly focused on solving technical difficulties. In the other two project courses the focus is shifted towards problems like managing the complexity of a large system, handling evolving requirements, splitting a piece of work on several members of a team and creating effective communication among the members.

In the project courses we are more concerned with the process of working than the actual results produced. Therefore, the only thing which determines whether a student passes the course or not is if he has fulfilled his commitment, delivered according the agreed product on time. The rule regarding the deadline is very strict. If the student misses the deadline by one day, he will not pass the course unless an agreement has been made with the customer. Grounds to accept a delay can be: notification with valid cause, early partial delivery, etc. Even a last minute delivery runs the risk of being rejected if it contains defects and the student has had insufficiently arranged possibilities to get advance feedback from the customer.

These strict rules must be balanced by the way the teachers act their roles as clients. Most often the students mis-judge the efforts needed, but the point is that they should

discover this for themselves. When they do, the clients are very reasonable and always agree to re-negotiate the tasks.

3.1 Individual project

The primary objective of the first project course is to introduce a working style based on a commitment culture and the ground rules of the role playing game, in particular the negotiation part. This course runs full time during five weeks and the size of the programs delivered is approximately 5000 lines of code. The students are shown a number of existing applications and asked to do "one of those". Presenting the tasks in this way naturally motivates the students to spend substantial effort on early activities like requirements analysis. Some examples of applications developed in this course are games, simple spread-sheets and process kernels for embedded systems. In the first course, the customer is only the user of the program. The required delivery only includes an executing program and user documentation, unless additional deliverables are agreed in negotiations. The voluntary aspect of a commitment is emphasized throughout the course. The students are encouraged to re-negotiate their tasks by the customers who are rather complaisant with students who find the situation difficult. In addition to teachers playing customers, there is also a more traditional role of technical advisor. To facilitate role playing, no teacher plays the role of both customer and technical advisor to the same student.

3.2 Group project

In this course the students work in teams of four to five during ten weeks. The course consists of two parts. In the first part a new system is developed, and in the second part the teams are asked to make modifications to those systems. The parts are both run as complete projects with the development task approximately twice the size of the modification task.

The goal of this course is to informally introduce project planning as a refinement to making commitments. The small team size in this course makes it possible to work in an intuitive, informal organization. Just working together, however, always poses problems which the students here have the opportunity to solve on their own. This creates the appreciation basis for a more strict organization which is introduced in the final project course.

The project management in this course practices tracking and successive planning. Achieving quality, i.e. customer satisfaction, must be addressed throughout the project life-

time. The key is to achieve mutual commitment, to have the customer involved in order to get continuous feedback and hopefully a blessing of the approach taken. Minuted meetings with the customer are used to formally agree progress. Quality control is transformed to the problem of having sufficient information at these meetings in order that the customer can make competent decisions.

In this course the teacher's role of technical consultant is augmented with a function of a quality controller. His task is to assist the teams to organize their work internally, but his role is only to give advice and let the student decide for themselves how they want to work. He participates by being present at the groups' meeting once a week.

Including a modification task in the course, motivates development in such a way that it facilitates future unknown changes. The problems of maintenance is made realistic and very concrete by having the groups switch systems so that everybody is confronted with someone else's code and documentation. The need for different kinds of system documentation is thereby further illustrated.

In parallel with the group project course a course in project organization and human work science is given. This course highlights topics like planning, distribution of work, leadership, conflict management, compromises and communication difficulties. The concrete experiences from the Group Project course provide motivation and experiences to actively discuss and find solutions to these issues. This course also covers the concept of commitment in more detail and its basis in that everybody must take active and voluntary responsibility to find and give information and to maintain an open and communicative atmosphere.

3.3 Large project

In this course, which runs for 15 weeks, the students work together in teams of 12-15. The first year this course was run meant that all students were on the same team. The project is divided into a pre-project phase and a development phase. During the pre-project phase, which constitutes about one third of the course, the problem is defined and the requirements are determined.

The development model used is based on evolutionary delivery, which is characterized by early and frequent iteration, where the requirements are redefined after each delivery based on the feedback from the user, Gilb (1988).

At this point in the education, the students have gained an appreciation for quality management. In the pre-project phase they develop a quality plan for the project. The quality plan defines procedures for configuration management, quality assurance and the various kinds of documentation.

The project leader of this course is an external software consultant. As before, the customer roles are played by teachers. Examples of other roles in the organization That are defined in the quality plan are subproject leader, configuration manager, delivery manager, documentation manager, inspection leader. These roles are all played by students.

4. Conclusions

The students were enthusiastic about the role playing approach used in the project courses. It turned out that already in the individual project, the students needed very little technical guidance and support. They were able to find answers to their problems themselves in manuals, literature or sometimes from the tool suppliers. We learned not to underestimate their technical abilities. On the other hand, the students found out that deceptively simple problems of making and keeping promises hide many difficulties to be mastered. Negotiating with a customer was a new experience to most students and in the first attempts they generally accepted too much work. They quickly learned the rules of the game, however, and soon started to explore the limits of their power. A serious issue is the grading of the project courses. So far we have graded a group as a unit, either pass or fail, and the main grading criteria has been the performance from the customer's point of view. Whether this is the right way to do it, we do not yet know. Running the course in Project Organization and Human Work Science in parallel turned out to be successful with mutual benefits to both courses. Having recently made concrete experiences, the students were strongly motivated for discussing problems of human interaction. It is always hard to say for sure whether the students have obtained a deeper knowledge of the traditional subjects than they would have with an hypothetical alternative approach. It is, however, our impression that we have succeeded in this respect.

Acknowledgements

We are indebted to many people who have taken part in this work. First on the list is Tomas Althen, without whom this program would not have existed. Secondly we want to thank the teachers, Johan Larsson, Per Dagermo, Roger Hedenäng, Lars Lundberg, Michael Mattsson and Peter Stevrin, who have contributed their ideas and efforts. Finally, we thank the students who have given us their guidance.

References

Jacquot, J.P. Guyard, J. Boidot, L. (1990) Modelling Teamwork in an Academic Envi-

ronment, in Proceedings of the 4th SEI Conference on Software Engineering Education (ed. L. E. Deimel), Springer Verlag, Heidelberg, 110-122.

Humphreys, W. S. (1987) Managing for Innovation - Leading Technical People, Prentice-Hall, Englewood Cliffs, NJ.

Gilb, T. (1988) Principles of Software Engineering Management, Addison-Wesley.