A software quality and software process improvement course based on interaction with the local software software industry

M. Letizia Jaccheri

Norwegian University of Science and Technology,
IDI/NTNU 7491 Trondheim, Norway
letizia@idi.ntnu.no
tel: +47-73-593469 fax: +47-73-594466

Abstract

This paper describes a redesigned course in software quality and software process improvement. The course encompasses lectures and project work in groups. Part of this work project is done in collaboration with the local software industry. The proposed interaction pattern with industry is the main contribution of this work. This interaction model is based on research questions, presentations from industry, and subsequent presentations and document writing by students. Both other related courses in software quality and process and other software engineering courses in
which industry interaction is required can adopt this interaction model.

The paper also presents related work by surveying general industry interaction patterns and courses, which provide process and quality education.

1 Introduction

In 1997, I was asked to redesign and teach a Software Quality and Software Process Improvement course at the Norwegian University of Science and Technology (NTNU). The existing course was based on a textbook [1] augmented by extra examples providing many figures from North American and European software industry and three mandatory group exercises. The existing course used to provide few "qualitative" examples (or motivating stories). Student participation in class was minimal.

The students who take this course are both 4th year undergraduate and graduate students. The course can be chosen by both software engineering students and business and administration ones. Student software engineering background includes a main software engineering course in the second year, a project course without external customer, and a project course with external customers.

My teaching background includes software engineering courses which we have mainly designed and run around projects in co-operation with industry [2].

The problems, when redesigning the course were: which software quality and process improvement methods and models to choose? Moreover, which teaching methods? This last question asks for a model which combines teaching in class
and practical exercises.

The main idea underlying this work is to link the course to the local software industry [3] through the SPIQ \(^1\) project. This will provide the students with concrete examples. In addition, this will provide the teacher and the students who are interested in process and quality research with a research background.

In this paper, a simple model of interaction with industry is presented. This model is based on research questions, presentations from industry, and subsequent presentations and document writing by students. This model can be seen as a concrete contribution to software quality and process education. Moreover it can be seen as a general model which can be adopted by other software engineering courses in which industry interaction is foreseen.

The structure of this paper is as follows: section 2 introduces the field of software process improvement and quality education and motivates the need for such education in general and for our course in particular (section 2.1). It also provides a brief history of the origins of this course and of its goals (section 2.2). Section 2 ends with a general discussion (section 2.3) about industry interaction patterns as described in the literature.

Section 3 describes our main choices and gives an evaluation of the first two iterations of the revised course. Section 4 compares our work to related work in the field of software process improvement and quality education. Conclusions are given in section 5.

\(^{1}\)This work was supported in part by the Norwegian Project Software Process Improvement for better Quality (SPIQ).
2 Background

2.1 Software process improvement and quality education

It is only recently that universities have started to offer courses in topics such as software process and quality. Traditionally these topics have been partly covered by other courses, such general software engineering courses, or they have been neglected.

The need for this kind of courses mainly stems from industry and it is well motivated in literature. For example, the survey of Lethbridge [4] gives, among other things, the following figures: On the relevance of formal education for their present job:

- For testing and quality assurance, the participants said that their learning from university courses was rated at 1.2 (measured on a scale with 0.0 as worst and 4.5 as best), while their present knowledge in the field was rated at 3.5.

- For configuration management there was a rating of 0.6 for what they were taught, compared to 3.0 for present knowledge.

- For process standards the rating was 0.4 (being taught), compared to 2.6 (present knowledge).

On the “perceived importance of the software topics”:

- Participants gave career importance of testing and quality assurance a rating of 3.7 (same scale as above), their desire to learn more in this area
was rated 3.8, while the corresponding university emphasis was rated to 2.3.

- For Configuration Management the ratings were: 3.3 (career importance), 3.2 (desire to learn more), and 1.7 (university emphasis).

- And for Process Standards the ratings were: 2.5 (career importance), 2.8 (desire to learn more), and 1.5 (university emphasis).

In the Guidelines for software engineering education [5], both software quality and software processes are listed as two of the 11 software engineering modules for a software engineering curriculum.

The need for a course about software process and software quality which encompassed also topics such as configuration management is recognised at both national Norwegian level and international level.

2.2 The Software Process Improvement and Quality at NTNU

Before 1997, quality and improvement issues used to be covered by an existing software engineering course and it was in 1997 that a revised course on Software Process Improvement and Quality was established.

Our goals, when designing the course can be stated as follows:

G1 Improve the quality of provided education in terms of more participation with respect to the quality and improvement module in the old course.

G2 Learn about local software industry.
G1 is clearly an educational goal. G2 is primarily an educational goal. If we establish contacts with the local software industry, we will be able to provide concrete examples which motivate the students in a more effective way than examples which are found in books and articles. In addition, G2 can be seen as a research goal as the contacts with the software industry should provide background for process and quality research for interested students and for the teacher.

2.3 Interaction with Software Organisations

In this work, the problem is that of interacting with software organisations to enable the students to get hands on their quality and process issues.

Software engineering education presents a set of challenges which have been thoroughly addressed in the literature (see for example [6]). One crucial issue is that of designing and running projects in collaboration with industry [7] [8]. As far as our knowledge, the existing literature concentrate on projects which incorporate clients expecting a software product. A fourth year course based on such kinds of projects is already offered at NTNU [9].

On the basis of our literature study and our experience, we have three alternatives (see figure 1) for designing the practical work or projects, to be performed in our course:

1. The first is the "null alternative" according to which the teacher designs one or more projects to be performed by the students without letting the students interact with any software organisations. The projects may be
based on real projects, but there is no real industry interaction. This is the case of our second year software engineering course at NTNU.

2. This second alternative is based on the classical project collaboration model (as described for example in [8]) according to which students work to produce a software product for a real customer of a real organisation. In this model, the students play the standards roles of analysts, designers, implementers, and testers. In addition, to cover quality and process issues students play process and quality roles, such as configuration management, quality assurance, and project administration (see [7]). This is the case of our fourth year project course at NTNU [9].

3. Our last alternative is a model in which the students work in collaboration with organisations at the process and quality level. In this model, the students have to play the role or at least they have to observe the work of quality and process managers.

Alternative 1 does not contribute to fulfil our goal 1 (Improve the quality of provided education in terms of more participation with respect to the quality and improvement module in the old course) as one of the main criticisms against the old course was that the examples were not concrete enough, therefore they were not motivating. Goal 2 (learn about local software Industry) can be fulfilled if the teacher already has good examples from the local software and manages to design the practical work so that students emulate the way local software industry works.
Figure 1: Different kinds of software projects.

Alternative 2 would increase participation. Our experience with several courses based upon production of a software product for a real customer [9] [2] tells us that student participation is high in this kind of courses. The disadvantages of this alternative is that process and quality issues often require large projects to be fully understood. Configuration management, for example, is not a real problem when six students work together for three months to deliver a software which is never maintained. Also, the students tend to focus on technology more than on document reading and writing. The risk is that students would fail to appreciate the difference between this course (on process and quality issues) and the other course (on software development issues).

Alternative 3 is our ideal one. If students are able to work close to real or-
ganizations as quality and process contacts they will be able to experience real
examples of quality and process issues. This would be interesting for them (goal
1) and both students and teacher would learn about the local software industry
(goal 2). It is relatively easy to convince organisations to share product and
technology problems with students (at our department, each year we get more
requests from organisations than we able to serve by our 4th year projects).
However, a first investigation showed that organisations are not willing to share
their quality and process issues with us. The reasons are twofold: Small or-
ganisations have ad-hoc solutions and they do not really have roles which are
specifically dedicated to process and quality issues. Large organizations, such as
telecom providers, cannot easily let 50 students work with their quality manuals,
process standards, etc..

After choosing alternative 3 above, we have again three alternatives of pos-
sible interaction with software companies:

3A Students work inside companies at quality and process level.

3B Students visit companies (structured interviews, questionnaires, access to
relevant documentation, etc.) to learn about their quality and process
initiatives.

3C Representatives from companies visit the university and share experience
with students, mainly by giving presentations.

We chose this last alternative (3C) because, as mentioned, organizations do
not have the capacity to let many students work at their process and quality
level. On the other hand, alternative 3B cannot be accepted as our computer science students do not have enough insight in research methods (neither qualitative nor quantitative). A solution could have been to teach these methods in the course, but for timing reasons and also not to loose the focus of the course, we decided not to do it.

In other words, we first chose alternative 3 \textit{(students work in collaboration with organisations at process and quality level)} and then alternative 3C \textit{(representatives from companies visit universities and share experience with students, mainly by giving presentations)}. In order to let students participate actively, we developed a model in which students have to produce documents and presentations which present their understanding of organization initiatives.

3 The course

The revised course was run the first time in 1998 [10]. In this section we show how our course model was organized for year 1998. We give a summary of the course evaluation which is the basis for the changes we implemented in 1999. A summary and an evaluation of course offered in 1999 course is also given.

3.1 The course during 1998

3.1.1 Summary

The students were provided with the following inputs:

- the textbook is Humphrey’s original book on Maturity Model [1]. It was
textbook used earlier. Although the book seems outdated, it is cited by almost every book on process and quality issues, and we regard it as a good starting point to study process issues for the first time.

- The lectures concentrated on ISO9000, TQM, and SPIQ. We also give some practical instructions about testing, inspection, configuration management (CM), and process modelling (PM), which are the topics of some of the exercises.

- For year 1998 there were three persons from three Norwegian companies (Statoil, Novit, and Ericsson Norway) who gave presentations according to a set of questions which had been previously agreed on with the teacher. The lectures were tape recorded.

The research questions were:

- Q1: Why is process improvement important [in your company]?
- Q2: Which processes does your company have?
- Q3: Which improvement initiatives does your company implement?
- Q4: Which relationships exist between software improvement and software quality?

Each group (10 groups each of 4 students) were expected to produce 6 reports and 3 presentations:

1. In the first exercise, students were asked to inspect a piece of C++ code
(approximately 700 lines). Checklists and instructions were given to students. The result was an inspection report.

2. In the second exercise, students were asked to test a C++ program against its specification (the source was almost the same size as in the inspection exercise). Students have to use a CM tool. The result was a test report.

3. The third exercise was about process modelling (PM). The students were supposed to model a toy software process by mean of a formal process modelling language. The result was a process model report.

4. Concerning the fourth exercise, also called a case study, the students were supposed to give 3 presentations and write 3 reports about their understanding of the answers of the three industry representatives, also in relation with the rest of the course.

3.1.2 Evaluation

Students generally showed up at lectures, industry presentations, and participated in student presentations. Industrial representatives were positively impressed by the initiatives. It was pleasant to teach such a course, though there was a lot of organizational work.

Reasons for implementing corrective changes were:

- Superficiality: Each group was able to reconstruct three "stories" about the process and quality initiatives at the three companies. In the end, students produced thirty documents. The quality of the single document was
rather poor, due to both the work load on the students during the semester and because the students received a superficial vision about the company by meeting only once with a person from each company. Although I had made three audio tapes, no student asked to borrow these tapes but they only used the materials distributed by the company representatives and their own notes.

- Little emphasis on software quality: The nature and the ordering of the questions (see section 3.1.1) gave much more emphasis to process issues than quality ones. Moreover, the invited persons were all process people who used very little time to answer the last question about quality and its relationship to process.

- Little interest for exercises on inspection, testing, and PM: This may be due to the fact that no student presentation in the class was required for these exercises. Also they preferred to concentrate on the bigger case study, than on these toy exercises for which external organizations were not involved.

3.2 The course during 1999

3.2.1 Summary

For Spring 1999, the following changes/improvements were made:

A There was only one company (Ericsson Norway) represented by three persons: the quality manager, the process group leader, and the leader for a
B Students were asked to give presentations about all four exercises.

C Concerning the process modelling exercise, the process to be modelled was a fragment of a real process description provided by Ericsson. Among the ten groups of students attending the course, five were supposed to use one process modelling language (E3) and five used another one (IDEF0). This was a starting point toward an evaluation of process modelling languages for software process improvement.

D The questions for the company representatives were:

- Q1: Describe a software system, also by help of quality attributes.
- Q2: Which processes exist around this system?
- Q3: Which are the improvement initiatives around these processes?
- Q4: How general is the software system and the respective improvement initiatives in the general context of the company?

Each group was supposed to take care of only one question. In this way we would get ten resulting stories. Three (of the ten) groups were allocated to questions Q1 and Q2 and two groups were allocated to questions Q3 and Q4.

Changes A and B can be regarded as corrective changes, while C and D as improvement changes. The modelling of a real process was more motivating than the modelling of a toy process. The process was explained by an industry person
and students developed some understanding of how that process was related to the general organization processes. Also the students knew that their models would be submitted again to the industry people. It was also more motivating for the teacher to go through student models when they represent fragments of reality than when they represent toy processes that had been modelled several times before.

3.2.2 Evaluation

The new questions were better than those from the previous year as they guided the industry people to start from a concrete software system and to relate the process and quality issues and initiatives around a concrete example. Concerning the industry case, the quality of the resulting student documents was undoubtedly better. They showed a deep understanding of the quality and process initiatives of the company and they also referred to discrepancies between the different answers by different representatives, as shown in table 1. Also the PM exercise provided valuable information for process research [11].

Almost each group borrowed the video tapes to get deeper insight in the presentations. There are two hypotheses about why students borrowed the videos, while no student had borrowed the audio cassettes the year before. The first possibility is that video tapes are a more familiar media than an audio cassette. The other one is that the exercise was less superficial and students needed a deeper understanding of the representatives’ presentations.

For the written exam, students were asked to re-answer questions Q1 (De-
Table 1: Different industry roles gave different emphases to different questions.

<table>
<thead>
<tr>
<th>Role</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality manager</td>
<td>no</td>
<td>little</td>
<td>CMM, ISO9000, TQM, Re-organization</td>
<td>CMM, ISO9000, TQM, Re-organization</td>
</tr>
<tr>
<td>Process group leader</td>
<td>no</td>
<td>yes</td>
<td>process maintenance</td>
<td>process maintenance</td>
</tr>
<tr>
<td>Project group leader</td>
<td>yes</td>
<td>little</td>
<td>Code generation, Design oriented</td>
<td>little</td>
</tr>
</tbody>
</table>

scribe a software system, also by help of quality attributes?) and Q3 (Which are the improvement initiatives around these processes?). Also the students who did not work with a specific question during the semester still showed a reasonable understanding of the topic at the exam.

4 Related Work

Here we have selected three courses which also address software process and/or software quality issues and which are project oriented. For a more complete survey of project oriented courses, see [12].

- The Real World Lab (RWL) [7] is a series of undergraduate software engineering courses started in fall 1992 at Georgia Institute of Technology. The students are involved in real industrial projects, products, and customers. RWL emulates an industrial organisation as closely as possible: projects are accepted from industry sponsors who act as customers
providing consulting, direction and resources.

The overall process objective is to improve the RWL processes by applying the CMM process improvement framework. To this aim, RWL carries out both parallel software development projects, and process improvement activities. Each software development project focuses on specific product development. During the development project, students experience RWL processes, and are able to recognize problem areas and devise solutions.

According to the CMM assessment, the following process improvement activities were performed:

- Initial survey and analysis: one student (playing the role of Software Quality Assurance manager) oversees the assessment process and gathers related data, being compiled and analysed to determine the overall attitude of the organization.

- Follow-up interviews: by following the CMM guidelines, both project leaders and customers were interviewed and involved in group discussions.

- Key process area analysis: interview results were compared with previously gathered survey data. Priority improvement areas were determined, and suggestions expressed during interviews and discussions were contrasted to improvement criteria, to create an action plan.

- Improvement implementation: actions were applied to the RWL process.
The role of the customer was always played by an industrial organization. Interaction with the customer is the full responsibility of each single team.

The idea presented in [7] of structuring several project courses as a laboratory was good, and we were also considering it at our institution. Also, we believed that our industry interaction model presented in section 3 would well fit a laboratory organization. By merging the laboratory idea with our industry interaction model, students would both experience process and quality issues as performers in a semi-real laboratory context, and they would observe real organizational process and quality issues in industrial settings. Eventually students could interact with the same organization which has played the role of the customer during product development.

- Another interesting software engineering project-based course is the one held at the University of Texas at Austin [13]. It is an undergraduate course that lasts two semesters. The team structure distinguishes between technical and process roles. There are two main phases, one for each semester. During the first semester students develop the process plans. Moreover they develop simple tools to automate the process (responsibility of the project administration team); to support the generation of test case documentation (responsibility of the testing and evaluation team); to produce functional specification and high-level design (responsibility of the Design Team and the Implementation Team). During the second semester a detailed design and implementation have to be produced. The process also prescribes weekly walk-through sessions of the emerging design. The
process is made visible via status meetings. The plan is developed by the students according to IEEE standards.

There are disjointed technical sub-teams (design, implementation, testing and evaluation, documentation) and process sub-teams (configuration management, quality assurance, and project administration). During the second semester two extra teams are added: Technical Lead and Copilot. They act as a configuration control board, and are responsible for change request management. The idea of defining both product and process roles is interesting [13]. The main disadvantages of this proposal is that students are asked to work with process and quality issues mainly before project start. In this way students base their decisions on theory rather than on experience. We believe that our industry interaction model could be merged with this model as well. In this way students would get also real examples and not only semi-real ones.

- At the Ecole Polytechnique de Montreal [14], a software engineering course is held in the fourth year of the computer engineering program, and lasts one semester. The project carried out during the course is inspired by the Capability Maturity Model and the Trillium Assessment Model. The latter is a framework of questions to perform self assessment of a project. The students are asked to use a simplified version of the Trillium Model to assess their project. The process model is an existing model designed and used by professional engineers at Software Engineering Research Laboratory. The process is described by a specific formalism, and it consists
of 5 phases, each terminating with a walk-through. The phases are: spec-
ification, planning, analysis, realization, and integration. The role of the
customer is played by the teacher.

The good idea of this project is that the industrial participation in the
project is represented by a Trillium evaluator from Bell Canada who did a
formal process maturity evaluation of each team. The evaluator presents
the results to the class, and holds a private three-hours meeting with each
team to reach a consensus for each evaluation road map.

5 Conclusions

The course presented in this paper is based on: a textbook; some extra lectures
that relate the material presented in the book with other improvement methods;
and group project work. Concerning project work, we have presented a model
of collaboration with the local software industry. According to this model, a set
of predefined research questions were developed by the teacher and agreed in
advance with representatives from the software industry. These representatives
were asked to answer these questions in a way they can provide students with
real examples. Moreover, students were asked to elaborate such answers in form
of presentations and reports.

Our experience with two iterations of this course gives evidence that this
model contributes to satisfy our goals. Students participate in company and
other student presentations. This contributes to satisfy our goal \[G1\] Improve
the quality of provided education in terms of more participation with respect to the quality and improvement module in the old course. The second iteration of the course resulted in 10 student reports that have been actively used by the teacher and other researchers as a background for process research [11]. This contributes to satisfy our goal [G2] learn about local software Industry.

The model is reusable, changeable, and extensible. After the first iteration we implemented two main corrective changes (see section 3.2.1) and two improvement changes. Among the improvement changes, we modified the process modelling exercise from a toy modelling exercise to the modelling of a real process fragment, provided by the interacting software organisation. We plan to gradually do the same with the other two exercises (testing and CM and inspection). An idea is to get a real subsystem that students can test and inspect according to given standards and routines. This is not easy as it is not always possible to isolate a stand-alone subsystem from the complex systems which industry deals with.

References


Acknowledgements

I want to thank Alf Inge Wang and Torgeir Dingsøy who were teaching assistants in this course during 1998 and 1999, respectively. I also would like to mention Dr. Patricia Lago from Politecnico di Torino for many interesting
discussions and common work in the field of software engineering education.

Finally, I thank Prof. Reidar Conrad for his comments on this work.