Marcelo Jenkins, University of Costa Rica

Marcelo Jenkins obtained a B.S. degree in Computer and Information Sciences at the University of Costa Rica in 1986, and a M.Sc. and Ph.D. degrees from the University of Delaware, USA, in 1988 and 1992 respectively. Since 1986 he has been teaching computer science at the University of Costa Rica. From 1993 until 1998 he coordinated the Graduate Committee, and from 1998 through 2001 he was the Chairman of the Department of Computer and Information Sciences. His areas of interest include software engineering, software quality assurance, and object-oriented programming. His research interests are in software engineering and software quality assurance and has authored more than 25 technical papers on the subject. As an independent consultant, he has worked with some of the largest software companies in the Central America region in establishing software quality management systems. In the last 12 years, he has taught several seminars on software quality assurance and software project management. Dr. Jenkins is an ASQ Certified Software Quality Engineer (CSQE) and a member of the ASQ Software Division.
Experience in Teaching Software Quality Management
at the Graduate Level

Abstract

Relatively little has been published on the issue of teaching software quality management. This paper reports a case study in teaching a graduate-level software quality management course using industry assessments as a learning vehicle. We explain the contents of the course, describe the process assessments the students perform at local software organizations as their term project, and summarize the outcome and lessons learned during six years offering this course. The issues discussed in this paper might help educational institutions and college professors in designing and implementing software engineering courses at the graduate level.

1. Introduction

Software quality management (SQM) is defined as the set of activities to direct and control a software organization with regard to the quality of its processes, products, and services. SQM encompasses the planning, control, assurance, and improvement of the organizational quality system, defined as the collection of policies, objectives, procedures, measurements, methods, and tools that are established and maintained by the organization to fulfill the requirements of the customer.

A good SQM system is an essential ingredient for software organizations to be competitive in today’s global information technology market, but at the same time SQM is certainly one of the most complicated subjects to teach at any level for the following reasons:

1. The huge size of the topic in terms of the number of software quality management models and standards available today (see Figure 2 below).
2. The lack of any substantial job experience in the subject from most of the students.
3. The difficulty of designing a term project the students can do in a real-world environment.
4. The difficulty of learning something that, unless it is observed and lived in a real environment, the students will not get a good understanding of the concepts underlying the theory and begin appreciate its benefits.

This paper reports a case study in teaching a graduate-level software quality management course using industry assessments as a learning vehicle. The issues discussed in this paper might help educational institutions and college professors in designing and implementing software engineering courses at the graduate level.

2. Related work

Relatively little has been published on the issue of teaching software quality management. Deane et al. describe a classroom exercise in which the firing of a Roman catapult is used to teach
process variation to business students. Their experience is interesting but narrow in the sense that it proposes a technique for teaching just one particular concept within SQM.

Bamberger et al.\textsuperscript{2,3} describes the Software Process Practicum, a one-semester course on software quality management and process improvement at the graduate level in which the students do a process improvement project in a real-world organization. Their experience is very similar to ours regarding the collaboration between academia and industry and the focus of the subjects taught in their course.

Dingsøyr et al.\textsuperscript{4} reports the experience of designing and teaching a senior-level software process improvement course around industrial case studies drawn from the local software industry in Norway. Besides invited industry presentations, the students conduct surveys of software organizations and observe their process improvement initiatives. As in our case, their students do not work in a software development project but rather assess a process in a real software organization, thus providing the students with a taste of an implemented process.

The model for an undergraduate software engineering curriculum proposed in D. J. Bagert et al.\textsuperscript{5} proposes a list of nine software engineering courses that can be part of a Computer Science or a Software Engineering degree. The list includes a one-semester course on software quality assurance but includes no guidelines or suggestions on how to teach the topic.

Jaccheri et al.\textsuperscript{6} describe a software process improvement course in which the students are given a documented quality manual that contains a general process model written in a formal software process modelling language and they have to perform improvement initiatives.

Gannod et al.\textsuperscript{7} propose a list of four software engineering courses at the junior and senior levels that cover many topics in software engineering, including but not centred in software quality. The students develop at least two group projects that try to simulate a real world experience.

Additionally, the Software Engineering Institute (SEI) offers training programs for professionals who want to become authorized instructors or lead appraisers for the Software Capability Maturity Model (CMMI)\textsuperscript{12}, the Personal Software Process (PSP)\textsuperscript{19}, or Team Software Process (TSP)\textsuperscript{20}. However, these are typical 3-to-5-day training courses geared towards industry practitioners, not college courses. Our graduate course includes 20 hours of CMMI lectures and a practical project in which the students perform an assessment, in addition to introductory overviews of PSP and TSP. There are many published experiences on teaching PSP in a college setting. We too offer a separate one-semester graduate course on PSP alone, but its description falls outside the scope of this paper.

3. The context of our course

The Software Quality Management (SQM) course is part of the Master of Science program of the Department of Computer and Information Sciences at the University of Costa Rica. Figure 1 shows the courses of the MS program related with software engineering. This paper describes only the experience in teaching one of them: Software Quality Management.

We designed our SQM course with two main objectives in mind. First, to introduce the student to the current software quality frameworks and tools, and second, to offer a practical component where students learn how to apply the SQM theory in a real-world organization.
One of the first problems we encountered was to choose which software quality management models and standards should be included in a course like this. Figure 2 shows just some of the main models and standards in use today. The most common are the Capability Maturity Model Integration (CMMI), the ISO 9000:2000 quality management standard, and the ISO 15504, along with all their derivatives. The Institute of Electrical and Electronics Engineers (IEEE) alone has 42 software engineering standards, and the International Standards Organization (ISO) has produced at least seven different standards applicable to software.

Centering our course specifically on software quality management, we chose to emphasize three models and standards: CMMI, ISO 9000:2000, and ISO 15504. Other software standards were also included but in less depth.

We have offered the SQM course five times since 2000 to a total of 84 graduate students. All of them work fulltime in various types of industries, from software organizations to IT departments of non-IT companies. Most of them are programmers, software developers, and project managers. We also get people with other job functions such as customer service.

The 4-credit-hour course has 64 hours of lecture time in a 16-week semester. Besides meeting for 4 hour a week, the students have to register for a separate practicum course of 2 credit hours in which they implement a two-part project in a software organization of their choosing. The main
goal of the practicum is to make the students practice a real-world assessment of some of the quality frameworks covered in the course.

4. The course objectives, structure, and content

The main objective of our SQM course is to introduce the main software quality management models and standards in use today, and analyze they suitability for performing process improvement in software organizations.

At the end of the course, we expect our students to be able to:
1. Differentiate the diverse SQM standards and models available today and their scope of applicability.
2. Apply proven SQM standards and models as guidelines to develop software processes.
3. Perform assessments and audits in software organizations using two different SQM standards and models.

Meyer argues that good educational institutions should aim at training people who will perform at the top tier in the profession. Professional certification training programs focus precisely on that. In particular, the body of knowledge (BOK) outlined by the Certified Software Quality Engineer (CSQE) program of the American Society for Quality (ASQ) constitutes an excellent reference point to devise the contents of a SQM course. We decided to select it as our guideline because its scope, focus, and level of knowledge are certainly suitable for our alumnae.

The CSQE certification, offered by the ASQ since 1996, is aimed at software engineers who specialize in software quality assurance. Table 1 outlines the main topics of the CSQE BOK that were included in our SQM course. Bloom’s taxonomy is used to specify the level of knowledge required for each subject. Our focus in this course is specifically on the subject of software quality management. Other CSQE subjects are covered in the other MS courses shown in Figure 1 above.

<table>
<thead>
<tr>
<th>SUBJECT</th>
<th>COVERED IN OUR COURSE?</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. GENERAL, KNOWLEDGE, CONDUCT, and ETHICS (10%)</td>
<td></td>
</tr>
<tr>
<td>A. Quality philosophy and principles</td>
<td>Yes</td>
</tr>
<tr>
<td>1. Benefits of software quality (Comprehension)</td>
<td>Yes</td>
</tr>
<tr>
<td>2. Prevention vs. detection (Comprehension)</td>
<td>Yes</td>
</tr>
<tr>
<td>3. Organizational and process benchmarking (Analysis)</td>
<td>Yes</td>
</tr>
<tr>
<td>B. Standards, specifications, and models. Identify and use software process and assessment models, including ISO 9001, ISO 15504, IEEE software standards, IEEE/EIA 12207, SEI Capability Maturity Model Integrated (CMMI), etc., in a variety of situations. (Application)</td>
<td>Yes</td>
</tr>
<tr>
<td>C. Leadership tools and skills</td>
<td>No</td>
</tr>
<tr>
<td>D. Ethical conduct and professional development</td>
<td>No</td>
</tr>
<tr>
<td>II. SOFTWARE QUALITY MANAGEMENT (18.75%)</td>
<td></td>
</tr>
<tr>
<td>A. Goals and objectives</td>
<td>Yes</td>
</tr>
<tr>
<td>1. Quality goals and objectives (Evaluation)</td>
<td>Yes</td>
</tr>
<tr>
<td>2. Outsourced services (Evaluation)</td>
<td>Yes</td>
</tr>
<tr>
<td>3. Planning (Evaluation)</td>
<td>Yes</td>
</tr>
<tr>
<td>4. Software quality management (SQM) systems documentation (Comprehension)</td>
<td>Yes</td>
</tr>
<tr>
<td>5. Customer requirements (Evaluation)</td>
<td>Yes</td>
</tr>
<tr>
<td>B. Methodologies</td>
<td>Yes</td>
</tr>
<tr>
<td>1. Review, inspection, and testing (Evaluation)</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table 2 outlines the main topics covered in our SQM course and the number of lecture hours spent in each topic.

<table>
<thead>
<tr>
<th>Subject</th>
<th># lecture hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. Introduction to Software Quality Management (SQM)</td>
<td>16</td>
</tr>
<tr>
<td>II. Capability Maturity Model Integration (CMMI)</td>
<td>20</td>
</tr>
<tr>
<td>III. Other SEI models</td>
<td>4</td>
</tr>
<tr>
<td>IV. The ISO 9000 Standard</td>
<td>8</td>
</tr>
<tr>
<td>V. The ISO 15504 Standard</td>
<td>4</td>
</tr>
<tr>
<td>VI. Other standards</td>
<td>12</td>
</tr>
</tbody>
</table>

Grading for the SQM course is as follows:

- Midterm exam: 40%
- Final exam: 40%
- Presentation: 20%
Both written exams consist of 40 multiple-choice questions. The presentation consists of preparing a one-hour lecture on a specific subject assigned by the instructor. There are no additional written assignments.

5. The term project

As Deane et al.\(^1\) points out, nothing teaches better than the experience of doing. Relatively complex tasks can be learned and retained quite well primarily through experience. In our case, learning by doing in a real-world setting is the best way to learn SQM concepts. Therefore, the term project has to be practical in nature and done in a real software organization.

The co-requisite practicum course is a supervised practical application of the theory included in the SQM course. The students register for the SQM course and the practicum simultaneously, but they are separate courses and thus graded independently. Figure 3 shows the time distribution of each topic as well as the relationship of the theory course and the practicum.

All of our students work fulltime in IT organizations. Hence, they can conduct a practical project in their place of work. This is possible because they know the context of their organizations and have access to the necessary information. Furthermore, management allows it because the company gets a process assessment for free.

The term project is part of the practicum and is performed in two parts in groups of 2 or 3 students, depending on class size. Not all of them work necessarily at the same organization. The main objective is to practice the use of two particular quality frameworks by assessing a real software process against them.

![Figure 3. Relationship between the SQM course and the practicum.](image-url)
Grading for the SQM practicum course is as follows:
- CMMI appraisal 50%
- ISO 9000 audit 50%

In the first part of the term project, the students assess their organization’s current software process against the CMMI. We provide them with the necessary guidance and assessments tools to do it, including a CMMI V 1.1 workbook and a ranking scheme to guide the performance of the assessment. Figure 4 shows an example of an assessment performed by one group of students in one organization. The capability profile shown here assigns a level from 0 to 5 to each one of the 25 process areas defined in the CMMI. The project’s scope requires assessing 9 of them. This particular organization has a capability level of 0 in Requirements Management and a 2 in Project Planning and Project Monitoring and Control. The highest ranked process area is Verification with 4, all the rest are in 0, the lowest possible level. This profile clearly shows management the strengths and weaknesses of the current organizational process.

![Figure 4. Example of the capability profile produced by a CMMI assessment](image)

The second part of the project consists of repeating the assessment in the same organization but this time using ISO 9001:2000 as the reference model. They may use the ISO 90003:2004 guideline to help them interpret the standard’s requirements in the context of a software organization. To perform this assessment, the students have to follow the auditing process outlined in ISO 10011-1 for process quality audits, and they have to use the checklist included in ISO 9004:2000 to document their findings.

One of our major concerns for the term project is how to assure the quality of the assessments performed by the students. Performing CMMI process appraisals and ISO 9000 audits require not only extensive training and experience from the assessors, but also to follow a formal assessment method to guarantee the accuracy of the findings. To address this problem, the following measures are taken:

1. We spend half a lecture explaining each of the appraisal and audits methods to be used for the term project, showing real-life examples of their use and emphasizing the importance of their correct application.
2. For the CMMI appraisal, we provide the students with a collection of workbooks that contains easy-to-follow structured forms and instructions for conducting the appraisal tasks and documenting the findings, as well as a straightforward rating system to produce the rankings. The workbooks can be downloaded from www.ecci.ucr.ac.cr/~mjenkins.

3. For the ISO 9000 audit, we provide the students with a standard questionnaire and structured forms for documenting the findings and reporting the results. They can also be downloaded from www.ecci.ucr.ac.cr/~mjenkins.

4. We make sure the students understand from the outset that for the term project we grade not the results of the organization on the appraisal, but the completeness, correctness, and consistency in which the assessment and audit methods introduced in class were implemented.

6. Findings

Table 3 summarizes the results of teaching the SQM course four times in the last six years to a total of 86 graduate students. Since the SQM course and the practicum are separate courses the students get a separate grade for each one. The passing rate of the SQM course is 99% since only one of the 86 students failed, whereas 100% of them passed the practicum.

<table>
<thead>
<tr>
<th>Offering</th>
<th>SQM course (4 credits)</th>
<th>Practicum (2 credits)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># of registered students</td>
<td># of students who passed the course</td>
</tr>
<tr>
<td>2000</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>2001</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>2003</td>
<td>22</td>
<td>21</td>
</tr>
<tr>
<td>2005</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>TOTALS</td>
<td>86</td>
<td>85</td>
</tr>
</tbody>
</table>

At the end of each offering, a 28-question questionnaire using a five-point Likert scale is filled up by the students to assess their level of satisfaction with the course and rate the instructor’s performance. Figure 5 shows the results. We know that 43% of the students rate the course as “very good”, and 55% as “good”, with only 2% of the students rating the course as just “fair”. Additionally, 72% of the students believe the instructor is “very good”, and 26% rate the instructor as “good”. Only 2% of them think the instructor is just “fair”. The same instructor has taught the four course offerings. Finally, 74% of the students “fully agree” and other 24% “agree” that the course objectives were achieved; whereas only 2% “more or less agree” the objectives were met.

Overall, we feel comfortable with the satisfaction level of our students and the realization of the course objectives. But there is always room for improvement. Some of the suggestions we have gotten from the students to improve the course include adding more real life case studies, doing more practice exercises in class, including additional standards in the course material, and changing the written exams (by and large they do not like exams with multiple choice questions).
The main lessons learned from our experience are the following:

1. Limiting the scope and depth of the term project is important. Otherwise the effort required for performing the assessments could be too large for a course project. Each assessment should not required more than 40 person-hours to complete.

2. Having a practicum as co-requisite to our SQM course and students who work fulltime allow us to assign the assessments as the term project. Neither of these two characteristics is present in our undergraduate students, hence when we teach this course at the senior level we are forced to change our strategy.

3. Management sponsorship is necessary for the students to have access to the organizational information required to complete the term project. The assessments require the students to have access to personnel for interviews, written documentation, and sensitive project data that can only be accessible to them by a management directive. For management to grant such access they have to be convinced of the benefits the organization gains from performing the assessments.

4. Appropriate guidance and a set of standard tools such as questionnaires and workbooks have to be provided to the students to assure the accuracy and consistency of the assessments. Otherwise it is easy for them to get confused interpreting specific components of two complex quality models they have little experience working with.

5. For some organizations information privacy has to be guaranteed. In some cases organizations required us to sign a non-disclosure agreement in order to allow the students to do their project. This mean among other things that the assessment results can no be made public or presented to the rest of the class.

6. A course like SQM has to be narrow in its objectives and the topics covered. We believe narrow courses that focus deeply on specific subjects such as SQM are more useful than general courses that browse superficially through a number of of software quality topics without going deep into any of them.
7. Conclusions

We believe that in the last six years our SQM course has fulfilled the objectives we outlined at the beginning. We provide the students with a solid theoretical background on the major software quality frameworks in use today, and at the same time they experience working with a real software organization. This is the main advantage of forcing the students to take the SQM course and the practicum at the same time.

Our experience demonstrates that term projects can be performed in real software organizations as part of a SQM graduate course. This is good for the students because it exposes them to the problems and restrictions of a real company, is good for the companies themselves because they obtain a process assessment for free, and is good for both academia and industry in general because it generates collaboration activities between them.

Some organizations have used the results of the assessments performed by the students as a starting point for launching their software process improvement initiatives. In many cases, it was the first process assessment they had ever performed, making the assessment an innovative project within the organization.

We are currently contemplating offering in near future a software process management course to complement the one described here. In it, the students would learn tools and techniques for process development, implementation, measurement, and improvement, as well as obtain some practical experience performing those activities in a real software organization.

With this course and its companion practicum we have overcome one of the major obstacles of teaching SQM: exposing the students to a real-world experience. In this regard, we believe our experience might be helpful in improving the design and implementation of graduate courses in software engineering.

Finally, our case study is limited to four course offerings totaling 86 graduate students in a six-year span. We hope in the near future to offer this graduate course several more times and add that experience to the one described here.

Bibliography


