

**Teaching Software Quality and Leadership:
Experiences and Successes**

Judy Bamberger and James Hook

Oregon Graduate Institute
Department of Computer Science and Engineering
20000 NW Walker Rd
Beaverton, OR 97006-1999 USA

Technical Report No. CSE 97-006
September 12, 1997

Teaching Software Quality and Leadership: Experiences and Successes

Judy Bamberger
Process Solutions
Adjunct Professor,
Oregon Graduate Institute
bamberg@eaglet.rain.com
+1-503-690-1206
+1-503-690-1548 (fax)
12440 NW Haskell Court
Portland OR 97229 USA

James Hook
Oregon Graduate Institute of
Science & Technology
hook@cse.ogi.edu
+1-503-690-1169
+1-503-690-1548 (fax)
Oregon Graduate Institute
PO Box 91000
Portland OR 97291-1000 USA
<http://www.cse.ogi.edu/~hook/>

Keywords

Education, Software Quality, Software Process, Software Process Improvement, Process Definition, Formal Inspection, Leadership

Abstract

Is it possible to teach software quality and leadership concepts and skills at the graduate level? Can this be done simultaneously within the nurturing environment of the classroom and the risky world of industry? Is it possible to provide students with enough skills and techniques to demonstrate immediate results and, at the same time, provide them with enough background and concepts that they become intelligent consumers and decision makers when it comes to software quality issues?

The Software Process Practicum: Lessons in Software Quality and Leadership, taught at the Oregon Graduate Institute fall term 1994 - 1996, provides a clear demonstration that the answer is an emphatic, "YES!" The Software Process Practicum won the 1996 Software Quality Excellence Award offered by the Pacific Northwest Software Quality Conference, confirming that it is not only possible, but that it can be done well, and that a university course can provide significant opportunities to its students to make a difference in the professional community.

This paper covers the following areas:

- The background of the Software Process Practicum and the premises on which it is designed
- The overall Practicum framework and modules taught
- An overview of the Software Skills and Competency Model on which the Practicum is based
- The impact the course has had on some of our students and the results that they, and the organizations for which they work, are seeing
- Future directions

1. Background of the Software Process Practicum

In 1993, the Oregon Economic Development Department funded an effort to:

... create a Software Engineering Model that characterizes performance excellence (current state and future state), create the capability to assess software engineers against that model, and provide a means to assess individual and organizational progress toward excellence on an ongoing basis.¹

The organizations participating in this effort were the four leading software/technology companies in the state of Oregon: Intel, Mentor Graphics, Sequent Computer Systems, and Tektronix.

In January 1994, this group made its first public presentation. At that meeting, Judy Bamberger (a report author from Sequent) and James Hook (an educator from OGI) met each other, began discussing the model, and soon found out they had many interests in common, including taking action to address the ideas in the model.

In April 1994, we² committed to teach a new course, which was to become the Software Process Practicum: Lessons in Software Quality and Leadership. Moreover, we agreed to leverage what we could from the *Report of Findings* to guide the structure and content of the course.

2. Basic Premises behind the Software Process Practicum

We agreed upon a set of guidelines for our selection of topics, readings, in-class exercises, and homework assignments. These were:

- Exploit the "adult learning paradigm" and make the learning experience and the materials relevant.

Recognizing that adults bring a multitude of ideas, job worries, motivations, and rich experiences with them when they come to class, we decided to leverage that background both in the classroom and in homework assignments. We also used a variety of teaching methods (lecture, discussion, hands-on labs and problem solving activities, student presentations; visual, aural, tactile; etc) to maintain interest.

- Ensure the topics are applied and reinforce each other.
We used in-class labs to practice new skills. Many homework assignments required the students to take those skills and apply them in a real-world setting with a partner organization. We took time in class to discuss and apply readings and ideas. In general, our approach was: teach, do, and discuss.
- Teach the "soft" stuff, the social processes, as well as the "hard" stuff, the quality processes; and emphasize the inherent interdependency of the two.

We leveraged the fact that there already are plenty of courses and materials that concentrate on basic quality tools, techniques, and frameworks. The value that we added

¹From the *Report of Findings: Joint Software Engineering Needs Analysis*, 18 February 1994, available from Oregon Economic Development Department, Salem OR.

²From this point on, "we," "us," and "our" refer to the authors.

was to integrate and apply many of these existing concepts and techniques to software engineering.

- Balance education (concepts, long-term focus) and training (skills, short-term focus) needs. This was, for each of us, our most interesting challenge, and a key risk. The challenge stemmed from the fact that each of us represented a community with a different primary driver for consumption of education: academia having a more research, theoretical, long-term tradition; industry have a more practical, skill-oriented, short-term tradition. One component of the resulting risk was how we were going to bridge that gap ourselves, as we collaborated in developing and delivering the course. Another component was how to build that bridge with our students, who we anticipated would be representing both academic and industrial communities. And a third component of the risk involved the selection of which topics to teach, and how to present them.

We put this to the test as we built the framework of the Practicum and populated it with specific readings, lectures, discussions, and homework assignments. We decided to teach multiple techniques, methods, and frameworks wherever possible, and to provide students with the ability to make decisions about what is appropriate under which circumstances. When it came time to practice (i.e., homework), the students were free to choose the specific techniques or concepts to apply to address the scenario we provided for them.

- Build on a variety of materials.

We distributed many of the seminal papers (such as Fagan on inspections, Radice on software process definition), outrageous editorials (such as the Bach/Curtis debate on the Capability Maturity Model), classic writings (such as Crosby, Deming, and Juran on quality), books taking a "system" view of software quality issues (such as Weinberg), current findings and issues, and even cartoons. We required the students to be exposed to some real-world practices in the industry (such as interviewing practitioners and working with a "partner organization" on a multi-part project) and to leverage experiences from their own work environment.

- Use information from the *Report of Findings* to select or validate what and how we are teaching.

This is discussed briefly later in this paper, and is presented in detail in an OGI technical report.³

- Ensure the students have fun while learning, and that we, too, both have fun and learn. We recognized that *software quality* may not be viewed as a "glamorous" topic by most people, even though it is increasingly recognized as being important. We also knew that to create an effective learning environment for students and instructors who had already put in an eight-hour day would be a challenge. We faced that challenge and, within the scope of our basic premises, consciously added some irreverent, off-the-wall items.

3. Framework of the Software Process Practicum

³Technical Report number CS/E 95-006, *Software Process Practicum: Lessons in Software Quality and Leadership*; March 1995, Oregon Graduate Institute of Science & Technology, Portland OR USA.

The Practicum resulting from this partnership shows its heritage: it is a synergistic mixture of tools, techniques, and practices important to practitioners in industry (as represented by Judy) and concepts, models, and data important to more theoretically-oriented academics (as represented by Jim). It is structured to meet both short-term training and long-term education needs. The resulting Software Process Practicum is a unique and highly effective course covering both the "hard stuff" of quality processes, the "soft stuff" of social processes, and specific techniques that require effective knowledge of both. Figure 1 illustrates this framework.

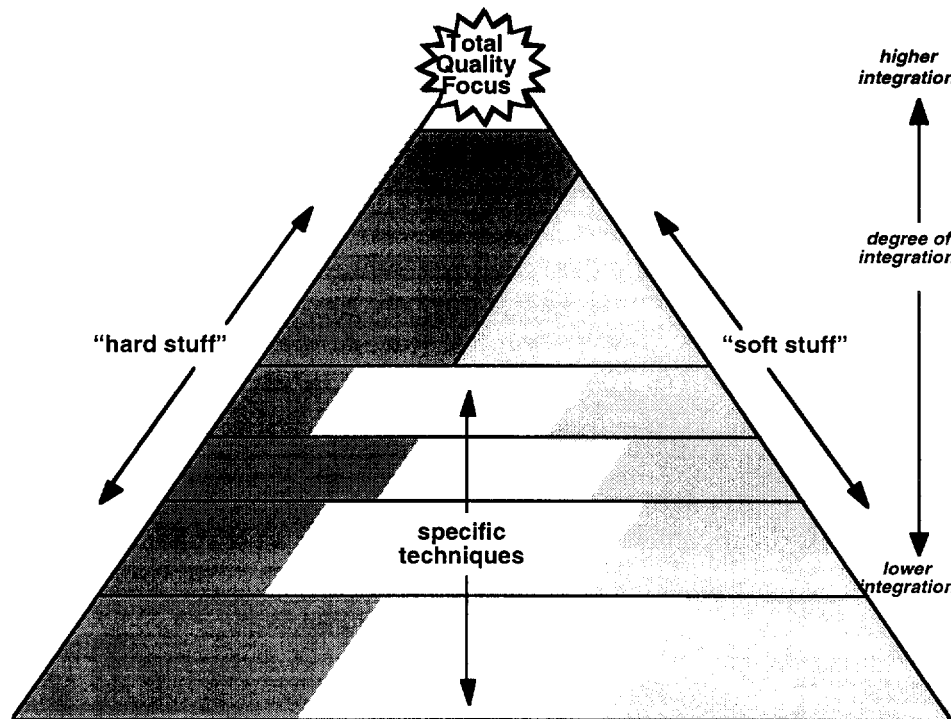


Figure 1. Framework for Software Process Practicum.

- The "hard stuff" represents *quality processes*: basic models, tools, and frameworks related to effective software process. Most of this is provided initially via readings and class lectures.
- The "soft stuff" represents *social processes*: the individual and team social concepts, models, and skills required to apply the hard stuff successfully. Most of this is provided initially via readings and class lectures.
- Specific techniques represent the *integration of the quality processes and frameworks with the social processes* and team skills in an applied setting.

Most of the specific techniques are introduced via readings and class lectures and reinforced and practiced via hands-on labs during the class, a Saturday workshop, and individual and team homework assignments. Moreover, team homework assignments focus heavily on the integration of the quality and social processes in an applied, real-world environment. Two homework assignments require the students to work in teams with a partner organization for an extended period.

Figure 1 also demonstrates our assertion that the overall goal of the course was to provide the students with a total quality focus, the integration of all the parts into an effective and usable whole. This is not to be mistaken to be TQM - Total Quality Management. The Practicum does not advocate any single way of going about achieving "quality"; rather, it focuses on providing the concepts, knowledge, and skills from which the students may select - as critical consumers - to apply as they determine appropriate to each context.

The course objectives we defined stated that, after this class, the students will understand and have demonstrated that:

- Software processes can be managed and controlled.
- Software engineering is a social process, too.
- The skills learned in the Practicum can be applied today at work and can be used effectively.
- The concepts, knowledge, and skills provided in the Practicum can be used to make well-informed decisions about applying software quality principles and tools to personal, project, and corporate software activities.

Given these objectives, we looked at the possible topics to teach. We recognized that there was a "hierarchy" of models, concepts, and skills: beginning with the simple and self-contained, and continuing to the more complex and more highly integrated.

When we started to populate this framework with actual topics, we found we had identified many more topics than there was time to teach in 30 contact hours (this was expanded to 40 contact hours in 1995 and 1996). To determine which topics we would teach in the first offering of the Practicum (fall 1994 term), we used two primary sources for guidance: the competency model defined in the *Report of Findings*, and our knowledge of the local software engineering industry from which our students would come. The resulting framework populated with the topics we taught in fall 1994 is shown in Figure 2.

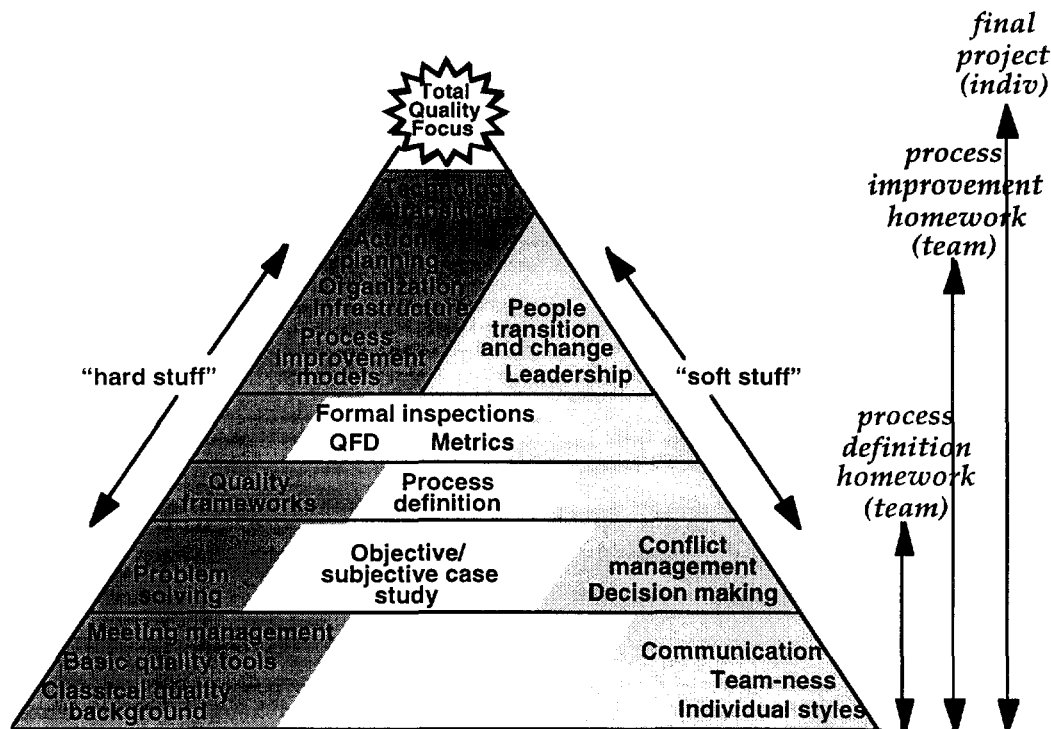


Figure 2. Topics Taught in Fall 1994 Software Process Practicum.

These same topics are being taught today, as the students continue to tell us they are relevant and useful to them.

4. Overview of the Software Skills and Competency Model

The *Report of Findings* contains a matrix of technical skills and competencies. These skills and competencies were identified via a thorough and structured information gathering process.⁴

The four technical skill areas (requirements) that were identified are:

- Process and Management: software process, software review techniques, project management, business literacy, configuration management
- Design: object-oriented design/programming, data structure design, design methods/simplicity, formal analysis techniques, client-server architecture, human interfaces/graphical interfaces
- Implementation: software optimization, debugging techniques, understanding foreign code, technical writing, software testing techniques

⁴This structured information gathering process is based on a job competency analysis developed by David McClelland (*The Achieving Society*; 1961; Princeton, NJ; Van Nostrand), a critical incident interviewing technique developed by J C Flannigan (*Critical Incident Interviews*, Psychological Bulletin, 51; pages 327 - 358; 1954), and an extension of this technique into critical behavior interviewing (Cambria Consulting, Inc; Boston MA; 1993). It is described in detail in the *Report of Findings*.

- Specific Technical Skills: windows application programming, CAE related technical areas, communications and networking, natural data types, device drivers, real-time systems

The identified competencies were organized into five groups:

- Concern for the Process (information gathering, efficiency, systematic thinking, discipline/rigor)
- Concern for the Team (collaboration, team building, technical leadership)
- Concern for Ideas (communication skills, influence/persuasion)
- Concern for the Company (risk management, results orientation, user orientation)
- Concern for the Solution (persistence, creativity, learning by doing, pattern matching, initiative)

The power of the software engineering competency model is demonstrated via the intersection of the technical skills and competencies. The Software Process Practicum satisfies the technical skills area of Process and Management very highly and three of the competency areas – Concern for the Process, Team, and Solution – very highly. These results are summarized in Figure 3.

The process and management technical skill area was the only one relevant to our course.⁵ While all the competency areas were somewhat relevant, we needed to restrict our scope. As a result, we chose to focus very heavily on the Concerns for Process, Team, and Solution.

Each competency area summarizes several specific competencies, each of which in turn summarizes multiple specific and observable behaviors identified via the in-depth interviewing and questionnaire steps described in the report.⁶

⁵OGI has issued a document mapping its curriculum to the Competency Model. This document is available from Oregon Graduate Institute, Computer Science and Engineering Department, Portland OR USA.

⁶OGI Technical Report number CS/E 95-006 provides a detailed mapping of each course module to each of the 17 competencies. The mapping indicates whether this was accomplished via readings, class lecture, class activities or labs, or individual or team homework assignments. Also, for each of the 17 competencies, Technical Report describes: what the class sees (specific topics covered via reading, lecture), how the students use it (reinforcement techniques used to ensure key points are mastered, skills built, and general awareness achieved), and how the instructors do it (application of these competencies to us as we built and taught the Practicum).

Technical Skills				
	<i>Process and Managem nt</i>	<i>Design</i>	<i>Implemen- tation</i>	<i>Specific Technical Skills</i>
Software Process Practicu m		None	None	None

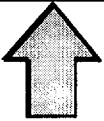
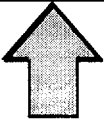
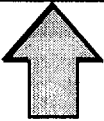
Competency Areas					
	<i>Concern for the Process</i>	<i>Concern for the Team</i>	<i>Concern for Ideas</i>	<i>Concern for the Company</i>	<i>Concern for the Solution</i>
Software Process Practicu m					

Figure 3. Mapping of Software Process Practicum to the Skills/Competency Matrix

5. Impact: Student and Instructor Results

The students who have taken the Software Process Practicum have used many of the concepts and techniques we taught in the classroom in their work environments. They are continuing to demonstrate objective evidence of improvement on personal and organizational processes, which is visible to many of their peers, managers, and Vice Presidents. They tell us the products are being built and released on-time and with high quality.

5.1. Student-Based Results

Nearly all of our students cited meeting management as the most important single topic we taught, and one that was immediately applicable in their work environments. One student was holding weekly meetings with his project team to share the Practicum information through the entire term.

Many students also cited the process definition representations and techniques as something they are using already at work. One student shared with us an observation from the process definition and improvement homework assignments done with her partner organization. The organization's staff stated that they finally saw the relevance of and tangible results from many quality techniques in which they had been trained, but had not used to that point. Another student used the information we provided to help revise many of his company's procedures in preparation for ISO 9000 registration.

Two students shared with us that their industry managers told them that the topics and methods being taught in our course are of significant importance to them. Two other students have used their newly discovered organizational skills and have been promoted to team leaders.

Not all results have been limited to the work environment. One student described how he gently applied some of the basic meeting management and interpersonal-oriented concepts we taught to a jury on which he was serving. Other students mentioned using some of the social style profiling models with their families and friends. In all cases, the students reported improvements in understanding of and insight into others.

Many students have been reaping the benefits of the concepts behind formal inspections by increasing their personal discipline and how they produce their software work products. One student used the information from the Practicum to update and refine his team's on-going inspection process, and to provide additional training in the improved process.

Two students are trying to introduce formal inspections into their workplaces. For example,

A project at ADC Kentrox scheduled a three week period specifically for code inspections, and they are already benefiting from the results of preparing for and conducting three code review sessions. While not as formally conducted as were the sessions taught in class which used uniform codes for defect categorizations and which also focused on accumulating defect metrics, the time spent has widened the scope of understanding of the software by each team member and has resulted in positive, identifiable improvements to the software.⁷

Another student, in her final project, prepared a three-year plan for the introduction of formal inspections across her organization.⁸ We received many positive comments about the exposure to the wide variety of quality frameworks we used, for example:

... exposure to the Trillium Model (which integrates and adds to aspects of SEI's CMM [Software Engineering Institute's Capability Maturity Model], ISO 9000, Malcolm Baldrige, and IEEE concepts, and targets these concepts to the Telecommunications Software industry), has led to intense interest in making use of the model for identifying specific areas for potential improvement at ADC Kentrox.⁹

One of the student process definition and improvement projects has been presented to two Vice Presidents. The instrumentation (metrics) requirements identified by this team of students will be included in the partner organization's metrics. These metrics will help the partner organization measure how well it is meeting its specific goals in the area of time commitments to customers. The proposed metrics are defined precisely and tied to specific process steps. Other process improvements were also proposed and will be adopted. These include better control of certain software products, and taking five, piece-meal, information-gathering process steps and merging them into a single step done once at the beginning.

⁷Personal correspondence from Gary Hanson, February 1995.

⁸This plan was not fully executed. Unforeseen, and real-world, events such as reorganizations and job changes have kept this from happening. However, the student tells us she is using the discipline she learned in many other areas, and this has been having a positive effect overall.

⁹Personal correspondence from Gary Hanson, February 1995.

Another student (full-time graduate student with no prior industry experience) who recently took a job with Motorola, which has a strong reputation for its emphasis on process and quality, let us know that the Practicum had set him up nicely. When he was going through the "quality school" at Motorola, he was way ahead of the pack, as he had already seen and used many of the techniques and skills introduced in the class.

One student was assigned leadership of a team trying to get a new product developed and delivered on an extremely tight schedule. He described how he is leveraging many of the interpersonal skills and quality tools he learned in the Practicum, and is pleased to report that the project was delivered on-time, within budget, with all of the functionality expected by the customers, and well within the desired quality. In fact, we were told that the reviews this product received when it was introduced were significantly more favorable than at any other product roll-out before, and those were already positive.

One of the intangible benefits to the students - or a benefit that we observe - is how they grow and develop as people throughout the term, and beyond. Two students in particular, who could have easily been "hidden" in project activities, have become peer leaders - a characteristic that has been noticed favorably by one of the team's customers.

5.2. *Instructor-Based Results*

We asked for, and received, feedback throughout the course. In fact, on the first day of the course, we asked the students, "What are your expectations from this course? What do you expect to get out of this?" As most of our students had never been in a classroom where someone asked them, the customers, about their needs and wants, they were somewhat surprised. Also, from the outset, we indicated that we were going to ask for feedback at specific points, and that we welcomed feedback and suggestions for improvement at any point. We asked, and the students responded.

From the 1994 term, the most significant piece of information we received was "too much, too fast!" When we heard this, we dropped many of the individual homework assignments, and pruned the readings by setting some aside as optional. To address this issue in the longer term, we increased the credit hours earned for this course from three to four, and we hold the class for two two-hour sessions each week. We did not add more material; we slowed down and spent more time making explicit ties with the readings and individual homework assignments. This, in fact, addressed another issue that was raised.

The independently administered end-of-term course evaluations yielded no surprises. We received very strong ratings in the following areas overall in all three years:

- Instructors knowledgeable
- Lectures well prepared, organized; presentations made for easy note-taking; topics appropriate
- Good communication skills
- Concerned about students; opportunity to ask questions; students treated fairly
- Course material challenging

- Increased my interest
- Built/augmented student competency

In 1994, we received lower ratings in the following areas:

- Related course concepts in systematic manner
- Textbook and supplementary materials valuable
- Pace of the course (it was too fast)

To address these, we included a version of Figure 2 in each lecture, and we systematically used it as a roadmap to explain the relationship of the topics as they were presented. We increased credit hours and other activities described above, and we took more time to integrate the results of the readings into class time. As a result, the ratings for the first two of these areas showed marked improvement. We still are grappling with the issue of pacing.

In 1996, we introduced two mechanisms to get additional in-process feedback from the students - timelogs to get quantitative information and a journal, to get qualitative insight. Both were optional; neither was marked in any way. We regularly received timelogs from about 20% of the students; and journals from about 15% of the students.

We asked the students to maintain and submit a timelog each week, reporting the time spent on readings, homework, projects, and other activities. We recorded, graphed, and posted these data each week. This provided us with a lot of information on the pace of the class and the time each student was spending outside of class, which we were able to discuss with the students and modify the reading assignments accordingly. And, for the first time (for the 1997 class), we will be able to show incoming students one view of the expected course load, and where the spikes occur.

We asked the students to maintain a personal journal of their learnings, which we collected and reviewed three times during the term. We suggested each student identify a set of learning objectives as part of the first entry, and then reflect on those throughout the entire term. This type of feedback was wonderful - we found students teaching *us* in many ways - writing about links between the course material and their real world situations, and seeing connections that we had not noticed.

Other comments (not from any journals) we received included:

- "Keep it as a pair of instructors." coupled with "Jim provided a good 'sounding board' and 'plant' for Judy's lectures. Jim's PacSoft work was good background."
- "As [a] professor, [Jim] gives a strong testimonial for the material's importance."
- "More on metrics." (We revised the session on metrics; and, by instituting the timelogs and using them, were able to demonstrate one simple and powerful use of metrics.)
- "Course content very important for OGI and community. Get more industry involvement."
- "[Aspects you would like to stay the same] Team projects. Software processes. Subject matter. Instructors excellent - retain and reward. Saturday labs very good."

- "I liked the 'practice what you preach' mentality ... "
- "The team exercises were exceedingly valuable. It gave us an opportunity to try out techniques in a non-threatening team before applying them in our own work environments."

6. In Retrospect: Revisiting the Basic Premises behind the Software Process Practicum

So how well did our basic premises serve us? How do they still affect us after three offerings of the Practicum and as we prepare for our fourth?

- Use the "adult learning paradigm" to make the learning experience and the materials relevant.

In retrospect: This continues to prove to be one of the most important decisions we made. By providing a combination of the traditional "lecture/homework" and experiential learning environments, we have been able to ensure the key lessons "stick." We have provided the students with a safe environment to practice new knowledge and skills - this is the class environment. And our students then tell us that they are able to use their new-found knowledge in the workplace with increased confidence. Another way of reinforcing this was to invite our 1994 students to the last class of the 1995 term, a practice we have continued each year. There, they shared their key learnings and described their own "real-world" experiences of how the information presented in Practicum helped them to achieve personal, team, project, and organizational goals.

- Ensure the topics are applied and reinforce each other.

In retrospect: There is an oft-cited guideline in marketing - it takes five to six contacts for a new message to be adopted. We recognize and exploit this in the Practicum. We introduce some basic concepts and skills early (e.g., meeting management, social styles, effective teaming) and discuss them in many different contexts throughout class, linking and reinforcing their importance. We also ask the students to use them and write about them in several homework assignments through out the term. By providing a safe place for the students to practice and many opportunities to refine their use of the concepts we teach, we contribute to student success in applying this information in the "real world."

- Teach the "soft" stuff, the social processes, as well as the "hard" stuff, the quality processes: emphasize the inherent interdependency of the two.

In retrospect: This has proven to be the biggest strength of the Practicum, the area the students most often cite as leading to their most significant growth. Most of our students are well versed in technical concepts, models, knowledge, and skills. They are most adept in the Innovator role of leadership.¹⁰ By focusing on the social processes as well, the Practicum introduces the students to many of the key skills needed by the Motivator role (and, to a lesser degree, the Organizer role) and provides the opportunity to practice it.

¹⁰See *Becoming a Technical Leader*, Gerald M Weinberg, Dorset House Publishing, for a discussion of the MOI leadership model - Motivator, Organizer, Innovator.

And these are the skills that industry representatives continue to tell us are typically lacking in students from most traditional software engineering courses and curricula.

- Balance education (concepts, long-term focus) and training (skills, short-term focus) needs.

In retrospect: We discussed this throughout each offering of the Practicum, and again as we do a post mortem from the previous offering and when we plan for the next offering. In fact, we find these discussions cover much more than just education vs training; they cover many of the issues crucial to building a successful academic-industrial collaboration. We recognized this early, and we remind ourselves of it continuously. We are both strongly committed to supporting the students, the department faculty, and OGI goals of meeting the needs of the industry in the local community; we have been able to build an effective partnership to meet those goals.

- Build on a variety of materials.

In retrospect: This is a continuing challenge - how to cover all the "interesting" material without overloading the students?! The other challenge is a truly administrative one - getting clean originals, getting the copyright permissions, and getting copies into the students' hands, and so forth. We can (and do) plan ahead for the "classical" writings. The more current and "breaking" information causes us more difficulty, both on an on-going basis during the term and at the beginning of each term as we put the "course pack" (readings) together. While preparing for the 1996 fall term offering of Practicum, we invested substantial time and effort into collecting and reproducing all the readings ahead of time; this minimized the overall effort in 1997, when we were able to "swap out" some of the readings for more current information.

- Use information from the *Report of Findings* to select or validate what and how we are teaching.

In retrospect: When the *Report of Findings* was published in 1994, we were not surprised at the skills and competencies that were identified as leading to success in industry. Even though some of the specifics have changed over the past three years, the basics remain constant. Because we built the Practicum around the more basic, longer-lasting, conceptual items (e.g., process, teamwork, learn-by-doing), the Practicum is still meeting the needs of the industry community we are serving. We hear this from the students; we also hear this from the companies for which our students work.

- Ensure the students have fun while learning, and that we, too, both have fun and learn.

In retrospect: We keep looking for humorous items to integrate in our class. We have found additional ways as well, things that some years ago we would probably never have considered doing - such as having "toys"¹¹ available to the students during many class

¹¹For example, koosh balls, yo-yos, pocket Etch-A-Sketches, and the like. There is a theory that espouses multiple dimensions are involved as people learn. The toys serve the kinesthetic dimension. We find our students may grab a koosh ball and "juggle" it, while remaining fully involved in the class activities. And the koosh balls, when lobbed gently, serve as a "friendly" reminder to a student or instructor that s/he is drilling a subject to death - "rat-holing" (we often have a class ground rule that covers this use).

sessions. As we have set the tone for allowing fun in class, our students contribute in unique ways. They sometimes bring in relevant comic strips or "real life humor" from their work. One student, in the fall 1995 term, brought in an exercise to define a process to make a peanut butter and jelly sandwich. Initially, we wove this into our last session on process definition. In the fall 1996 term, it became a regular part of our class.¹²

7. Future Directions

We have been holding a periodic "get together" for more than a year, open to all those who have completed the Practicum. We meet at someone's house for dinner or just afterwards to discuss a topic, work on a problem, hear a visitor from out of town sharing some expertise related to Practicum topics, and more. Subjects that have been addressed include leadership, communication, change, and resistance to change. The meetings are now "self tending"; with groundrules and common vocabulary from the Practicum itself, we announce our meetings, manage our time, coordinate our sessions, and plan for the next time together. We are building a small and viable network of software quality leaders.

While we do not have much data to report yet, we find the former students still talking about the value of and using much of the information they learned in the Practicum. In this next series of sessions, we will be exploring the personal software process.

In any case, we will continue to host periodic follow-up gatherings. We have several purposes for this:

- We will obtain an "N-months after" perspective from our past students. This will be very useful, as we will factor that information into our preparation of course modules for the next offering.
- We will try to understand progress to-date in some of the longer-term improvement efforts. This will feed into our course material and presentation improvement process (above) and enable the students to leverage the experiences of each other (below).
- We will provide an opportunity for the students to continue learning and supporting each other, growing a network of quality and process knowledgeable resources within the local software community. This will build on one of the key premises behind the Practicum - that adults bring much to the learning experience, and can learn from each other.

We are teaching this course again in the fall 1997 term. That gives us sufficient time to execute, once again, our own process improvement cycle. And we continue to be in touch with local industry groups (e.g., Software Association of Oregon) to describe this educational opportunity.

¹²Each team has about seven minutes to define a process to make a peanut butter and jelly (PB&J) sandwich using a well-defined set of supplies. Following this, that process definition is passed to another team, which has about eight minutes to follow that process and make the sandwich. There are many variations on the theme - allowing a process improvement round, allowing training and user consultation - the possibilities are endless. It is quite interesting if the person who executes the process has never heard of a PB&J sandwich! Numerous discussions abound afterwards - about the principles of process definition, about teamwork and communication, and more. Try it and see!

Many of the improvement efforts that our students began are continuing - albeit slower than anticipated; some are completing, with much success. The students are making progress; they know we are available and able to assist them; and they have contacted us for assistance and have used our materials. We believe strongly that it is not the classroom performance that will prove the benefit of our course, but rather how that information is taken back into the workplace and applied successfully. Our continuing follow-up will provide us - and the students - with more information.

In the meantime, we have been pursuing additional liaisons with local industry. Through the Oregon Graduate Institute Computer Science and Engineering Department and the Continuing Education organization, we initially targeted the four companies that participated in creating the software competency model - Intel, Mentor Graphics, Sequent, and Tektronix. Our goal is to improve collaboration between industry and academia. Industry has stated its needs; it is now academia's opportunity to put a program together - like we did for the Software Process Practicum. And then it is back to industry to provide the opportunities and encouragement to its employees to participate in the educational opportunities that have been created.

The OGI Continuing Education organization offers a number of short courses to organizations of all sizes, often at their site. We are exploring several alternatives to making all or portions of the Software Process Practicum available, on-demand, to local industry. This activity clearly supports the goal of building an effective, highly productive, extremely competitive workforce in this area. Many of these short courses are also being presented as "workshops" at national and international conferences.

Our conclusions are straightforward:

- We have demonstrated that it is possible to have a working, effective, collaborative partnership between industry and academia wherein everybody wins - especially the students.
- We have demonstrated that via this collaboration, academia (the Oregon Graduate Institute) can respond to industry needs effectively, once those needs are articulated (as they are in the *Report of Findings: Joint Software Engineering Needs Analysis*).
- Our students tell us that they are using what we taught them, that it is working for them, and that they are making improvements and changes in their personal processes, their projects, their organizations.
- We have demonstrated that students can be taught software quality concepts, knowledge, and skills in a way that enables them to begin to use it immediately and to demonstrate objective and quantitative results.
- We believe as strongly as ever that the "soft" stuff is just as important as the "hard" stuff. In fact, it is the interpersonal and team techniques and concepts that underlie and enable effective process definition and improvement techniques.
- Learning by doing - again - and again - and again - is a very powerful teaching technique. The academic environment provides a "safe haven" to experiment with new concepts and

methods. By tying our learning-by-doing assignments to real-world issues, we have provided a safe and relevant environment that fosters effective learning.

8. Some Closing Thoughts

The Software Process Practicum won the 1996 Pacific Northwest Software Quality Conference's award for Software Quality Excellence. This process was ardently supported by the Computer Science Department Head at the Oregon Graduate Institute and by our students. We continue to learn from our students many exciting ways they have used the materials and opportunities provided in the Practicum to "make a difference." Specifically, we heard how the students have truly changed some of their own professional lives and how they have had a positive impact on the quality and timeliness of the products they are involved in producing.

What we did "isn't rocket science"; we have demonstrated that process improvement concepts and practices can be taught to a mixed group of students in a for-credit, graduate class. Thus, the software engineers of the next century are being exposed to state-of-the-art and state-of-the-practice process improvement literature, tools, and methods. In addition, by bringing such a course into the academic setting, into a class of students both from academia and industry, by working hands-on projects with both academia and industry, we are definitely broadening the perspective of where process improvement is applicable.

9. Acknowledgements

We appreciate the efforts of the State of Oregon and the team from Intel, Mentor Graphics, Sequent, and Tektronix for their efforts in creating an outstanding model that proved very useful to us. Thank you to the support staff of OGI who went beyond the call of duty to support this initial teaching of the Software Process Practicum, notably Phyllis Raymore and Kerri Burke (in 1994) and Kelly Atkinson (in 1995), Jeff West (in 1995 and 1996), and Barb Mosher (1996). Most important thanks go to our students, who took the material we offered and made brilliant use of it in their own work environments.

From 1994: Jim Bindas (Intel), Debbie Blanchard (Consolidated Freightways, now with Con-Way Transportation Services), Michael P Gerlek (OGI, now with Intel), Gary Hanson (ADC Kentrox), Allyn Jackson (ADP Dealer Services, now with NEC), Celeste Johnson (Mentor Graphics), Alexei Kotov (OGI), Christos Mandalides (OGI), Thom Parker (Intel, now back in school), and William Trost (OGI, now an independent consultant)

From 1995: Tito Autrey (OGI), Jef Bell (OGI, now with ABC Technologies), Kathy Fieldstad (now with Revision Labs Inc), Priyadarshan Kolte (now with Motorola, Austin), Eldon Metz (now with Rogue Wave), Joe Mueller (Mentor Graphics), Sudarshan (Sun) Murthy (Tiger Systems Inc), Ian Savage (CFI ProServices, now with Tiger Systems Inc), Ingrid Sutton (now with Intel), Jim Teisher (Credence), Doug Vorwaller (Wacker Siltronic Co), Tanya Widen (OGI), Chad Willwerth (Intel)

From 1996: Chris Bachmann (Coe Manufacturing), David Butt (Medtronics), Laurie Duff (ADP), Pedro Fajardo (Intel), Bill Magee (FLIR Systems), Shary McIntee (FEI Company), Brain McLaughlin (Orcad), Jamshid Mehr (Intel), Steve Rosenberg (FEI Company), Susan Strayer (Sulzer Bingham), Bruce Thom (Dynamics Research Corp), Don Wakefield (Mentor Graphics)

Our course, and this paper, reflects many of their suggestions for improvement. Our appreciation also to those partner companies who provided a real-world laboratory for our students, and who benefited from the results.

Appendix A. Course Logistics

We are often asked for a description of the course logistics; this Appendix provides a summary of that information.

Ten students enrolled in the fall 1994 offering of the Software Process Practicum, an acceptable enrollment for a first offering of a course like this, based on past history; there were 13 in fall 1995; and 12 in 1996. Two of the 1995 students came as direct referrals by a 1994 student. Much to our surprise, this was from a research lab at OGI as opposed to being from industry. One 1996 student came based on the interviews for the Software Quality Excellence Award (she was a committee member), and two came based on the subject matter and having one of us (Judy) as an instructor in OGI's Introduction to Software Engineering course.

In both 1994 and 1995 offerings, the industry students were from various sized companies, and had a variety of job responsibilities, including software engineering, customer support, process, and training. Of the 1994 students, very few students had any prior software engineering education beyond unstructured "on the job" training. On the other hand, of the 1995 students, a larger percent had received some software engineering education in school and/or at their companies. The 1996 students were different: all were from industry (i.e., no OGI students); all were involved in software development or test in some way.

<i>Attribute</i>	<i>1994</i>	<i>1995</i>	<i>1996</i>
Total Enrollment	10	13	12
Full-Time OGI Students	4	3	0
Full-Time Employed (vs student)	6	10	12
Pursuing Masters or Doctorate Degree	6	4	6(?)

In 1994, the class was scheduled to meet twice each week for one and one-half hours, 5:30pm - 7:00pm; several classes ran over by up to 15 minutes. The class met for 10 weeks, with one all-day, Saturday lab. Class time was mostly lecture, along with significant time for discussion and hands-on classroom activities. This same basic structure was retained in 1995; however, in response to student suggestions and a reality check on our part, we increased the number of credits from three (in 1994) to four (in 1995 and 1996), and ran class times from 5:30pm - 7:30pm (5:00pm - 7:00pm in 1996), with almost no classes running over.

Readings were assigned each night; individual homework assignments were assigned intermittently; there were two major team homework assignments. As soon as the students indicated the load was just too much, we reduced the amount of reading and eliminated many of the remaining individual homework assignments. We faced this issue both years, even though we reduced the required readings for the fall 1995 offering. This was less of a problem in 1996.

In the fall 1995 term, we also began asking the students to keep metrics on the time spent on out-of-class activities. As this was our first time collecting these numbers in a defined way, we did not use them very actively in 1995. However, we got enough information to validate that, on the average, many students were within the bounds of a four-credit, graduate course at OGI (about two hours of outside work for each hour of contact), with notable peaks and valleys. This is one area we addressed more proactively in the fall 1996 term. We asked the students to turn in a timelog on a weekly basis. We built a simple spreadsheet to collect, summarize, and graph the results. As a result, we now have some quantitative data we can begin showing to prospective students. We did not, however, get sufficient data points (i.e., have enough students submit timelogs) on a regular basis - only about 20% of the students submitted timelogs - and these data will state that when they are shared.

The two centerpiece homework assignments were to be done in teams and with the participation of a partner organization. These were to work with a partner organization to define one of their real-world process (mid-term homework) and to work with them to create a process improvement plan for that process (end-term homework). Both these assignments required significant effort on the part of the students - including identification and enlistment of the partner organization, interviewing and working with its staff, and providing information back to various levels of the organization. In addition, the students used meeting management, interpersonal, communication, and team skills, and they gave presentations of these two major homework assignments. Using the 1996 metrics, we saw a noticeable spike in time spent as the due dates for these assignments approached. We hope to use these data to encourage future students to plan ahead.

As it happened, the teams basically stayed constant throughout each offering; one of the 1995 teams lost some members and had to address those team issues as well. For some of the in-class activities, we specifically selected other team arrangements to ensure the students had an opportunity to work with a variety of people.

The Saturday lab was primarily a hands-on formal inspection workshop, concluding with the presentation of the process definition projects. The students presented their process improvement projects at the penultimate class session. The final session was a review and synthesis session in preparation for the final project. ¹³

The students had a final project to complete, to be done individually although peer review was allowed. This project was to produce a process improvement plan for a process in which the student was involved - a personal process or a larger, organization-wide process. We proposed a cap of eight hours to be spent on this final project; some students chose to spend more time than this because they intended to use the results directly in a team environment at work.

Grades were computed as follows, with about half the grade based on individual performance and half the grade based on team performance:

- | | |
|---|-----|
| • Homework - individual/team | 20% |
| • Mid-term homework - Process Definition - team | 20% |

- Mid-term homework - Process Improvement - 30%
team
- Final project - individual 30%
- Class participation (individual) subjective

Appendix B. Bibliography and Syllabus

The Bibliography and Syllabus for the fall 1997 Software Process Practicum is attached.

Course Number	CSE 503
Course Title	Software Process Practicum: Lessons in Software Quality and Leadership
Instructors	Judy Bamberger
Days	Mondays, Wednesdays (and one Saturday)
Times	5:30pm - 7:30pm
Room	Cooley Center (CC) 371
Number of Units	4 credits

***** WINNER 1996 Software Quality Excellence Award ***
offered by Pacific Northwest Software Quality Conference**

The software process practicum is designed to immerse the working student in topics relevant to software process improvement and quality management, and to introduce them to the supporting theory. Topics include quality management frameworks (Capability Maturity Model for Software, ISO 9000), measurement for process improvement, process definition, formal inspections, and key team skills necessary for effective collaborative software engineering efforts. In addition to lectures and in-class "labs," the class will include one Saturday workshop. There will be individual and team assignments.

OBJECTIVES / VISION

After this class, you, the students will understand and have demonstrated that:

- Software processes can be managed and controlled.
- Software engineering is a social process, too.
- You have real skills that you can apply today at work.
- You have a framework on which to build your own educated decisions about applying software quality principles and tools to personal, project, and corporate software activities.

- You have identified three things to improve at your own work place (or within your own personal process), and you have begun working on them.

To the Students:

This is a list of required and recommended books.

Readings will be derived from the required books and readings throughout the semester. We will be discussing some of them as part of the class session. They present a unique view of many of the concepts, models, and skills we will be covering - often a different view than would be found in most computer science courses.

The recommended books and readings will provide additional breadth and assistance throughout the class.

I have selected these books and readings because I believe they will be useful to you after this course, in your work environment and in your professional activities. Your feedback throughout the course and afterward will be appreciated.

Required Books

- (1) Scholtes, Peter R et al. *The TEAM Handbook, Second Edition*. Madison WI: Joiner, 1996. ISBN: 1-884731-11-2.
- (2) Weinberg, Gerald M. *Quality Software Management, Volume 1, Systems Thinking*. New York NY: Dorset House, 1992. ISBN: 0-932633-22-6.
- (3) Course pack, required readings (available first day of class at class; available at CSE office prior to class).

Recommended Books

- (1) Brassard, Michael. *The Memory Jogger Plus+*. Methuen MA: GOAL/QPC, 1989. ISBN: 1-879364-02-6.
(general good reference for quality improvement teams)
- (2) Grady, Robert B and Caswell, Deborah L. *Software Metrics: Establishing a Company-Wide Program*. Englewood Cliffs NJ: P T R Prentice Hall, 1986. ISBN: 0-13-821844-7.
(good "how to" for metrics; Class 13 readings taken from here)
- (3) Pardee, William J. *To Satisfy & Delight your Customer: How to Manage for Customer Value*. New York NY: Dorset House Publishing, 1996. 0-932633-35-8.
(one of the most readable references on QFD I have found; Class 18 readings and session taken from here)
- (4) Wiegers, Karl. *Creating a Software Engineering Culture*. New York NY: Dorset House Publishing, 1996. ISBN: 0-932633-33-1.
(a down-to-earth, practical description of people and cultural issues)

- (5) Weinberg, Gerald M, *Quality Software Management, Volume 2, First-Order Measurement*. New York NY: Dorset House Publishing, 1993.
ISBN: 0-932633-24-2.
Volume 3, Congruent Action. New York NY: Dorset House Publishing, 1994.
ISBN: 0-932633-28-5.
Volume 4, Anticipating Change. New York NY: Dorset House Publishing, 1997.
ISBN: 0-932633032-3.

(I find I use a lot of Jerry's ideas in working with the people part of process, quality, and improvement)

- (6) Course pack, optional readings (one copy in library; one copy with instructor).

ORGANIZING FOR SUCCESS

Things to think about from the beginning:

- What are your learning objectives?

What is it that prompted you to take this course? What do you hope to gain by being here? What's in it for you? By considering these types of questions from the start of our time together, we can collaborate on helping you meet your needs as much as possible.

- Who would you like on your team?

You are not required to keep the same team members throughout the entire course. However, especially once the mid-term project is begun, this does have significant advantages. I suggest you begin now, and think about how you could build an excellent, effective, and high-performing team (I will be giving you some hints, too). I also suggest that you use your first team project to try some of those ideas, and set the tone for success.

There are three significant team projects: one homework assignment (H4a), the mid-term project, and the end-term project.

- With which "partner organization" would you like to work?

The mid-term and end-term projects will focus on working with what I call a "partner organization." This could be a team with which you work at your company or school (highly preferred), or it could be a team I recommend to you. You will be collaborating with them to define a process and then to create a plan to improve that process. This will involve some of their time - in the past, it has been about 2 - 12 hours total over the entire term (depends on number of people involved and depth of their involvement).

GRADING CRITERIA

There will be individual and team assignments. I have tried to give most assignments on a Monday, with the turn-in date generally on the following Monday. Team assignments will be given a single grade, which will be assigned to each team member. Team assignments will have an individual component associated with each (graded individually) to analyze team effectiveness overall, and the individual's effectiveness within that team.

The goal of these assignments is to allow you to reinforce the concepts and skills learned in one or more Practicum sessions.

For Class 9, Quality Frameworks, your total grade will be an average of the content you present and your presentation style. I am looking for a clear, logical, persuasive, and authoritative presentation, not "show."

For the mid-term project, the team's total grade will be an average of the written portion, and the content and effectiveness of the team presentation.

For the end-term project, the team's total grade will be an average of the written portion, the content and effectiveness of the team presentation, and the individual written component.

There will be a mid-term and end-term project to be done as a team. The mid-term and end-term projects are related (general descriptions are included in the syllabus). The mid-term project focuses on working with a partner organization to define a software-related process using the techniques we learn in Practicum. The end-term project focuses on working with that same partner organization to identify and plan for improvements to that process.

The goal of these projects is to allow you to synthesize the concepts and skills learned in several Practicum sessions and practice them in a real-world setting. Past projects have also resulted in significant benefits to the partner organization as well, a secondary goal.

There will be a final project to be done individually. This project will be to create an improvement plan for a process in which you are involved personally - individually or as part of a team at work or outside of work.

The goal of this project is to allow you to synthesize the information learned in Practicum and apply it in a real-world, relevant context.

There are two un-graded elements as well.

The goal of both of these are to help me continuously improve the Practicum - both for you this term, and for future offerings of the Practicum.

I will be asking you to keep a Timelog - the amount of time you spend preparing for each class (e.g., reading) and doing the homework assignments. This will have absolutely no bearing on any grade. In fact, I will not look at it until any related assignments have been graded. I will be sharing the class aggregate information with you periodically, and looking at how it relates to the data I collected throughout the last offering of Practicum. I will use this to help me assess and tune the overall workload, week-by-week, assignment-by-assignment. A Timelog template (with instructions) is included in this syllabus.

Please turn in your Timelog sheets each class session.

I will also be asking you to keep a Journal - short notes about the readings and learnings. I will ask to see this three times during the term. I will use this to help me assess the impact of the readings and the messages you take from the classes. Again, this will have absolutely no bearing on any grade, and I will not look at it until any related work has been graded. I will use this to help me identify "what works" and "what doesn't," as well as those articles, class sessions, exercises, etc that have the most/least impact. A set of suggested items to cover in your Journal entries is included in this syllabus.

Please try to make your Journal entries each day, or as new learnings come to you.

The final grade computation. This is what goes into the final grade:

- | | |
|---|------------|
| • Homework (individual/team) | 20% |
| • Mid-term project (team)
Process Definition | 20% |
| • End-term (team)
Process Improvement | 30% |
| • Final project (individual) | 30% |
| • Class participation (individual) | subjective |
| • Timelog (ungraded) | 0% |
| • Journal (ungraded) | 0% |

In general, I expect all written material to be in English, legible, and with appropriate graphics.