Computer Productivity Initiative: Past, Present, and Future

Kurt J. Maly[1], Dennis E. Ray[1], J. Christian Wild[1], Irwin B. Levinstein[1], Stephan Olariu[1], C. Michael Overstreet[1], Nageswara S. V. Rao[2], Deane Sibol[1] and George Panayides[1]

[1] Department of Computer Science, Old Dominion University
Norfolk, VA 23529-0162, (804) 683-3915, email: cpi@cs.odu.edu
[2] Intelligent Systems Section, Center for Engineering Systems Advanced Research, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6364

Abstract. The Computer Productivity Initiative (CPI) is a jointly funded effort by Old Dominion University and the National Science Foundation to address some shortcomings of the traditional CS curriculum. In CPI students apply CS knowledge in the context of a broad range of issues affecting the productive employment of CS technology. The CPI program is also directed towards the development of career skills including group interaction, technical communications, and interviewing as well as domain analysis. This paper discusses the lessons learned from the two year effort to implement this new program. These include: using an external board of industry executives for the final review of senior-level course projects is highly motivating and effective; students take longer than expected to gain competence but then become more competent than expected; a relatively small class size is necessary to implement a "learn by doing" approach; building a prototype to demonstrate concept and assess risks is very effective but can be time consuming; evaluation of a proposed implementation is difficult in the absence of a "real" customer; a better method of reality checks is needed; tight schedules are difficult to fit into traditional semester boundaries; students are enthusiastic about the program and gain confidence in their ability to enter their careers; the level of effort is comparable to that in project oriented courses for both students and instructors; CPI graduates report increased responsibilities and pay as compared to their counterparts.
Overall, we have found that the CPI program is an effective means of involving outside employers with the department, and provides our students a competitive edge in the market place. Starting Fall 1994, CPI will become part of our CS core.

1 Introduction

Typical computer science curricula often teach principles and programming skills in isolation from an application perspective, provide limited laboratory experience, and introduce insufficient integration with non-CS components. The Computer Productivity Initiative (CPI) is a concerted effort to alleviate these problems. CPI is designed to enhance the traditional computer science curriculum to better prepare new computer professionals for their careers. It began as a response to complaints from various sources concerning the quality and preparation of newly graduated students. Many of these issues are recorded in reports by the ACM/IEEE Joint Curriculum Task Force [2] and NSF Computer Science Education Workshop Report (Foley *et al.* [4]), British Computer Society and Institution of Electrical Engineers [1], and articles by Denning [5], Parnas [11], Shaw [12] and Wulf [14]. The continuing discussion of what higher education institutions should be teaching the new computer scientist is documented in several studies. Parnas [11] and Shaw [12] call for changes in the current educational curriculum to meet the needs of year 2000 and beyond (also see the report of the Commonwealth of Virginia's Commission on The University of the 21st Century [3]). Annual departmental self-assessment activities helped uncover the need to modify the curriculum to meet changing needs of industry. The CPI innovation evolved during departmental faculty meetings and was subsequently supported by Old Dominion University. This support led to the commitment of financial resources from the National Science Foundation and Old Dominion University.

The current CS curriculum seemed too narrowly focused. It became evident through discussions with business, industry, and government leaders, including those on our advisory board, that students need a range of skills to help them put their technical expertise into practice.

The initiative developed new courses to alleviate these problems by integrating a multi-year project into the curriculum. This unique multi-year project encompasses courses taken in three different years of the curriculum. It provides significant exposure to the system development process, beginning with a broad, unstructured problem and aiming at a structured solution to that problem.

While software and hardware issues are discussed in CPI, business, engineering, societal and other non-CS issues are emphasized. Career-oriented skills such as budgeting, interviewing, scheduling, and formal presentation skills are taught. Other topics include cost estimation, market research, prototyping, simulation and technical writing. But these subject areas are not just taught in lectures, they are practiced by the students. CPI students apply their newly obtained knowledge to the project. They work in groups and are equipped with leading-edge technologies.

CPI is designed to be a portable curriculum modification. That is, the changes proposed by the CPI concept are flexible and can be adapted to most CS department's current curriculum. For the Computer Science Department at Old Dominion University, two new senior-level courses, Computer-Based Productivity I and II were added. Two existing courses, Computers and Society (junior-level) and Problem Solving (CS 2 in the ACM curriculum) were slightly modified to support the senior level courses. Students in the senior-level courses task the students in the lower level courses to provide them with additional information. For example, students in the Computers and Society course may be asked to research and discuss what impact the implementation of a proposed system might have on society-at-large. Students in the sophomore Problem Solving class may be asked to perform an experiment and provide their results to the senior-level students who might use this information to justify a design choice.

The main goal of Computer-Based Productivity I (CPI 1) is to generate and justify a solution to a broad, unstructured problem. This involves a thorough understanding of the problem area (domain analysis). As solutions are suggested and refined, they are presented to the class. Early in the semester, students are taught presentation skills which are practiced in 4 formal presentations given throughout the semester. Students make site visits and interview local experts. The semester ends with a final formal presentation to the CPI advisory board (consisting of local business, industrial, and government leaders) who evaluate the students proposed solution to the project.

Computer-Based Productivity II (CPI 2, the second of the two course series) aims to refine, define, and demonstrate the solution to the problem. Given the proposal of the previous semester's class, the students must evaluate and enhance these ideas and write functional specifications. Students identify the areas of greatest risk for deeper evaluation. Prototyping and simulation are utilized to evaluate and narrow the focus of the functional requirements of the proposed system.

In its first year, CPI focused its energy on defining the contents of these two new senior-level courses and the necessary changes to the sophomore and junior-level courses. An implementation plan was proposed and an initial CPI senior-level course was offered, as described in [9]. In its second year, the initiative offered the full sequence of senior-level CPI courses. In Fall 1994, CPI will begin its final phase of development in which course content and inter-class cooperation will become stable.

While the development of CPI continues, the purpose of this paper is to provide a summary of the initiative's current accomplishments, emphasizing the means by which it overcame the many obstacles it faced. In Section 3, we present the steady state requirements of the CPI curriculum. Finally, we report on CPI's current status and future work in Section 4.

2 Lessons Learned

The implementation of CPI has been shaped by two major conflicts:

- On the one hand we seek to engage students in a problem-solving process which is by its nature ill-defined and unpredicatable and on the other hand we require that this process occur, with a proposal as its product, within an academic semester.
- On the one hand we are delivering a course which should provide approximately the same information and experience to all its participants and on the other hand we are using as a delivery vehicle a project which by nature uses division and specialization of labor in order to meet a one semester deadline.

These conflicts have been manifested in several problems which arose during the development and delivery of CPI. For example, the first conflict is found in the problem of scheduling outside speakers and other activities at an appropriate time and with appropriate content when it is difficult to anticipate the progression of the problem-solving process in which the students will engage; while the second conflict led us to neglect some aspects of the project so that all students would be involved in the development of one prototype or another.

In this section we describe each CPI related course and present a brief evaluation of each, based on its current contents. Fig. 1 shows the interactions among these courses.

CPI 1 Course Contents and Lessons Learned

In order to achieve the broad set of goals established for CPI 1, the traditional class lecture has been augmented by guest speakers, class field trips, and individual student visits to corporations. Deciding on the tasking of the supporting classes added a new element to the requirements on the students. The infusion of project support equipment and computer application tools into the course expands student's use of various systems including planning, development, presentation tools, and multi-media applications. Several formal presentations by each class member with emphasis on professionalism required students to become more involved in the direction of the class. The establishment of a self-organizing student structure with defined goals and responsibilities highlights the type activity that is expected of students after graduation.

The broad scope of topics and concepts presented in CPI 1 critically emphasizes the fundamental instructor's need: know your subject.

The broad perspective of topic material can easily consume the instructor's available time for course preparation. It is a challenge to adequately budget the time to provide simple directions to students. The goal has been to lead students through the decision making and problem solving phases of the project, requiring the students to make the critical decisions essentially on their own. The requirement that the students succeed in developing a project solution requires a very close scrutiny on the part of the instructor. The approach of having students "learn by doing" remains as a superior method of providing the students with the necessary lifelong skills goal of CPI.

It has become clear that only with a relatively small class can the instructor fully comprehend how well the students are solving problems and employing the productivity tools made available to them. This teaching requisite cannot be delegated nor obtained without the close and conscientious vigilance which an instructor can provide.

Students entering the course generally have received no direct education in the methods and practices of preparing a technical presentation. The development of these lifelong skills is critical in the overall success of increased student productivity. Students must be provided a firm foundation in the effective methods of presenting.

The introduction of multi-media and other leading-edge technologies has a mixed acceptance among students. Some see the technology applied to the solution of their problems, others view the technology as exciting, and still others appear somewhat ambivalent as to its use. By allowing much greater freedom of choice by students as to what set of technology they will take the time to learn and use effectively, a larger number of students benefit and much more productive work is accomplished by the class as a whole.

In almost every situation, the student learning curve greatly exceeded the expected value. Yet with only an estimated apparent knowledge and capability, students have repeatedly overcome their skills inabilities and met imposed course deadlines. In general students have not just met the deadlines, but excelled in their performance. It would appear that instructors might routinely underestimate the students' capabilities.

Tasking of CS 300 (Computers in Society)

Senior students initially seem enthused with the prospect of tasking CS 300 (Computers in Society) with research concerning their proposed topics. This enthusiasm has worn thin as the necessity of precision of definition and the timing requirements begin to generate work for the seniors. When the results do come back from the tasked class, they are both later than desired and not at all what the seniors had expected. The CS 300 data tend to show a broader perspective and include many negative aspects that the designers of CPI 1 had not considered.

The original choice of the course project was obtained using as much information as could be identified. The open ended broad topic area has been one of the best decisions that the CPI committee made. Students like the flexibility and the close relationship that they can make between the project topics and their own personal lives. They see the current advertising market place that highlights many of their ideas. This correct choice of project topic area seems to stimulate student's interests and act as a fluid vehicle for the digesting of the course topics.

To date the course completion interviews with students has been very positive. Even more so are the positive comments from CPI graduates. While the potential sample size is still small, the success stories are clear indications of the overall success of the CPI methods. The students' overwhelming response has been to continue with CPI. CPI graduates report that their current working positions are enhanced as a direct result of the work accomplished during the CPI semesters.

While these feelings of success abound, they come with a cost. The time and effort to describe and develop the course materials has been large. The combined efforts of the entire CPI staff again is large. The results however have been so clearly in keeping with the achievement of the prime CPI goals that these one-time efforts are worth the results. With the availability of the rough draft of materials for the continued steady state teaching of the CPI concepts and topics, time and effort to continue the CPI work should be no greater than other project based courses.

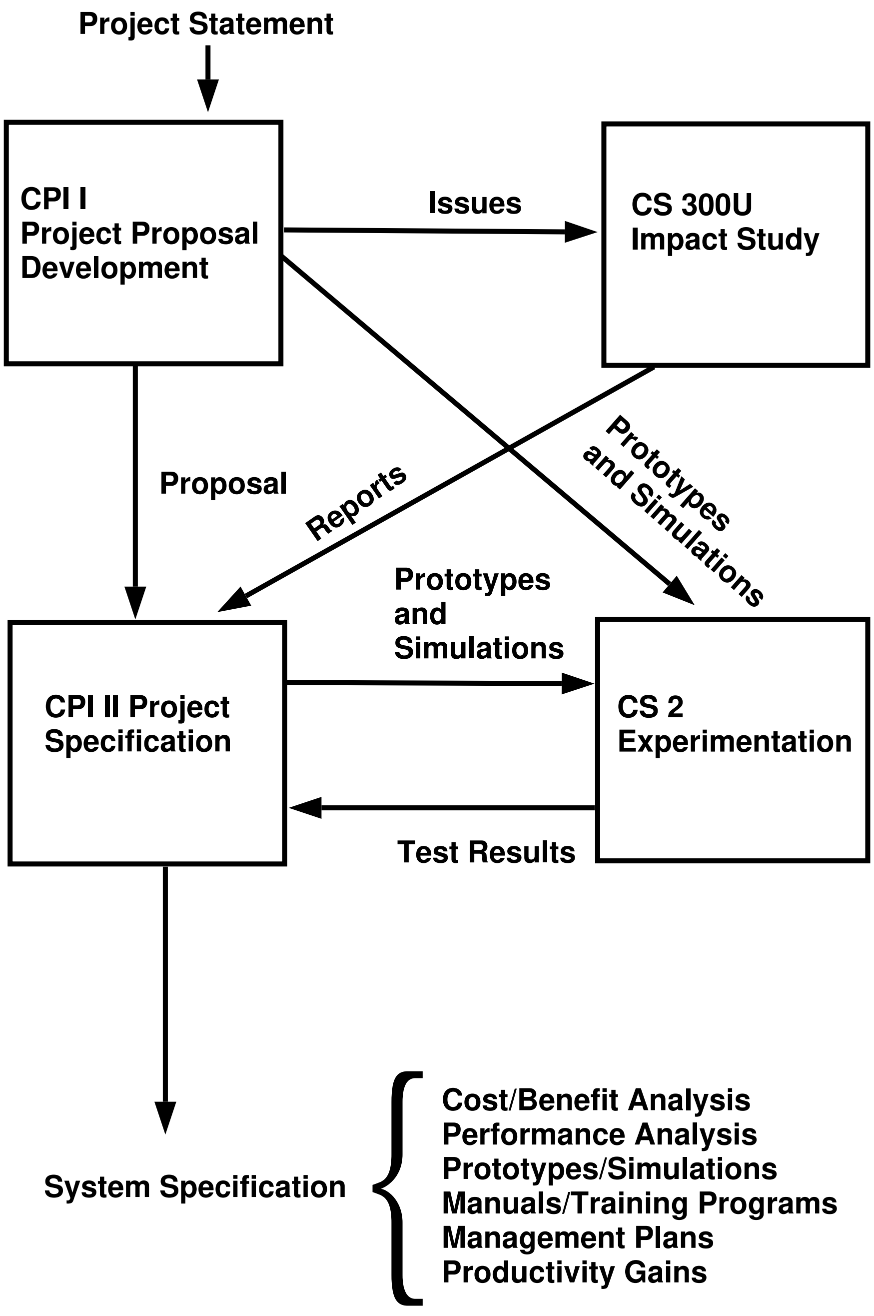

Fig. 1. Interaction among CPI Related Courses

CPI 2 Course Contents and Lessons Learned

A key focus of CPI 2 is to acquaint students with procedures and tools for evaluation and planning. This is done through their evaluation of the high level product specification completed by CPI 1 students during the previous semester and preparing more detailed requirements and plans for further development of the product. Students also provide supporting materials to explain and justify all decisions which they make. As with CPI 1, the class ends with a major presentation to an external advisory board, complete with demonstrations of prototypes and written back-up materials.

The major student activities planned for CPI 2 included a review of the CPI 1-prepared specifications, a risk analysis (with an emphasis in refining/improving the requirements to reduce risk of failure in the envisioned product), planning and scheduling (both short-term limited to the class activities for the semester, and long-term, for complete product development), product development cost estimation, preparation of a project management plan, and preparation of a verification and validation plan. While the results of students' risk analyses affect the choice of activities for the semester, the faculty have identified a specified set of activities which should be included in each offering of CPI 2: we planned to introduce prototyping and simulation as basic tools to evaluate and refine specifications and to introduce students to existing document standards (such as those published by IEEE for requirements or planning [7]) whenever appropriate. In addition, we intended to include a discussion on the design of computer-human interfaces since for many consumer-oriented products design of the user interface is critical to its success.

Not all of these topics were covered, primarily due to time constraints. Students did prepare individual risk analyses based on the CPI 1-specification, prepared schedules for their semester-long activities, identified two components whose development seemed to entail the highest risk (or whose prototype might be impressive to the external board): a home grocery shopping system (which raises interesting user-interface issues) and an integrated telephone/television system which uses the TV to display videophone images. They used COCOMO to prepare cost estimates for the development of the software for these two components. This was based on a modified Delphi technique: students prepared individual high level designs for each component, agreed on a single design for each, and prepared individual size and factor estimates (as required by COCOMO) for each component, refined these into single consensus estimates, and used these as inputs to COCOMO. They also prepared an initial draft of the system specification using the format described in IEEE-830 [7]. Some of the draft came directly from the specification prepared the previous semester, particularly for hardware since this was not identified as a major risk area. Since IEEE-830 is a standard for software requirements rather than system requirements, additional sections were added to describe hardware requirements and the partition between hardware and software functionality. Students also prepared user's guides for each component as supplements to IEEE-830. Upon completion of experimentation with the prototypes, the draft systems specification was completed.

In order to get a quick initial evaluation of the basic functional and marketability of the home grocery shopping component, students conducted a focus group [6, 13] using adults who might be interested in such a system. The purpose was to get an initial reading on desirable functionality from people who might buy the product before time was invested in specifying and building a prototype. The purpose of the prototypes was to evaluate the proposed functionality for the two components; no effort was made to evaluate system performance. Thus the prototypes were used to evaluate the user interface and most key features of the functionality of the proposed system. Students specified the behavior of the prototypes before beginning their design and construction. These prototypes were later used by people assumed to have similar demographics to potential product customers, non-computer science students at Old Dominion University.

Several planned topics were not covered; no simulation due both to time constraints and the fact that the class identified no risks whose resolution was reasonable addressed by a simulation study. In addition no project management plan nor project validation and verification plan were developed.

Several lessons were learned through this initial offering of CPI 2.

- The prototypes were well received by the board – better then "just a stack of reports." The user's manuals were also positively perceived by the board, apparently more than the IEEE-830 based specifications. But everyone, students and board members, like to see things that actually do something.
- Too many activities were scheduled. With more careful planning and scheduling, some more can be included, though scheduling will still be tight.
- Use of the focus group was effective, particularly for the general consumer-oriented product the students were working with. This gave the students early feedback about perceptions of potential consumers. In addition, the depth of the study can be controlled based on available time.
- Our prototype evaluation was weak, but with forethought, can be easily improved. Each CPI 2 student recruited three non-computer science students to evaluate each prototype. Evaluation included performing an assigned set of tasks while an observer (a class member) noted any difficulties and was available to answer questions if required. Each evaluator then completed a questionnaire. The problem with this approach was that the evaluators were often friends of the students and were usually not critical of what they had seen. More independent evaluators should be used.
- A strong point in the use of both focus groups and prototype evaluation is the direct interaction of computer science students with potential users.

Tasking of CS 2 (Problem Solving and Programming)

A central concept in CPI is the involvement of students in several computer science classes in key aspects of the current CPI project. Students taking CS 2 have participated in CPI through the programming assignments used in that class. Our original idea was to have CPI 2 students task students in CS 2 to help with certain aspects of the project. For this initial offering, timing constraints of both, CPI 2 and CS 2 were such that participating faculty identified laboratory assignments for CS 2 which would contribute to the project. The instructor for CS 2 picked a set of programming assignments motivated by the current CPI project. In addition, CS 2 students, using a toolkit developed for them, experimented with several forms of an interface for picking one item from a very long list of items (such as selecting one grocery item from the inventory list of a typical grocery store). They then measured time and accuracy for different versions of the interface by having other students use the software. This information was then made available to the CPI 2 students.

Proper timing of these activities is difficult. The information from the CS 2 student studies came too late in the semester to influence the interfaces incorporated into the prototypes. In the future, we think we can better address these timing problems by having students in CPI 1, with the cooperation of the CPI 1 and CS 2 instructors, request information which CS 2 students can provide though a lab assignment, then this information will be used the following semester by the CPI 2 students.

3 Steady State Requirements

In this section we consider what will be required to operate the CPI curriculum once the cost and effort of introduction has been accomplished. While the cost in terms of effort and actual dollars is expected to be somewhat higher than the cost of a conventional curriculum, it is not markedly so; we feel that the advantages are worth the small increase.

The steady state itself may take two different forms, one in which the project area is known and relatively familiar and one in which the CPI program is transitioning between project areas. The former condition is discussed first, assuming that the instructors are familiar with the project area, have been trained or self-taught in areas such as risk analysis and presentation techniques which are prerequisite for teaching the courses and are in possession of teaching materials to support their classroom work. This is followed by a discussion of the costs of making a transition from one project area to another. Finally, software and hardware costs are discussed.

Curriculum Requirements. Adoption of the CPI curriculum requires the inclusion of two new courses for seniors, the modification of one sophomore and one junior level course, and the establishment of an advisory board. Including the two new courses may increase the total number of hours required of students in the major. In our case students take three more hours in CS than they did under the previous curriculum. The effect on the department is that we have slightly heavier staffing requirements (1 course per year) than we did previously. Obviously this will vary with each department's situation.

The staffing requirements of the two existing courses that are modified to interact with the added senior level courses will be unchanged, but the question is whether the workload will also be unchanged. In the case of our Computers and Society course, the modification of the course was minor: the issues students were assigned to investigate were oriented toward the CPI project and student reports were turned over to the CPI project.

In our CS 2 course, a section on scientific method was added and students were assigned to design and perform experiments related to the CPI project. Typically the experiments involve aspects of user interfaces related to the project. We have been developing an X windows interface design kit which allows the TA for the CPI course to create quickly the alternative interfaces which are the basis for the experiments. In steady state there is little overhead since the kit will have been built. The kit will be available from us to interested parties. There is some overhead in coordinating details of the CS 2 experimentation with the needs of the CPI project, but that is fairly small so long as the project remains the same from one year to the next.

The CPI advisory board is a group familiar with business and government practice in the real world. In our case we have been fortunate to recruit CEOs and government executives, but the main requirement is that the board bring a breath of reality to an otherwise academic exercise. The board is consulted when the department is seeking an appropriate project. It serves as the audience for, and critic of, the students' proposals and designs. Once the board is established it becomes a valuable source of ideas and criticisms and the cost is small: a few meetings and working luncheons per year.

Effort Analysis – Faculty. The brunt of the effort falls upon the instructors of the two CPI project courses. The faculty members who have taught the courses report that the effort required is similar to that in a project oriented course or a large course with programming laboratories. One instructor, who regularly teaches our Software Engineering course, thought that there was somewhat less effort required for CPI because the size of the class was considerably smaller. Another instructor, who regularly teaches large sections of our CS 1 course, noted that CPI required more class preparation but that the CS 1 course required more administrative overhead. We have not been able to measure the time required of the instructors of the courses in any other way than this self-reporting fashion.

During the development period of the course we allotted two TA's to CPI. We feel that one TA is a necessity.

Effort Analysis – Students. To determine the effort expended by students in the CPI project compared with those enrolled in traditional courses, we used electronic mail to poll students enrolled in CPI and two other courses each week for 13 weeks for reports on the number of hours they spent on their classes. The courses involved were CPI 2, a graduate/undergraduate class in systems programming, and a class in formal languages. The systems programming class is clearly a project class, while the formal languages class most definitely was not (See Tab. 1).

Students were simply asked to send by return mail the number of hours expended that week. We collected only the number of respondents and the total hours reported each week for each of the classes. The number of students responding each week remained fairly consistent for each class, but it is impossible to tell whether the same people were the ones responding. While self-reported effort numbers may not be reliable, we assume that the numbers from the different classes are comparable and subject to similar distortions. Consequently our data can only be suggestive.

Class	*Avg. Hrs./Week*
CPI	14
Systems Programming	
Undergraduate	13
Graduate	15
Formal Languages	6

Table 1. Student Effort Statistics

This data suggests that the effort expended by students in the CPI course was approximately the same as effort expended in a typical project course.

Transition Between Projects. Selecting a new project topic is a time-consuming effort. Potential topics have to be discovered, investigated and evaluated for suitability – an undertaking appropriate for a committee. Once chosen, the instructor must acquire more knowledge of the subject and locate instructional and other resources for the class. We anticipate that a well-chosen topic can support a CPI program for about three years. Fortunately, such efforts can be shared among institutions. Our chosen topics and supporting materials are available on line and we will cooperate with institutions adopting CPI in creating a repository for such projects. While this will greatly mitigate the effort required for transitional years, there will still be an extra burden on the instructors familiarizing themselves and locating local resources at those times.

Software Support. The software required to support CPI falls into 4 categories: course materials, decision management, project management, and presentation graphics. The course materials (including the interface design kit) and the decision management tool (D-Hyper Case) [10] will be available at no charge from ODU and thus involve no cost to any Unix-based department. However, the project management and presentation graphics software may present more of a problem. These two kinds of software are vital to the courses since the students' work is focussed on presentations made to our advisory board and they are responsible for developing their own schedules and projecting others.

It has been our experience that Unix-based software in both areas is very sophisticated and consequently both expensive for a department to purchase and difficult for students to master. On the other hand, PC-based software tends to be much cheaper and more user-friendly. We hoped to keep the course on one computing platform (Unix) but found that we could not afford to purchase both kinds of software. We decided to use PC-based project management software (Microsoft Project) and Unix-based presentation graphics (IslandPresents). This course was taken because many of the materials residing on the Unix system would be incorporated into the presentations while the project management software was relatively independent of the rest of the course. Other departments may well choose PC-based presentation software. It is much cheaper – less than \$200 per copy as opposed to \$1500 for an annual license for 2 simultaneous users, easier for students to use, and unencumbered with an intrusive license manager. The software used to support CPI is summarized in Table 2.

Hardware Support. Once hardware for the curriculum has been purchased, there is only maintenance cost. Some departments may spread the acquisition of hardware over several years and could thus regard this cost as part of the steady state. Our implementation of CPI provided a separate classroom/library equipped with a workstation and a large-screen television, a small (4 workstation) lab which included one multi-media equipped PC. The hardware used for CPI is summarized in Table 2. Since the senior courses are presentation oriented, the large screen (or projection) television and the multi-media workstation are necessities, but the separate laboratory and classroom, while highly conducive to teamwork and group identity, are not as vital.

	CPI				Comparison	
	CS2	CS300	CPI 1	CPI 2	CS390 (theory)	CS476 (project)
#TAs (20hrs/wk)	.5	.5	1	1	.4	.4
#speakers	1	3	3	3	0	0
Hardware	wkst	wkst PCs	4wkst 2PC TV+VCR 2camcds 2way video cards	4wkst 2PC TV+VCR 2camcds 2way video cards	wkst	wkst
Software	Unix/X Test environ.	Unix/X, Emacs WordPerfect, Word, etc.	Unix/X Island MS Office MS Project DHC Mosaic Emacs	Unix/X Island MS Office MS Project DHC Mosaic Emacs	Unix/X	Unix/X

Table 2. Summary of Requirements Data for CPI

4 Current State and Future Work

Starting in the Fall 1994, CPI will become part of our undergraduate curriculum. However, CPI is still an experiment in progress and must evolve as we gain a better understanding of its role in an undergraduate education. Our immediate goal is the development of a multimedia hyperlinked instructor's manual which includes slide packages for each curriculum module, example documents supporting the course objectives and the decision space which describes how we arrived at the particulars of the course contents. This manual will be available through MOSAIC over the Internet as part of our effort to disseminate the CPI project to other educational institutions.

The issue of transportability will not be solved simply by the availability of an instructor's manual. Therefore we are pursuing two other methods of dissemination. As part of a university wide effort to provide remote education throughout the state of Virginia, we are developing a networked, interactive multi-media workstation-based educational facility. We will use this facility to support team project courses like CPI in which students at different sites can interact using Computer Supported Collaborative Work (CSCW) technology [8]. Although this will initially be an internal effort, we anticipate that it can be used to transfer CPI to other institutions through a collaborative team teaching effort. A second method of transporting CPI will be through the use of a sabbatical program wherein instructors from other institutions will spend one semester at ODU team teaching part of the CPI project.

Although the course objectives for CPI have stabilized, we will need to continually monitor skills learned and to assess the value of these skills against others which could be substituted in their place. We are aware that any addition to the curriculum means that something else must be sacrificed if the time to complete the undergraduate degree is to remain constant. Thus, while we feel that the set of skills offered in CPI are vitally important to our students and that they are not available elsewhere in the curriculum, we are sensitive to fact that other skills will not be learned. One unforeseen side effect of CPI is the emphasis on user interface design issues which are only briefly covered elsewhere.

Evaluation of CPI will be an ongoing activity as the curriculum evolves and matures. We have found that the external board of advisors provides invaluable insights into current and future directions. Having the students present their project to the board provides a very real incentive towards excellence and can reflect poorly on the institution if poorly done. Although it can be a nerve-wracking experience, it builds the student's confidence in a way that would be difficult to do so otherwise.

Because group interactions are an essential part of the CPI experience, assigning individual grades becomes a difficult problem. In other classes in which we have team projects, the tasks have been divided in a way which allows individual credit to be assigned even if some members of the team fail to perform their assignments. However, with CPI, one of the lessons to be learned is that the success of the team is important and that part of the student's grade will depend on the success of the team. We feel strongly about this since in their careers, students will also be judged in part on the success of their team. However, we also recognize each student needs to be evaluated separately from the team. Therefore, for the fall 1994, we are planning to use self evaluations to assign individual grades. At the beginning of the semester, the class will be told what the objectives of the course are, and that they must maintain a portfolio which documents their individual efforts in meeting those objectives. At the end of the semester, each student will present this portfolio to the instructor and must persuade the instructor that they have earned whatever grade they feel their efforts justify. It is planned that part of the portfolio will be monthly progress reports.

Although we have only had a few students who have finished the CPI project and graduated, their satisfaction with the experience and the degree to which it prepared them for their careers seems to validate the concept.

5 Conclusion

The role of undergraduate education is being redefined in an era when business and government are striving to become more productive and competitive. The most successful graduates may not be those with the best technical education but rather those with the better understanding of how to apply that education to the solution of real problems. The CPI project has been developed through a joint effort of NSF and ODU in order to provide a curriculum which better prepares students for their future roles in society. CPI has matured to the point where it has become part of a multi-year coordinated project within our CS curriculum. Initial feedback from our students and from an external board of advisors confirms our feelings that this effort can provide an additional dimension to the traditional CS curriculum which better prepares students to contribute to the solution of ill-structured but real problems. Our current efforts are to extend CPI to a networked remote learning environment.

6 Acknowledgments

This work is funded the by National Science Foundation under grant #CDA-9214930 and Old Dominion University.

References

1. A report on undergraduate curricula for software engineeing. British Computer Society and Institution of Electrical Engineers, 1989.
2. ACM/IEEE-CS Joint Curriculum Task Force, 1991. Computing Curricula 1991.
3. The Case for Change. Commonwealth of Virginia, Commision on The University of The 21st Century, Richmond, Virginia, 1992.
4. J. Foley et al. Report of the NSF Computer Science Education Workshop. *SIGCSE Bulletin*, 20(3), 1988.
5. P. Denning et al. Computing as a discipline. *Communications of the ACM*, 1, 1989.
6. Thomas L. Grrenbaum. *The Handbook for Focus Group Research.* Lexington Books, 1993.
7. The Institute of Electrical and Electronics Engineers, Inc. *IEEE Standards Collection*, 1993.

8. K. Maly, C. M. Overstreet, H. A. Wahab, and A. K. Gupta. Melding television, networking, and computing for interactive remote instruction: Exploiting potentials. accepted for presentation, June 1994. World Conference on Educational Multimedia and Hypermedia'94, Vancouver Canada.
9. K. Maly, D. Ray, I. Levinstein, S. Olariu, C. M. Overstreet, N. Rao, T. Ireland, and G. Kantsios. Computer productivity initiative. *Seventh SEI Conference on Software Engineering Education*, San Antonio, Texas:441–459, January 1994.
10. K. Maly, C. Wild, C. Zhang, D. Ray, I. Levinstein, S. Olariu, C. M. Overstreet, N. Rao, D. Rosca, T. Taylor, T. Ireland, and D. Sibol. A decision-based hyper-multimedia case environment for the computer productivity initiative, a large undergraduate project. accepted for publication, June 1994. World Conference on Educational Multimedia and Hypermedia, Vancouver Canada.
11. D. L. Parnas. Education for computing professionals. *IEEE Computer*, 23(1):17–22, 1990.
12. M. Shaw. We can improve the way we teach CS students. *Computing Research News*, 4(1):2–3, 1992. letter to the editor.
13. Jane F. Templeton. *Focus Groups: A Guide for Marketing and Advertising Professionals*. Probus Publishing, 1987.
14. W. A. Wulf. SE programs won't solve our problems. *Computing Research News*, 3(5):2, 1991. letter to the editor.

This article was processed using the LaTeX macro package with LLNCS style