WPI-CS-TR-99-24 August 1999

1999 Computer Science Department MQP Review

by

Micha Hofri
Craig E. Wills

Computer Science
Technical Report
Series

WORCESTER POLYTECHNIC INSTITUTE

Computer Science Department
100 Institute Road, Worcester, Massachusetts 01609-2280

Abstract

This report presents results of a peer review of MQPs conducted within the Computer Science Department during the Summer of 1999 as part of a campus-wide MQP review. The goal of the report is to assess whether the department MQPs are accomplishing their educational goals. An additional goal of this year's report is to examine student learning outcomes as they relate to departmental goals.

The report identifies problems that need to be addressed and trends that need to be continued to make the MQPs a worthwhile learning experience. It reflects data and evaluations for 31 MQPs, involving 65 computer science students, that were completed between the Summer of 1998 and the Spring of 1999. The report also makes comparisons to similar reviews done in the past.

Overall, the large majority of the projects are meeting the educational goals of the department as good learning experiences. The reviews indicate that the overall quality of the projects is up a bit from the 1997 MQP Review. The reviewers believe that the faculty must set reasonable expectations for all project students. This will be increasingly important as the department continues to grow in size. The report draws a number of conclusions about the success of the projects based upon the data collected and evaluations done for this review. The report concludes with recommendations for future reviews as the department uses the MQP Review as part of a larger department assessment effort.

1 Introduction

1.1 Purpose

The Major Qualifying Project (MQP) is required of all undergraduate students at Worcester Polytechnic Institute. The MQP within the Computer Science Department is a capstone experience, requiring one unit of work, that gives students practice on applying the fundamentals and skills they have learned to a large problem in the field of Computer Science. The project may involve original research, data collection, analysis, or design of a system and often a software implementation. The approach is determined by the student/advisor team. The MQP allows students to study an area of Computer Science in depth, or allows them to combine areas into a single project.

This report presents results of the fifth biennial peer review of MQPs conducted within the Computer Science Department during the Summer of 1999 as part of a campus-wide MQP review. The goal of the report is to assess whether the department MQPs are accomplishing their educational goals. The report identifies problems that need to be addressed and trends that need to be continued to make the MQPs a worthwhile learning experience. It reflects data and evaluations for 31 MQPs, involving 65 computer science students, that were completed between the Summer of 1998 and the Spring of 1999. The report makes comparisons to the following reviews:

Year	Number of MQPs	Number of Students
1991	19	31
1993	26	44
1995	23	43
1997	29	57
1999	31	65

1.2 MQP Role in Student Outcome Assessment

Examination of student learning outcomes is a point of emphasis for the campus as a whole and it is particularly important to the computer science department as it seeks to measure the effectiveness of its degree program. The MQP is the capstone component in the degree program so it is natural to use it for measuring student learning outcomes.

During the 1998-99 academic year, the department began discussions on writing a mission statement along with a set of desired student learning outcomes, but at the time of this review such a statement had not been approved by the department faculty. However, a working draft version of a mission statement, modeled on the WPI mission statement, is posted on the Department Web site [1]. This draft, along with CSAB (Computing Sciences Accreditation Board) [2] program criteria and ABET (Accreditation for Engineering and Technology) [3] outcomes criteria were used to develop expected learning outcomes for purposes of this review. The data collected during the review process was then used to determine if these learning outcomes had been exhibited by MQP project members.

1.3 Procedure

The peer review was conducted during the Summer of 1999 by Micha Hofri, department head, and Craig E. Wills, associate professor. The review was to be for projects completed during the 1997-1998 and 1998-99 academic years. Rather than examine a sampling of reports for a two-year period, the peer review team examined projects completed between the Summer of 1998 and the Spring of 1999. This methodology is consistent with past MQP review practice. The report for each MQP was obtained from either the project advisor or from the Gordon Library. Additional project information was gathered from CDR (Completion of Degree Requirement) records.

Despite the desire to measure student learning outcomes in this year's review, the review process was generally unchanged from previous years [5, 6, 7, 4]. This approach was used to ensure longitudinal analysis of results with previous years and because the review team believed that measurement of learning outcomes could be derived from the data that would be collected. The reviewers conducted a detailed evaluation of all projects using the review form in Appendix A. The form contains information used in classifying the projects, questions quantified on a scale between 1 and 5, and has questions for written comments concerning the report. The form was designed to be easy to fill out with information that could be quickly collected and compared. Questions for written comments concerning the report were used to gather more detailed information about the project and give a means to express specific project strengths and weaknesses. Project grades and registration information was obtained from CDR records. Grades were not consulted until after the MQPs were reviewed.

The MQP reports were divided between the two reviewers for evaluation. A common MQP was initially read and evaluated by both reviewers as a basis for discussing evaluation criteria. After all evaluations were completed, the data from the forms were collected and analyzed. This report is the outcome of the peer review process. Section 2 presents the results from the evaluation forms. Section 3 analyzes and correlates the results. Section 4 analyzes the results relative to desired student outcomes for the undergraduate computer science degree program. Section 5 discusses conclusions and recommendations.

2 Results

This section presents the results of the Computer Science MQP evaluations. Along with presentation of the results are included reviewer comments (denoted by **Comment:**) which highlight the results and contrast them against those from previous reviews when appropriate. Note: All data are presented on a per project and not per student basis.

All percentages are represented in whole number amounts (i.e., 1/31 is represented as 3%), and all number averages are represented to one decimal accuracy (i.e., 1.97 is shown as 2.0). Because of this formatting, the percentages do not always total to 100%.

2.1 Faculty/Student Ratio

Table 1 shows the percentage of projects with the given numbers of students and faculty. None of the projects were completed with students from other departments, nor were any faculty from outside of the department listed as official project advisors.

The average number of students per project was 2.1. The average number of faculty per project was 1.3.

Comment: The results show that eleven projects (35%) of the projects were done by a single student. This number is up from the 1997 figure of 28%, but down from previous studies where the figure was over 40%. However, the average number of students per project increased slightly to 2.1 from 2.0 in 1997 with the number of three- and four-student team projects increasing from 20% to 38%. The number of projects advised by a single faculty member was 23 (74%), virtually the same as 73% in 1997. The formal involvement by students and faculty outside of the department fell to zero for the first time in the history of the MQP peer review.

Table 1: Percentage of Projects with the Given Number of Students and Faculty

	Students				
Faculty	**1**	**2**	**3**	**4+**	**Total**
1	26	19	26	3	74
2	6	6	6	3	23
3+	3	0	0	0	3
Total	35	26	32	6	100

2.2 Faculty Project Load

Table 2 shows the distribution on the number of projects (co-)advised by each faculty member. There were 16 full-time faculty in Computer Science during AY98-99 (one faculty was on sabbatical) plus one visiting professor who advised projects. Table 3 shows the same data, but in cases where projects were co-advised a weighting of one-half project load was given to each advisor.

Comment: The data show that one professor advised seven (23%) of the projects while two professors had no projects (one had just returned from sabbatical, the other was a first year faculty member). Aside from these, the project load was dispersed among the remaining faculty members. The average project load dropped to 2.4, from 2.6 in 1997. The comparable average loads shown in Table 3 dropped from 2.2 projects/faculty in 1997 to 1.8 projects/faculty in 1999. This number is more in line with numbers from 1991 and 1993 peer reviews. However, these numbers are expected to significantly increase since the number of computer science majors at the lower levels has steadily increased each year.

The Gini Coefficient, a number between zero and one was also calculated for Tables 2 and 3. This coefficient measures the degree to which projects are evenly distributed amongst faculty with a coefficient of zero indicating perfect distribution and a value of one indicating all projects being advised by a single faculty member. The respective coefficient for the two measures of project load are 0.406 and 0.453. The comparable figures for 1997 are 0.484 and 0.554 indicating an improvement in more evenly distributing the projects amongst department faculty.

Table 2: Distribution of Projects Advised or Co-advised

Number of Projects (Co-)Advised	Number of Faculty
0	2
1	4
2	5
3	2
4	2
5	1
7	1

avg: 2.4 projects/faculty

2.3 Off-Campus Projects

Five (16%) of the projects were sponsored or involved off-campus companies and organizations. The sponsors were Goddard Space Flight Center, Worcester Telegram & Gazette, Clariion, Natural Microsystems and the Commonwealth Scientific and Industrial Research Organization (CSIRO) in Sydney, Australia. The remaining projects were done on-campus and not sponsored by off-campus companies.

Comment: The number of off-campus sponsored projects was lower than the 24% number in 1997.

2.4 Project Grades

In the projects reviewed, 71% of the projects (77% of the students) received a final grade of A, 26% of the projects (17% of the students) received a final grade of B, and 3% of the projects (6% of the students) received a final grade

Table 3: Distribution of Load of Projects Advised

Load of Projects Advised	Number of Faculty
0	2
0.3	2
1	4
1.5	3
2.8	1
3	2
3.5	1
4.5	1
5	1

avg: 1.8 projects/faculty

of C. These numbers are in line with the campus-wide historical averages where 70-75% of the students receive an A on their project. Note: three projects resulted in members on a given project receiving different individual grades. For purposes of this review, the project grade used for these projects was the average of the individual student grades.

Comment: These data indicate the number of A grades given to projects was about the same as the 1997 figure of 72% (71% of the students). Fewer projects, but about the same number of students, received a grade of C when compared to 1997 when 14% of the projects (7% of the students) earned that grade.

2.5 Project Continuation

Nine projects (29%) were continuations of prior MQPs and MS theses. The other projects were not directly related to other projects.

Comment: These numbers are substantially up from 1995 and 1997 when 14% and 7% of the projects were continuations. The results indicate that faculty are doing better in integrating new projects with previous work.

2.6 Project Duration

Table 4 shows the duration of each project. 87% of the projects finished with one unit of work. 26% of the projects were done in a shorter time frame than the "traditional" three term span. One of these projects was done is a single term with most done in two-terms often due to students wanting to graduate in mid-year.

Comment: This number compares to 42% (1991), 54% (1993), 63% (1995) and 73% (1997) projects completing with one unit of work. These figures indicate a steady trend of better efficiency by students and faculty in completing projects on time. Faculty may also be less willing to award students more than one unit of work as 16% of the projects spanned four or more terms, but resulted in only one unit of credit.

Table 4: Percentage of Projects with the Given Duration in Terms and Units Earned

	Units		
Terms	**1**	**1 1/6**	**Total**
1	3	0	3
2	23	0	23
3	45	0	45
4	13	10	23
5+	3	3	6
Total	87	13	100

2.7 Project Report Size

The average size of the project reports was 50 pages (with a range of 22–146), which excludes appendices and code. The average size of the appendices for a report was 26 pages (with a range of 0–200).

Comment: The length of reports is about the same as previous years 45 (1991), 49 (1993), 50 (1995) and 59 (1997)

2.8 References

The average number of references was 12 (with a range of 0–31) for each report. Many projects did not have an explicit literature review section, but referenced additional work through the course of the document. Each MQP was rated on its literature review, which also includes previous and current practice, as shown in Table 5.

Table 5: Evaluation of Literature Review

Grade	Percentage
High	6
Good	19
Satisfactory	39
Poor	32
None	3

Comment: The numbers of references are similar to previous years. Students did a bit better in referencing prior work with 25% missing or poor compared to 1997 where 59% were judged in these categories, but had a bit fewer good/high ratings (35%) versus 41% in 1997. This year's review team introduced a "Satisfactory" level in 1999, which explains some of these differences, but overall there is an improvement in deficient literature work sections of projects. However, 25% is still too high.

2.9 Type of Projects

28 (90%) of the projects contained design and implementation of a piece of software. Two of the other projects involved some implementation without significant design and the last project involved design without actual coding of software. 4 (13%) of the projects involved data collection. 13 (42%) of the projects involved evaluating other systems or having the developed system evaluated. 5 (16%) projects involved original research. Two (6%) of the projects involved simulations and two (6%) other projects involved a significant survey component.

Comment: As in previous years a significant number of the projects involved a design component and in most cases implementation of a program.

A number of projects involved integration of a number of software pieces—either through continuation of prior project work or of existing software tools and components. This aspect is an improvement from previous MQP reviews. The reviewers believe that more of the developed systems should be evaluated by other users as part of the project life cycle.

2.10 Project Area

Table 6 shows the percentage of projects that involved different areas of Computer Science. In some cases a project involved only one area while in other cases it involved multiple areas (thus the percentages total to over 100%).

Comment: As the data show there is a variation in the sub-areas of Computer Science covered by the projects, but there is a continued focus on human-computer interaction, networks and software engineering. A number of this year's projects focused on aspects of the World Wide Web as well. Database, graphics, artificial intelligence and operating systems areas were also found in multiple projects.

Table 6: Project Areas by Percentage

42%	Human-Computer Interaction
39%	Networks
26%	Web
23%	Software Engineering (principally part)
19%	Database
16%	Graphics/Visualization/Animation
13%	Artificial Intelligence/Robotics
13%	Operating Systems
10%	Information Management/Retrieval
6%	Distributed Systems
6%	Languages
6%	Multimedia
3%	Theory

2.11 Software Used

Table 7 shows the relative use of different programming languages and other software in the projects. Some projects used more than one software tool (e.g. MS-Windows and C++) thus the percentages total to over 100%.

Comment: The use of the Java programming language continues to increase, particularly using it to develop applets for Web applications. A significant number of projects continue to use C++ with the number of projects using C falling. Scripting languages such as Perl and Tcl showed increase use with Visual Basic also used in few projects developed on PCs.

Table 7: Software Used by Percentage

48%	Java language
45%	C++ language
13%	C language
13%	Perl
13%	Visual Basic
10%	other software packages (Oracle,CORBA,VRML)
6%	HTML
3%	Tcl
3%	no software

2.12 Hardware Used

Table 8 shows the percentage of projects that used different types of hardware platforms for their work. The numbers do not add to 100 since some projects involved different hardware platforms.

Comment: The data show that more projects were done using PCs than workstations, a continuing trend from the last two reviews. Reasons for these results may be the increased personal use of PCs by students and the wide availability of Java and C++ on PCs. The platform independence of such languages made it difficult to determine which platform was used in many cases. Also the increased use of Linux on PCs (at least one project used this environment) clouds the distinction between PCs and workstations.

Table 8: Hardware Used by Percentage

58%	PC
23%	workstation (Digital, Sun)
29%	Unknown

2.13 Computer Science Classes

Early MQP reviews tried to determine the courses contributing to a project, but the 1997 review found this task too difficult. These reviewers left the entry on the MQP review form, but quickly decided the task too difficult as well and will make alternate suggestions for handling this portion of the review.

2.14 Project Evaluations

The numerical evaluations of the projects are shown in Tables 9 and 10 based on the questions from the form in Appendix A. The average and distribution (by percentage) of evaluation for each question is shown. Note: The "stat" level on the Math Level question represents any mathematics between calculus and the senior-level, such as probability and statistics or linear algebra.

Comment: The number of poor abstracts in 1999 (10%) was substantially better than the 34% figure of 1997. However, the number of abstracts better than adequate was only 9% this year compared to 28% in 1997. These differences indicate some improvement, but also the fact that this review team found it difficult to differentiate the quality of relatively short abstracts. Two missing abstracts (6%) are unacceptable.

Projects did about the same as 1997 in stating project objectives, but generally did better in meeting these objectives.

All projects were judged to be at the 3000-level or above in terms of Computer Science knowledge. This is a marked improvement over 1997 when 17% of the projects were judged to be at the 2000-level. The projects exhibited roughly the same (low) mathematics level as in previous years.

The motivation for the projects shows a bit less variation than in the previous year with 52% of the projects judged as adequate. The poorly

Table 9: Project Evaluations by Percentage

	1	**2**	**3**	**4**	**5**	**avg.**
Abstract accurate and complete	missing 6	poor 10	adequate 74	6	excellent 3	2.9
Clearly stated project objective	poor 0	13	adequate 45	42	excellent 0	3.3
Objective met	unk 0	no 10	mostly 29	yes 58	exceeded 3	3.5
CS Level	1000 0	2000 0	3000 13	4000 84	grad 3	3.9
Math Level	none 77	calc 3	stat 19	4000 0	grad 0	1.4
Motivate the project?	poor 3	10	adequate 52	32	excellent 3	3.2
Style, grammar, spelling	poor 0	10	adequate 71	16	excellent 3	3.1
Quality of Tables/ Diagrams/Figures	-	poor 3	adequate 52	32	excellent 6	3.4

Table 10: Project Evaluations by Percentage (cont.)

	1	**2**	**3**	**4**	**5**	**avg.**
Project Methodology	unknown	poor	adequate		excellent	
	6	13	61	19	0	2.9
Issues/Problems Discussed	poor		adequate		excellent	
	3	16	52	19	10	3.2
Overall report organization	poor		adequate		excellent	
	0	6	58	35	0	3.3
Programming Effort	none		some		considerable	
	3	6	45	39	6	3.4
Overall Effort Level (worth 1 unit/student)	too little		about right		too much	
	3	16	58	23	0	3.0
Quality of report	poor		adequate		excellent	
	0	19	55	23	3	3.1
Quality of project	poor		adequate		excellent	
	0	16	35	45	3	3.4
Quality of presentation	unknown	poor	adequate		excellent	
	26	6	29	35	3	3.5

motivated projects often failed to explain why they were implementing a particular application.

Project methodology results show most of the reports as adequate. In general the reports do a satisfactory job in motivating and explaining the context of why the project was done. The reports were less thorough in discussing the design and methodology of how the project was carried out, and in discussing the issues and problems faced in the course of working on the project.

Related to report quality, the style and grammar of the reports was adequate in 71% of the cases. Again there were fewer reports judged less and better than adequate than in 1997. In evaluating the overall quality of the report, the average quality for 1999 was judged to be 3.1 versus 3.2 in 1997. The number of less than adequate reports was reduced in 1999, but there were also fewer reports that the reviewers were willing to rate as better than adequate.

The overall effort of the students appears to be better than in 1997 with also more programming effort than in 1997.

The overall project quality is better than 1995 and 1997 with fewer inadequate projects an more projects rated better than adequate. Fewer projects were rated as excellent, although we believe this is the result of reviewers being more stringent in assigning this rating.

The oral presentation evaluations were done by Prof. David Brown on MQP Presentation Day in April, 1999. Some MQPs finished in the Fall semester for which no evaluations are available. Overall, the large majority of presentations were at least adequate with many of them rated as good.

2.15 Project as a Learning Experience

Almost all of the projects (87%) were rated a good learning experience by the reviewers. The few projects that were not so rated resulted from: too simplistic a computer science component, little rationale for choosing the design of an implementation, lack of consideration of alternatives, not enough effort.

2.16 Project Strengths

Table 11 contains specific reviewer comments extracted from the evaluation forms concerning project strengths.

Comment: As in previous reviews, the projects were good when they were well-motivated, had a clear presentation indicating what was done, had a good design, and followed through on a particular topic.

Table 11: Project Strengths

Tried to utilize new interface technologies not taught in courses.
Off campus. Real life problem.
Tested different approaches.
Went beyond original project design.
Interdisciplinary.
Student learned a lot.
Learned about specific SE techniques.
Learned about CORBA.
Application of CS to a research problem.
Learned about design issues and testing.
Good balance of design issues.
Cares about extensibility.
System integration of a large variety of tools.
Significant programming effort.
Details handled in clever, good ways.
Good use of tools.
Good documentation.
Integrates new work into old.
Thorough testing.
Discussed implementation considerations.
Good user evaluation.
Application and analysis of a real algorithm.

2.17 Project Weaknesses

Table 12 contains specific reviewer comments extracted from the evaluation forms concerning project weaknesses.

Comment: As in previous reviews, projects with problems showed simplistic objectives, poor planning, and poor presentation of what was done.

Table 12: Project Weaknesses

Couldn't adequately test.
Project not well-focused.
No external evaluation.
Technologies not really applied to problem.
Did not clearly motivate particular features.
Odd choice of programming language.
Rather weak purpose.
Basic idea seems odd.
Limited design initiatives.
Rather low-level.
Poor report.
Lack of motivation where design comes from.
No documentation of user study.
Not enough effort.
Scope and methodology mismatch.
Need better writeup of evaluation.
Need more discussion on details.
Simulation partially worked.
Weak analysis.
Details of how accomplished.
Add-on project not focused.
Changing goals led to incomplete project.
Lack of literature review.
Unclear development methodology.

2.18 Interdisciplinary Work

There were eight projects involving other departments disciplines, but no students from other departments was a member of any project team. The disciplines represented by these projects were biology, fire safety, theatre lighting, music, dance and biomedical engineering.

Comment: There were more interdisciplinary projects than 1997, but with no students from other departments.

3 Analysis of Results

This section correlates various aspects of the MQPs with the evaluations the projects received. This analysis is intended to help identify which project characteristics tend to yield good projects and which traits result in lower quality projects.

3.1 Correlation of Evaluations

The following correlations show the relationship between various results and the project evaluations. The project grades and project evaluations are shown for all projects. Note: For sake of comparison the value 4 is assigned to an A project grade, a value 3 to a B project grade and a value 2 is assigned to a C project grade. Recall the project evaluations had a 1 to 5 range where 1 is poor, 2 is fair, 3 is adequate, 4 is good, and 5 is an excellent project. Because of the difference in these scales, the 1997 review team set the standard for correlation as shown in Table 13, suggesting that an A should never be rated less than a 4, a B should receive an evaluation of 2, 3, or 4, and a C should receive a 1 or a 2. Each entry with an "X" shows good correlation.

To start our analysis, we compare the two evaluation criteria taken from the reviewer questionnaire: project grade assigned by the advisor and the project quality (PQ). Table 14 shows the correlation between the project evaluation and the project grade assigned by the advisor. The projects were evaluated before obtaining the project grade.

Comment: There is a disparity between the two evaluation measures for the projects. There are three cases to consider as again defined by the 1997 review team:

Table 13: Expected Correlation Between Project Quality and Grade

	Project Quality				
Grade	1	2	3	4	5
C	X	X			
B		X	X	X	
A				X	X

Table 14: Correlation of Project Grade with Quality of Project

	Project Quality					
Grade	1	2	3	4	5	Total
C	0	0	**3**	0	0	3
B	0	10	13	3	0	26
A	0	**6**	**19**	42	3	71
Total	0	16	35	45	3	100

C1 The adviser and reviewer agree in their assessment of the project.
C2 The adviser graded too harshly or the reviewers overrated the project.
C3 The advisor graded too easily or the reviewer underrated the project.

The results show that while 72% (versus 62% in 1997) of the projects have correlating evaluations (C1), 25% (versus 35% in 1997) fall into case C3, and 3% (versus 3% in 1997) fall into case C2. In Table 14, cases C2 and C3 are represented by **bold** faced entries. These numbers indicate a closer correlation between reviewer rating and project grade than occurred in 1997.

For case C3, the reviewers agree that the quality of the projects is not entirely correlated with the individual grades assigned by the project advisor. 6% of the projects received an A grade although they were assessed to be less than adequate. 19% of the A projects were rated as being adequate, but the A grade should be reserved for those projects that are more than adequate. Either the reviewers did not fully comprehend the significance of the work or the students and advisors agreed upon a less than adequate project. There is continued room for improvement here, and as the number of MQPs in our department grows, the faculty needs to pay attention to standardizing the quality and effort of all MQPs.

For case C2, one project that was rated as adequate received a C grade. This situation most likely reflects the reviewers judging the project better based on its report without being familiar with the process and detailed outcome of the project.

3.2 Correlation of Faculty Team Size and Evaluation

Table 15 shows the correlation between the number of faculty and the project evaluations. The two indicators are report quality (RQ) and project quality (PQ).

Table 15: Correlation of Faculty Team Size and Evaluation

Faculty Team Size	% of Projects	avg Grade	avg RQ	avg PQ
1	74	3.6	3.0	3.1
2+	26	4.0	3.5	4.1

Comment: The data show that single-faculty projects received both lower grades and ratings. These results are consistent with 1995. The data were mixed in the 1997 review.

3.3 Correlation of Student Team Size and Evaluation

Table 16 shows the correlation between the number of students and the project evaluations.

Table 16: Correlation of Student Team Size and Evaluation

Student Team Size	% of Projects	avg Grade	avg RQ	avg PQ
1	35	3.5	3.0	3.3
2	26	3.8	3.2	3.6
3	32	3.7	3.2	3.2
4	6	4.0	2.5	3.5

Comment: As generally found in reviews prior to 1997, single-person projects generally result in the lowest average grade. In 1997, two-person projects had the lowest grade and project quality rating. This year's review had a lower project quality for the three-person projects. The four-person projects rated the least quality when the size of the group is taken into consideration.

3.4 Correlation of On/Off-Campus Projects and Evaluation

Table 17 shows the correlation between projects that were sponsored on/off-campus and the project evaluations.

Comment: There was no difference between projects that were either off-campus or associated with an organization when compared with on-campus projects. In 1997, the off-campus projects received higher evaluations.

Table 17: Correlation of On/Off-Campus Projects and Evaluation

Type	% of Projects	avg Grade	avg RQ	avg PQ
On	84	3.7	3.1	3.3
Off	16	3.6	3.2	3.4

3.5 Correlation of Project Duration and Evaluation

Table 18 shows the correlation between the units earned for a project and the project evaluations.

Table 18: Correlation of Units Earned and Evaluation

Project Units Earned	% of Projects	avg Grade	avg RQ	avg PQ
1	87	3.7	3.1	3.4
1 1/6	13	3.2	3.2	3.2

Comment: Projects that were completed with more than one unit of work typically evaluated lower.

Table 19 shows the correlation between the project duration measured in terms and the project evaluations.

Table 19: Correlation of Project Duration (Terms) and Evaluation

Project Duration in Terms	% of Projects	avg Grade	avg RQ	avg PQ
1-2	26	4.0	2.8	3.5
3	45	3.7	3.4	3.6
4+	26	3.3	3.0	2.9

Comment: The data show the shorter term projects had the highest grades and nearly the highest evaluations. Projects taking longer than three terms had the lowest grades and ratings. shorter term projects distributed have the lowest report quality grade indicating that while the projects are good, the shortened time frame results in poorer reports.

3.6 Correlation of Project Report Size and Evaluation

Table 20 shows the correlation between the project report size and the project evaluations. Note: The report size in Table 20 does not include code and appendices, which in some cases were larger than the report itself.

Table 20: Correlation of Project Report Size and Evaluation

Project Report Size	% of Projects	avg Grade	avg RQ	avg PQ
0–39 pgs.	35	3.6	2.6	3.1
40–69 pgs.	52	3.8	3.2	3.4
70+ pgs.	13	3.5	4.0	3.8

Comment: The results of this correlation show that evaluations correlate to the size of the project report, although that does not hold for the grades assigned. Historically, shorter reports indicate that students did not accomplish much or that they did not allocate enough time to write an adequate report.

3.7 Correlation of Computer Science Level and Evaluation

Table 21 shows the correlation between the Computer Science level and the project evaluations.

Comment: The data show that projects done at the CS 4000 level and higher tend to receive the best evaluations for report quality from the reviewers. The project grade is highly correlated to the CS level, as it should be.

3.8 Correlation of Math Level and Evaluation

Table 22 shows the correlation between the math level and the project evaluations.

Table 21: Correlation of Computer Science Level and Evaluation

Computer Science Level	% of Projects	avg Grade	avg RQ	avg PQ
3000	13	3.5	2.5	2.5
4000	84	3.7	3.2	3.5
grad	3	4.0	3.0	4.0

Table 22: Correlation of Math Level and Evaluation

Math Level	% of Projects	avg Grade	avg RQ	avg PQ
none	77	3.6	3.1	3.2
calc	3	3.0	2.0	2.0
stat	19	4.0	3.3	4.2
4000	3	4.0	4.0	4.0

Comment: Projects that involved some math generally received better grades and evaluations. Part of the reason may be that stronger students are taking on these projects. Another consideration is that the topic requires more effort. Current percentages are roughly the same as previous results.

3.9 Correlation of Overall Effort Level and Evaluation

Table 23 shows the correlation between the overall effort level and the project evaluations.

Comment: As expected, there is a strong correlation.

4 Assessment of Student Learning Outcomes

Although the Computer Science currently does not have as yet a specific set of expected student learning outcomes, the review team felt it important to address learning outcomes as part of this review. Consequently the review team adopted a set of expected learning outcomes from existing materials including the department's own working mission statement draft, CSAB

Table 23: Correlation of Overall Effort Level and Evaluation

Overall Effort Level	% of Projects	avg Grade	avg RQ	avg PQ
1	3	3.0	2.0	2.0
2	16	3.8	2.4	2.4
3	58	3.6	3.2	3.4
4	23	3.9	3.6	4.1
5	0			

accreditation criteria and primarily from ABET outcomes criteria. These adopted outcomes are listed below with an assessment of the degree of success in demonstrating each of these outcomes using the results of the MQP review process.

4.1 Ability to Apply Knowledge of Computer Science and Mathematics

The projects demonstrated student application of computer science at a high level across a variety of computer-science sub-areas. 87% of the projects demonstrated computer science knowledge at least at the 4000-level. The remaining projects were at the 3000-level. The projects demonstrated a low-level of mathematics. 77% of the projects demonstrated no mathematics.

Comment: Application of computer science knowledge does not appear to be a large problem. Application of mathematics is not done well by our students and ways to include it, perhaps in analysis of results, need to be introduced.

4.2 Ability to Design and Implement a Software Solution for a Given Objective

In terms of the design and implementation of a specific solution, 90% of the projects demonstrated this type of work. Students generally did a good job in clearly stating the project objective. Only 13% were judged deficient in this capacity with 42% better than adequate. Project teams were not as good in more than adequately describing the project methodology and issues in attaining the objective. While a majority of projects were judged adequate in these aspects, 19% were less than adequate. 61% of the projects met or exceeded their objective.

Comment: While the results are not poor in this area, students need to be more aware of and better at demonstrating how they defined the objective, the methodology in attaining it, the issues that arose in the course of doing the work and whether the objective was achieved.

4.3 Ability to Evaluate a Software System and Analyze Results

42% of the projects were identified to involve some amount of evaluation as part of the work with some analysis of the results.

Comment: Overall, there were few projects that did a good job of both collecting and analyzing the results of the project. In most projects evaluation was missing altogether and even when present the analysis was weak.

4.4 Ability to Function as Part of a Team

54 of the 65 project students (83%) in the MQP review participated as part of a team in completing the project.

Comment: This figure is relatively high, but a goal of 90% appears reasonable, particularly that more students will be needing MQPs in the near future. More information needs to be gathered from project advisors on the effectiveness of students in these teams.

4.5 Ability to Function Professionally and Ethically

No information is available from the MQP review to measure this outcome.

Comment: The MQP reports used for the review primarily provide information on the end result of the project and not on the specific conduct of project members. Alternate means of obtaining information about student conduct during the project needs to be obtained directly from advisors and external project sponsors.

4.6 Ability to Communicate Effectively

The MQP review judged that 19% of the project reports were less than adequate based on examination of report organization, style, grammar and use of diagrams and figures. 26% of the reports were judged to be better than adequate. 91% of the projects that were evaluated on the oral presentation were at least adequate with 52% of these projects better than adequate.

Comment: These figures are neither particularly good nor bad. They showed that most project reports "were good enough," but more of them should be better than that. Table 19 shows that many of the reduced term MQPs are good in their technical work, but have less time to do a good job in communicating the project. The oral presentations were generally good.

4.7 Ability to Understand Impact of Computer Science in a Societal Context

Fourteen (45%) of the projects were identified as either involving off-campus organizations or involving interdisciplinary work. In addition, three other projects involved work related to education and natural language for a total of 17 (55%) projects judged to work involve work strictly outside of computer science.

Comment: Many other projects had potential impact on society, for example the mainstream use of the World Wide Web, makes these projects have larger societal implications.

4.8 Curriculum Prepares Students for Fundamental Concepts of Discipline

As noted in Section 2.13, determination of the specific courses used for the projects was difficult.

Comment: Alternate means of assessing students specific use of the curriculum needs to be integrated with the review process.

4.9 Ability to Engage in Life-Long Learning

Although not directly measured, virtually all of the projects involved students applying some prior knowledge with the need to learn new concepts and tools in the course of doing the project.

Comment: This outcome needs to be more specifically measured as part of the review process. As part of a wider evaluation of the program in could be measured by surveying alumni.

4.10 Ability to use Modern Computer Science Tools and Techniques

Sections 2.11 and 2.12 indicate that students used a range of tools in the projects.

Comment: Better means of determining the specific software and hardware tools need to be incorporated in the review process. Assessment of techniques used needs to be a separate category in the review.

5 Conclusions and Recommendations

The 1999 review of Computer Science MQPs reflects data and evaluations for 31 MQPs, involving 65 computer science students, that were completed between the Summer of 1998 and the Spring of 1999. In this section, we attempt to draw some conclusions from the data collected during the evaluation process. Although 31 reports do not provide a large set of data points, some conclusions can be drawn from the data collected from the evaluation process.

5.1 Quality of Project

The overall project quality shows that many more projects were judged as at least adequate (84%) in this year's review compared to the previous year with a better correlation between project evaluations given by the reviewers and project grades assigned by the advisor.

Most of the MQPs were good capstone learning experiences for CS majors and meet the educational goals of the department. There was some concern on a few of the projects as good learning experiences. These problematic projects showed little rationale for choosing the design, displayed a lack of consideration for alternatives or indicated the students did not expend enough effort.

81% of the MQPs were judged to involve at least an adequate amount of student effort. Typical Computer Science MQPs include the design and implementation of a large piece of software with many following the software life cycle from requirements gathering to implementation. Unfortunately not enough had results on testing and evaluation of the work. Also, the project methodology was poorly documented so the students could not explain why the students carried out their actions.

5.2 Quality of Report and Abstract

The quality of the reports themselves exhibited a "middling effect" from the previous review where fewer reports were judged to be less than adequate, but fewer reports were also judged to be better than adequate. As in previous MQP reviews, report project areas included project goals not always being clearly stated, and the conclusion chapter occasionally not evaluating how well the original objectives were met. Most of the reports could have been improved by a better literature review or by an explanation of how the MQP fits in with previous work, particularly other MQPs. The most common causes for weaker projects were lack of a clear plan of attack, insufficient work completed by the students, difficulties with the posed problem, underestimating the required work, or inadequate time allocated to writing the report. Not enough time and planning for the report was a problem with both some good and fair projects. In fact, correlation with project duration shows that projects done in less than three terms result in poorer reports, although the project quality does not appear to suffer.

5.3 Students per MQP

The number of single student CS MQPs was rose to 35% (versus 28% in 1997, 44% in 1995 and 42% in 1993). However, the number of average number of students per project increased from 2.0 in 1997 to 2.1 in 1999 due to more larger student teams. This trend is important and shows that the faculty are successfully able to group students together on projects. However, in the next two years, the number of CS senior-level students will more than double, thus there will be increasing pressure to do more multi-student projects to keep faculty project load at a reasonable number. The results show that multi-student projects receive higher evaluations.

5.4 Distribution of CS Faculty over MQPs

There is some improvement in both measures of faculty loading in regards to number of projects advised (Table 2) with the Gini Coefficient dropping from 0.484 in 1997 to 0.406 in 1999. The results in Table 3 show a decrease from 0.554 in 1997 to 0.453 in 1999. We think a value of 0.2-0.28 is a more appropriate measure.

5.5 Project Resources

The project data show a large increase in the use of Java with about as many projects using C++. There was a decrease in use of the C programming language. The availability of Java and C++ for both Windows and Unix platforms resulted in more platform independence for the projects with the particular hardware/Operating System platform not always stated in the project report.

5.6 Faculty Efficiency

Overall, the results of the MQP Review show that faculty were a bit more "efficient" in terms of project advising. Not only was the average faculty project load a bit lower, but faculty did a much better job of advising projects that directly followed previous project work. The number of off-campus projects was slightly reduced with the number of interdisciplinary projects with students and faculty from other departments dropping to zero. With the increased number of students in the coming years, more efficient project advising is essential in accommodating these students.

5.7 Recommendations for the Next CS MQP Review

The evaluation process generally worked reasonably well with a few minor changes of the previous MQP Review form. However, the reviewers identified four areas of the evaluation process that need to be addressed in future reviews:

1. The evaluation of MQP oral presentations needs to be incorporated better in the review process. An evaluation form needs to be created for faculty to fill during the MQP presentations. This form could be used

both for the MQP review process and as feedback for the Outstanding MQP Award.

2. The evaluation of expected student outcomes needs to be better incorporated into the evaluation process. Most of the outcomes in Section 4 could be evaluated with the data from the MQP review, but in a few cases more complete information needs to be gathered. The outcome criteria approved by the department will also likely differ from these criteria adopted for this review.

3. The department needs to involve external (to the department) professionals in the review process. This step was not done as part of this review, but needs to be incorporated as part of a larger department assessment program.

4. The department needs to consider inclusion of a form for students and perhaps advisors to capture information that is difficult to glean from MQP reports themselves. Such information might include:

    - the course numbers and topics that contributed directly to work on the project,
    - what additional material needed to be learned for the project,
    - the hardware and software used for the project,
    - student-perceived project strengths and weaknesses,
    - student-perception of the project as a learning experience,
    - the amount of work on the project as perceived by the students,
    - student-perceived project, report and presentation quality,
    - faculty advisor-perceived project strengths and weaknesses, and
    - the amount of work on the project as perceived by the faculty advisor,

This information could be obtained using a student and/or faculty form separate from the MQP report itself or some of the information could be obtained with a form to be included as an appendix of the MQP report.

An outcome of the MQP review process and this subsequent report will be a presentation by the review team to the department faculty. These recommendations will be brought to the department and any changes in the advising of CS MQPs will be initiated at that time.

5.8 Recommendations for Improving CS MQPs

The following list of recommendations are drawn from the analysis and conclusions of this Computer Science MQP Peer Review.

- Increase student team size and avoid single-student projects when possible. Increasing enrollments in the department make this recommendation a necessity. Better mechanisms for bringing project groups together earlier need to be investigated. Working in project groups improves cooperative and communication skills of the students. Larger MQP teams offer more efficient use of a faculty member's time.

- Use better project planning. The project team (faculty and students) need to do a better job at planning the project and organizing the work. The report should document the planning stage of the project and give a better sense of problems, design considerations, and adjustments in both the direction of the project and work assignments. More emphasis should be given by faculty on expecting formal project proposals.

- Emphasize the testing and evaluation phase. Lack of adequate evaluation by external sources was a problem with many of the design and implementation projects. Serious analysis of projects is a general weakness in department MQPs. More formal analysis would also increase the level of mathematics and statistics displayed by the projects.

- Pay attention to technical writing methodology. Standard technical writing issues such as clear objectives, adequate literature search, report structure, and a thorough review of the project in the conclusion continue to need emphasis when producing a project report.

- Emphasize the need for students to indicate why the MQP was a good experience and what experiences/courses the MQP builds upon. It was difficult with some projects for the reviewers to understand the significance of the work and upon which prior student work the project built upon. One approach for solving this problem is to require students to include an explicit statement stating the courses and materials that the project built upon.

- Allot more time for writing the reports. This recommendation was made in prior reviews and needs to be emphasized again. Many of the shorter reports were from projects that were not as good, although some of the better projects were diminished by reports that were not as high of quality as the project. Part of the problem is that students spend too much time working on the project and not enough time in conveying its significance in the report.

- Strive to have MQPs build on previous MQPs and projects. In industry, our graduates will have to learn how to work with old code from old projects, and one way we can address this is through building upon previous MQPs and theses. This approach makes faculty more efficient and creates a pipeline of projects so the students can see the larger objective for their individual project.

References

[1] About the computer science department.
`http://www.cs.wpi.edu/About/`.

[2] Computing Sciences Accreditation Board.
`http://www.csab.org/`.

[3] Accreditation for Engineering and Technology.
`http://www.abet.org/`.

[4] George T. Heineman and Robert E. Kinicki. 1997 computer science department MQP review. Technical Report WPI-CS-TR-97-07, Worcester Polytechnic Institute, August 1997.

[5] Robert E. Kinicki and Craig E. Wills. Computer science department MQP review. Technical Report WPI-CS-TR-91-13, Worcester Polytechnic Institute, July 1991.

[6] Robert E. Kinicki and Craig E. Wills. 1993 computer science department MQP review. Technical Report WPI-CS-TR-93-5, Worcester Polytechnic Institute, August 1993.

[7] Robert E. Kinicki and Craig E. Wills. 1995 computer science department MQP review. Technical Report WPI-CS-TR-95-1, Worcester Polytechnic Institute, August 1995.

A Review Form

The following form was used to evaluate all MQP reports.

Project Students:____________________Reviewer:________________

1999 Computer Science MQP Review Form

1. Number and department of advisor(s)________________
2. Number, year and department of MQP student(s)________________
3. On/off-campus project and sponsor________________
4. Final grade given to report
5. Terms to complete MQP________________ Units Earned________________
6. Report length in pages (excluding appendices and code)________________
7. Pages of appendices ________________. User manual? Y/N.
8. Quality of literature review? None/Poor/Sat/Good/High. How many references?____________
9. Circle the following types of work and areas of computer science that are relevant for this project.

Analytic	AI	Theory
Data Collection (Empirical)	Architecture	Info Mgmt
Design	DataBase	Webware
Design/Implementation	Graphics	Networks
Evaluation/Testing	HCI	
Research	Languages	
Simulation	Software Engineering	
Survey	Operating Systems	
Other____________________	Distributed Systems	
	Other____________________	

10. Circle the following software languages, tools, and hardware resources used for this project.

C
C++
HTML
Lisp/Scheme
Lisp
X-Windows
Java
Perl/Tcl/Tk
Other____________________

Macintosh
IBM/PC
CCC Unix
CS Unix
Other____________________

on-campus/off-campus?

11. What Computer Science classes were background for this project?

Abstract accurate and complete	1 missing	2 poor	3 adequate	4	5 excellent
Clearly stated project objective	1 poor	2	3 adequate	4	5 excellent
Objective met	1 unknown	2 no	3 mostly	4 yes	5 exceeded
CS Level	1 1000	2 2000	3 3000	4 4000	5 grad
Math Level	1 none	2 calc	3 stat	4 4000	5 grad
Motivate the project?	1 poor	2	3 adequate	4	5 excellent
Style, grammar, spelling	1 poor	2	3 adequate	4	5 excellent
Quality of Tables/ Diagrams/Figures	______ quantity	2 poor	3 adequate	4	5 excellent
Project Methodology	1 unknown	2 poor	3 adequate	4	5 excellent
Issues/Problems Discussed	1 poor	2	3 adequate	4	5 excellent
Overall report organization	1 poor	2	3 adequate	4	5 excellent
Programming Effort	1 none	2	3 some	4	5 considerable
Overall Effort Level (worth one unit/student)	1 too little	2	3 about right	4	5 too much
Quality of report	1 poor	2	3 adequate	4	5 excellent
Quality of project	1 poor	2	3 adequate	4	5 excellent
Quality of presentation	1 unknown	2 poor	3 adequate	4	5 excellent

1. Was this project a good learning experience? What was learned by the student(s)?

2. Was this project a continuation of an earlier project, and if so, did the students indicate the part of the work that is theirs?

3. Project strengths: Project weaknesses:

4. Did this project involve any interdisciplinary work? What departments and subjects were involved?

5. Other comments.