WPI-CS-TR-08-05 August 2008

2008 Computer Science Department MQP Review

by

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 2008 as part of a campus-wide MQP review. The intent of the report is to assess whether the department MQPs are accomplishing their educational goals and fulfilling department-defined student learning outcomes.

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 28 MQPs, involving 52 computer science students, that were completed between the Summer of 2007 and the Spring of 2008. 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. The reviews indicate that 83% of the projects were evaluated with an overall quality of at least adequate with 46% better than adequate. However, these measures are down from similar measures in the past two MQP reviews. The grades for this year's projects are a bit lower than the grades for 2006 projects, although the grades are still higher than most previous reviews.

This report examines these issues as well as drawing a number of conclusions about the success of the projects based upon the data collected and evaluations done for this review. For the first time, this year's MQP review not only includes faculty evaluation of projects relative to department learning outcomes, but also correlates these outcomes with project grades and quality. The report concludes with recommendations for future reviews as the department continues to use 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 (CS) 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 a peer review of MQPs conducted within the Computer Science Department during the Summer of 2008 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 28 MQPs, involving 52 computer science students, that were completed between the Summer of 2007 and the Spring of 2008. This review does *not* include non-CS MQPs (such as for Interactive Media and Game Development (IMGD) and Robotics Engineering (RBE)) advised by CS faculty members. The report makes comparisons to the reviews shown in Table 1.

1.2 Procedure

The peer review was conducted during the Summer of 2008 by Craig E. Wills, a department faculty member. The review was for projects completed during the 2007-08 academic year and was done with a methodology consistent with past MQP review practice. The report for each MQP was obtained from an electronic submission via the Gordon Library E-projects system. Four of the projects reports were unavailable altogether or only contained an executive summary due to sponsor restrictions. Reviewer evaluation of these four projects was not done, but other data about these projects is included in

Table 1: MQPs and Students for All MQP Reviews

Year	Number of CS MQPs	Number of CS Students
1991	19	31
1993	26	44
1995	23	43
1997	29	57
1999	31	65
2001	47	104
2006	44	85
2008	28	52

the overall results. Additional project information was gathered from CDR (Completion of Degree Requirement) records. This review also includes data obtained from advisors on how well an MQP demonstrates each of the CS department's 21 learning outcomes. This inclusion is a new feature of this year's review. Advisor-provided learning outcomes were available for all but three projects.

Unlike the 2006 review process, this year's review process did not include a faculty advisor survey for some aspects of each project. Rather the review team determined that much of the previously obtained data was available as part of the learning outcomes already evaluated by each advisor. Desired information previously obtained from the advisor, but not part of the learning outcomes, was added to the evaluation form used by the review team. Otherwise the peer review team used a similar process from previous years [5, 6, 7, 1, 3, 4, 2]. This approach was used to ensure longitudinal analysis of results with previous years.

The MQP review team filled out the form shown in Appendix A for each project. This form gathered additional information, beyond the advisor-supplied learning outcomes, about the project including an overall assessment of project and report quality. Project grades and registration information was obtained from CDR records. Grades were not consulted until after the MQPs were reviewed.

The single member of the review team evaluated all projects. The reviewer was advisor for one restricted project, which was not reviewed. An evaluation “conflict” existed for another project, but given this was was only one amongst a much larger set the overall tone of the MQP review is not affected. After all evaluations were completed by the reviewer and the faculty learning outcome data obtained, 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 reviewer evaluation form and advisor learning outcomes. Section 3 analyzes and correlates the results. Section 4 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., 1/28 is represented as 4%), and all number averages are represented to one decimal accuracy (i.e., 1.97 is shown as 2.0). Because of this format, the percentages do not always total to 100%.

2.1 Faculty/Student Ratio

Table 2 shows the percentage of projects with the given numbers of students and faculty. No faculty from outside the department were listed as the official project advisor. 52 CS students completed MQPs, working with 4 non-CS students (1 IMGD, 1 MG, 2 ECE).

The average number of students per project was 1.9. The average number of faculty per project (including faculty from other departments that co-advised projects) was 1.3.

Comment: The results show that 8 (29%) of the projects were done by a single CS student, which matches the number of single-student projects observed in recent reviews.

The average number of students per project is down from 2.2 in 2006 and 2.3 in 2001 with the number of three- and four-student team projects falling to 15% from 51% and 40% in recent reviews. The number of projects advised by a single CS faculty member was 24 (86%), an increase from 2001 and 2006.

Table 2: Percentage of projects with the given number of CS students and CS faculty

	Students				
Faculty	**1**	**2**	**3**	**4+**	**Total**
1	21	50	14	0	86
2	7	7	0	0	14
3+	0	0	0	0	0
Total	29	57	14	0	100

2.2 Faculty Project Load

Table 3 shows the distribution on the number of projects (co-)advised by each faculty member. There were 17 full-time faculty in Computer Science during AY07-08 (two faculty were on sabbatical and one was not full-time) plus two affiliated faculty members and two professors of practice who advised projects. Table 4 shows the same data, but with per-advisor weights of 1/2 for projects with two advisors. Note: Loads for co-advisors from other departments are not shown in the tables.

Table 3: Distribution of Projects Advised or Co-advised

Number of Projects (Co-)Advised	Number of Faculty
0	9
1	3
2	4
3	1
4	1
5	3

avg: 1.6 projects/faculty

Table 4: Distribution of Load of Projects Advised

Load of Projects Advised	Number of Faculty
0	9
1	5
1.5	1
2	2
3	1
3.5	1
4.5	1
5	1

avg: 1.3 projects/faculty

Comment: The average project load has decreased to 1.6 from nearly 2.0 in 2006 and 2.8 in 2001. The comparable average loads shown in Table 4 also decreased from previous reviews. These numbers were expected given the shrinking CS undergraduate population.

The Gini Coefficient, a number between 0 and 1 was also calculated for Tables 3 and 4. 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. Table 5 shows comparable figures for all years in which this coefficient has been computed. The results indicate that the MQP project load has become unbalanced in the department. The imbalance for the number of advised projects is at its highest point of all reviews, for which these coefficients were calculated. This imbalance is because of a few faculty, primarily because of project centers, with relatively many projects and almost half of the faculty advising no projects.

2.3 Off-Campus Projects

Fifteen (54%) of the projects were sponsored or involved off-campus companies and organizations. The sponsors were:

NASA Goddard, EBay, NVidia, Microsoft, NASA Ames, Harvard Forest, Lincoln Labs, Bank of America, MITRE, ATR Labs of Kyoto, Japan, and

Table 5: Gini Coefficient for Project Advising Load Amongst Faculty

Year	Number of Projects (Table 3)	Load of Projects (Table 4)
1997	0.48	0.55
1999	0.41	0.45
2001	0.40	0.40
2006	0.56	0.54
2008	0.60	0.62

MTA-SZTAKI in Budapest.

Twelve (43%) of these projects involving 22 (42%) of CS students were done by students off campus.

Comment: The percentage of off-campus sponsored projects represents more than half of the projects done in the department this past year.

2.4 Project Grades

In the projects reviewed, 79% of the projects (77% of the CS students) received a final grade of A, 21% of the projects (23% of the CS students) received a final grade of B, and 0% of the projects (0% of the CS students) received a final grade of C. None of the projects resulted in members on a given project receiving different individual grades.

Comment: The numbers reflect a smaller percentage of A's than in the previous review, but generally higher than previous reviews. It is notable that all projects received at least a grade of B. Table 6 shows the distribution of project grades found during each MQP review.

2.5 Project Continuation

Two projects were continuations of prior MQPs, two projects were part of a larger software project at WPI, one project built on existing work at a project center and one project related to an existing CS graduate research project.

Table 6: Percentage of MQP Project Grades for Each MQP Review

	Project (Student) Grades		
Year	A	B	C
1991	58 (71)	42 (29)	0 (0)
1993	69 (73)	23 (20)	8 (7)
1995	63 (60)	22 (30)	15 (9)
1997	72 (71)	27 (21)	14 (7)
1999	71 (77)	26 (17)	3 (6)
2001	66 (65)	32 (33)	3 (2)
2006	84 (80)	7 (11)	9 (9)
2008	79 (77)	21 (23)	0 (0)

Comment: These numbers are smaller than in past reviews, although with fewer on-campus projects there is less opportunity to integrate with previous work.

2.6 Project Duration

Table 7 shows the duration of each project. The data show a variety of combinations for the number of terms and the amount of unit registration. Note the table shows *registered* units and not *earned* units so that a project needing an extra one-sixth unit to complete the project may not correspond to earned credit.

Comment: The results show that the majority of projects (60%) were completed in three or four terms, which is more than 2006 when 46% used at least three terms. The results also show that no project was registered for more than 1 1/6 units.

2.7 Project Report Size

The average size of the project reports was 44 pages (with a range of 20–86), which excludes appendices and code as well as pages before the body of the report.

Table 7: Percentage of Projects with the Given Duration in Terms of Registered Units

	Units			
Terms	**1**	**1 1/6**	**1 1/2+**	**Total**
1	11	4	0	15
2	19	7	0	26
3	26	4	0	30
4	11	19	0	30
Total	67	33	0	100

Comment: The length of reports is about the same as previous years: 45 (1991), 49 (1993), 50 (1995), 59 (1997), 50 (1999), 58 (2001), and 48 (2006).

2.8 References

The average number of references was 15 (with a range of 5–30) for each report. Results reported in Table 12 show only 4% of the project reports were less than adequate with 41% better than adequate in terms of the quality of the background and literature review.

Comment: These numbers are better than the previous review and more comparable to reviews prior to 2006.

2.9 Type of Projects

Table 8 shows the percentage of projects that involved different types of work. In many cases a project involved only one area while in other cases it involved multiple areas (thus the percentages total to over 100%). Also in past reviews this determination was made by the faculty advisor, but in this review it was made by the review team.

Comment: As in previous years almost all projects involved an implementation and design component. A majority of the projects demonstrated testing of the software. Many fewer projects demonstrated other types of work.

Table 8: Types of Work on Projects by Percentage

96	Implementation
88	Design
54	Testing
17	Analysis
13	Performance Evaluation
9	Research
4	Data Collection (Empirical)
4	Survey
0	Simulation

2.10 Project Area

Table 9 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%). Again this determination was made by the review team where in the past couple of reviews in was made by the faculty advisor.

Comment: There is a variety of Computer Science sub-areas covered by the projects, with Software Engineering and Webware being involved in the most projects. HCI was smaller than the previous reviews as well as systems-related projects.

2.11 Software Used

Table 10 shows the relative use of different programming languages and other software in the projects. It some cases it was not easy for the review team to determine the software used so the reported numbers are likely smaller than actually used.

Comment: The use of the Java programming language is again the highest, up from a value of 42% in 2006. The C++/C# was comparable to C/C++ in 2006, although C was not used for any projects this year.

Table 9: Project Areas by Percentage

33	Software Engineering
33	Webware
17	HCI
17	Networks
13	Graphics
8	Database
8	Algorithms
4	Artificial Intelligence
4	Languages/Compilers

Table 10: Software Used by Percentage

54	Java
21	C++
17	Perl/Python/PHP/TCL/Tk
13	MySQL
8	C#
8	XML
4	Ruby

2.12 Hardware Used

Table 11 shows the percentage of projects that used different types of hardware platforms for their work. The numbers do not add to 100% because it was not possible to determine which platform was used in many cases.

Comment: This project aspect is increasingly less relevant as student projects are being performed at a higher level of abstraction where the underlying machine does not matter.

Table 11: Hardware Used by Percentage

25	PC/Windows
4	PC/Linux

2.13 Reviewer Project Evaluations

Additional numerical evaluations of the projects are shown in Table 12 based on the questions from the form in Appendix A. This form was completed by the MQP review team reviewer.

Comment: When compared with 2006 numbers, each of the average values from Table 12 are up and more comparable to previous reviews.

The average quality of projects is the same as the previous review at 3.5 while the report quality improved from 3.2 to 3.5.

2.14 Project Strengths

Table 13 contains specific reviewer comments extracted from the evaluation form 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.

2.15 Project Weaknesses

Table 14 contains specific reviewer comments extracted from the evaluation form concerning project weaknesses.

Table 12: Reviewer Project Evaluations by Percentage

	1	**2**	**3**	**4**	**5**	**avg.**
Abstract accurate and complete	missing	poor	adequate		excellent	
	0	0	58	38	4	3.5
Clearly stated project objective	poor		adequate		excellent	
	0	0	38	62	0	3.6
Quality of Background/	N/A	poor	adequate		excellent	
Literature Review	0	4	54	33	8	3.5
Style, grammar, spelling	poor		adequate		excellent	
	0	0	62	33	4	3.4
Project Methodology	unknown	poor	adequate		excellent	
Issues/Problems Discussed	0	12	38	46	4	3.4
Quality of report	poor		adequate		excellent	
	0	4	50	42	4	3.5
Quality of project	poor		adequate		excellent	
	0	17	38	29	17	3.5

Table 13: Project Strengths

work done in a research lab
solve a real problem
part of a bigger problem
long-term impact
real company, addresses real need
contribution to existing project
integration of tools
accomplished a lot
ambitious project goals

Table 14: Project Weaknesses

did not complete the intended project no motivation
depth of project not clear
relation to previous project not clear
lack of evaluation
no testing
virtually no implementation
lack of explanation of project details

Comment: As in previous reviews, projects with problems showed simplistic objectives, poor planning, and poor presentation of what was done. The most common problem were issues with the evaluation and testing portion of the projects.

2.16 Interdisciplinary Work

There were three projects involving other majors (Music, ECE and IMGD).

2.17 Advisor Evaluation of Project Learning Outcomes

Numerical evaluations of the project outcomes are shown in Tables 15 and 16 based on the online form filled out by faculty project advisors. The tables reflect the results for the 25 projects in which the form was completed.

Comment: The use of the information is new to the 2008 MQP review. In the 2001 and 2006 reviews, a separate advisor evaluation form seeking additional information about each project was used. This is the first CS MQP review to combine the project outcomes results with reviewer results. This approach is an improvement in the process so that faculty advisors only need to complete one assessment form for each advised project. Results from correlations of these outcomes with project quality are shown in Section 3.7.

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

Table 15: Advisor Evaluation of Project Learning Outcomes by Percentage

Learning Outcome	**1 Not At All**	**2 Some-what**	**3 Well**	**4 Excel-lently**	**avg.**
Demonstrated an understanding of programming language concepts.	20	24	24	32	2.7
Demonstrated knowledge of computer organization.	40	36	12	12	2.0
Demonstrated an ability to analyze the behavior of computational systems.	24	28	40	8	2.3
Demonstrated knowledge of computer operating systems.	16	44	32	8	2.3
Demonstrated an understanding of the Foundations of Computer Science.	56	36	8	0	1.5
Demonstrated an understanding of Software Engineering principles and the ability to apply them to software design.	0	12	40	48	3.4
Demonstrated an understanding of Human-Computer Interaction.	8	36	36	20	2.7
Completed a large software project.	0	12	36	52	3.4
Demonstrated advanced knowledge of Computer Science topics.	0	25	50	25	3.0
Demonstrated an understanding of the mathematical foundations for computer science.	68	20	12	0	1.4
Demonstrated knowledge of probability or statistics.	72	20	8	0	1.4

Table 16: Advisor Evaluation of More Project Learning Outcomes by Percentage

Learning Outcome	**1 Not At All**	**2 Some-what**	**3 Well**	**4 Excel-lently**	**avg.**
Demonstrated an understanding of scientific principles, and the scientific method.	56	32	8	4	1.6
Demonstrated the ability to design experiments and interpret experimental data.	56	24	16	4	1.7
Demonstrated independent learning.	0	8	32	60	3.5
Demonstrated the ability to locate and use technical information from multiple sources.	0	12	28	60	3.5
Demonstrated an understanding of professional ethics.	28	40	28	4	2.1
Demonstrated an understanding of the links between technology and society.	44	28	28	0	1.8
Belong to at least one professional organization, including IEEE, ACM, and UPE.	95	0	0	5	1.1
Participated in a class or project team.	16	0	28	56	3.2
Demonstrated the ability to communicate effectively in speech.	0	12	40	48	3.4
Demonstrated the ability to communicate effectively in writing.	0	4	60	36	3.3

Table 17: 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

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 17, 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. Table 18 shows the correlation between the project evaluation and the project grade assigned by the advisor. The projects were evaluated before obtaining the project grade. Results are shown for the 24 projects that were evaluated by the review team.

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:

C1 The adviser and reviewer agree in their assessment of the project.

Table 18: Correlation of Project Grade with Quality of Project by percentage

	Project Quality					
Grade	1	2	3	4	5	Total
C	0	0	0	0	0	0
B	0	8	13	4	0	25
A	0	**8**	**25**	25	17	75
Total	0	17	38	29	17	100

C2 The adviser graded too harshly or the review team overrated the project.

C3 The advisor graded too easily or the review team underrated the project.

The results show that while 67% (versus 63% in 2006) of the projects have correlating evaluations (C1), 33% (versus 34% in 2006) fall into case C3, and 0% (versus 5% in 2006) fall into case C2. In Table 18, case C3 is represented by **bold** faced entries in the lower left portion of the table. These numbers indicate that the reviewers again believe there are projects receiving a higher grade than the work deserves.

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. Two projects (8%) received an A grade although they were assessed to be less than adequate. 25% 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 review team 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 an increasing number of our projects are completed at off-campus project centers, the faculty needs to pay attention to standardizing the quality and effort of all MQPs.

3.2 Correlation of Faculty Team Size and Evaluation

Table 19 shows the correlation between the number of faculty and the project evaluations. The two indicators are report quality (RQ) and project quality (PQ).

Comment: There is some difference in the results, but relatively few projects were advised by more than one faculty member so inferring too

Table 19: Correlation of Faculty Team Size and Evaluation

Faculty Team Size	% of Projects	avg Grade	avg RQ	avg PQ
1	83	3.7	3.4	3.4
2+	17	4.0	4.0	4.0

much significance from the difference is difficult.

3.3 Correlation of Student Team Size and Evaluation

Table 20 shows the correlation between the number of students and the project evaluations.

Table 20: Correlation of Student Team Size and Evaluation

Student Team Size	% of Projects	avg Grade	avg RQ	avg PQ
1	29	3.7	3.3	3.4
2	54	3.8	3.6	3.6
3	17	3.5	3.2	3.0

Comment: Again there is not a significant difference based on the project team size although the larger teams had the worst performance in this review. Two-person project teams showed the worst performance in 2006, but the best here.

3.4 Correlation of On/Off-Campus Projects and Evaluation

Table 21 shows the correlation between projects that were sponsored on/off-campus and the project evaluations. The 33% figure for projects completed off campus is smaller than given in Section 2.3) because not all off-campus projects could be evaluated for this correlation.

Comment: As in the 2006 review, the off-campus projects for this review appear to do better than on-campus projects in all categories. Most

Table 21: Correlation of On/Off-Campus Projects and Evaluation

Type	% of Projects	avg Grade	avg RQ	avg PQ
On	67	3.6	3.2	3.1
Off	33	4.0	3.9	4.1

notably, the report quality for off-campus projects was 0.7 points higher, and the project quality was 1.0 points higher. We expect that the added rigor of having an external sponsor helps motivate students to perform at a higher level. Often students also have to interview for off-campus projects so stronger students may be selected.

3.5 Correlation of Project Duration and Evaluation

Table 22 shows the correlation between the registered units for a project and the project evaluations.

Table 22: Correlation of Registered Units and Evaluation

Registered Project Units	% of Projects	avg Grade	avg RQ	avg PQ
1	67	3.8	3.6	3.7
1 1/6	33	3.8	3.1	3.0

Comment: In the past, projects that were completed with more than one unit of work typically evaluated lower. This trend continues here. Note that a high percentage of projects are completed in just one unit of work.

3.6 Correlation of Project Report Size and Evaluation

Table 23 shows the correlation between the project report size and the project evaluations. The report size does not include code and appendices.

Comment: The results of this correlation show that the quality of both the report and project track with the size of the project report. This result has generally been the case in previous reviews as shorter reports indicate

Table 23: Correlation of Project Report Size and Evaluation

Project Report Size	% of Projects	avg Grade	avg RQ	avg PQ
0–39 pgs.	50	3.8	3.0	2.8
40–69 pgs.	38	3.8	3.8	4.0
70+ pgs.	12	3.7	4.3	4.3

that students did not accomplish much or that they did not allocate enough time to write an adequate report.

3.7 Correlation of Project Learning Outcomes and Evaluation

Tables 24-43 show the correlation between the advisor evaluation of each learning out (save for professional organization) and the project evaluations. Because this is the first review for which advisor-supplied outcome evaluations are available, correlation with report evaluations is shown for all so as to better understand which outcomes lead to better projects. Note that these correlations represent the 21 projects for which both advisor and reviewer assessments are available.

Table 24: Correlation of Evaluation and Demonstrated an Understanding of Programming Language Concepts

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	19	3.5	3.2	3.2
Somewhat	29	3.8	3.5	3.3
Well	24	3.6	3.6	3.8
Excellently	29	4.0	3.2	3.2

Table 25: Correlation of Evaluation and Demonstrated Knowledge of Computer Organization

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	38	3.6	3.1	2.9
Somewhat	38	3.9	3.6	3.5
Well	10	3.5	3.5	3.5
Excellently	14	4.0	3.3	4.3

Table 26: Correlation of Evaluation and Demonstrated an Ability to Analyze the Behavior Of Computational Systems

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	29	3.7	3.3	3.0
Somewhat	29	3.7	3.0	2.8
Well	33	3.9	3.6	3.7
Excellently	10	4.0	4.0	5.0

Table 27: Correlation of Evaluation and Demonstrated Knowledge of Computer Operating Systems

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	19	3.5	3.2	3.0
Somewhat	48	3.7	3.3	3.2
Well	29	4.0	3.5	3.7
Excellently	5	4.0	4.0	5.0

Table 28: Correlation of Evaluation and Demonstrated an Understanding of the Foundations of Computer Science

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	52	3.7	3.4	3.2
Somewhat	38	3.9	3.4	3.6
Well	10	3.5	3.5	3.5
Excellently	0	—	—	—

Table 29: Correlation of Evaluation and Demonstrated an Understanding of Software Engineering Principles and the Ability To Apply Them To Software Design

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	0	—	—	—
Somewhat	10	3.5	3.0	3.0
Well	48	3.7	3.5	3.5
Excellently	43	3.9	3.3	3.3

Table 30: Correlation of Evaluation and Demonstrated an Understanding of Human-Computer Interaction

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	10	4.0	4.0	4.0
Somewhat	38	4.0	3.4	3.5
Well	33	3.4	3.3	3.0
Excellently	19	3.8	3.2	3.5

Table 31: Correlation of Evaluation and Completed a Large Software Project

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	0	—	—	—
Somewhat	14	3.3	3.7	3.7
Well	43	3.7	3.3	3.0
Excellently	43	4.0	3.3	3.7

Table 32: Correlation of Evaluation and Demonstrated Advanced Knowledge of Computer Science Topics

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	0	–	–	–
Somewhat	30	3.7	3.3	3.0
Well	50	3.7	3.1	3.0
Excellently	20	4.0	4.0	4.8

Table 33: Correlation of Evaluation and Demonstrated an Understanding of the Mathematical Foundations For Computer Science

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	71	3.7	3.3	3.2
Somewhat	14	3.7	4.0	4.3
Well	14	4.0	3.3	3.3
Excellently	0	—	—	—

Table 34: Correlation of Evaluation and Demonstrated Knowledge of Probability Or Statistics

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	81	3.8	3.2	3.2
Somewhat	14	3.7	4.0	4.3
Well	5	3.0	4.0	3.0
Excellently	0	—	—	—

Table 35: Correlation of Evaluation and Demonstrated an Understanding of Scientific Principles, and the Scientific Method

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	52	3.8	3.1	3.0
Somewhat	33	3.7	3.9	4.1
Well	10	3.5	3.0	2.5
Excellently	5	4.0	4.0	4.0

Table 36: Correlation of Evaluation and Demonstrated the Ability to Design Experiments and Interpret Experimental Data

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	52	3.8	3.1	2.9
Somewhat	24	4.0	3.8	4.6
Well	19	3.5	3.5	3.2
Excellently	5	3.0	4.0	3.0

Table 37: Correlation of Evaluation and Demonstrated Independent Learning

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	0	—	—	—
Somewhat	10	3.0	3.0	2.0
Well	33	3.7	3.6	3.4
Excellently	57	3.9	3.3	3.6

Table 38: Correlation of Evaluation and Demonstrated the Ability To Locate And Use Technical Information From Multiple Sources

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	0	—	—	—
Somewhat	14	3.3	3.3	2.7
Well	33	3.6	3.4	3.3
Excellently	52	4.0	3.4	3.6

Table 39: Correlation of Evaluation and Demonstrated an Understanding of Professional Ethics

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	29	3.7	3.3	3.2
Somewhat	48	3.9	3.3	3.4
Well	19	3.5	3.8	3.8
Excellently	5	4.0	3.0	3.0

Table 40: Correlation of Evaluation and Demonstrated an Understanding of the Links Between Technology and Society

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	52	3.8	3.4	3.2
Somewhat	24	4.0	3.6	4.0
Well	24	3.4	3.2	3.2
Excellently	0	—	—	—

Table 41: Correlation of Evaluation and Participated In a Class Or Project Team

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	19	3.8	3.5	3.8
Somewhat	0	—	—	—
Well	33	3.6	3.1	2.9
Excellently	48	3.9	3.5	3.6

Comment: The results of the correlations between learning outcomes and project quality measures varies. Some learning outcomes show clear positive correlations with project quality. These outcomes include: ability to analyze the behavior of computational systems, knowledge of operating systems, advanced knowledge of computer science topics, demonstration of independent learning, ability to locate and use technical information from multiple sources, ability to communicate in speech and the ability to communicate in writing. Apart from perhaps operating systems, these outcomes relate to core aspects of all successful projects and the results show that students who do them better produce the best projects.

Other learning outcomes have no, or at best weak, correlation with project quality. Some of these outcomes are at least well represented in a majority of projects, but do not necessarily translate into higher project quality. These outcomes include: understanding of programming language concepts, understanding of software engineering principles and design, understanding

Table 42: Correlation of Evaluation and Demonstrated the Ability to Communicate Effectively In Speech

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	0	—	—	—
Somewhat	14	3.3	3.0	2.0
Well	43	3.8	3.2	3.2
Excellently	43	3.9	3.7	4.0

Table 43: Correlation of Evaluation and Demonstrated the Ability to Communicate Effectively In Writing

Advisor Eval	(%) of Projects	avg Grade	avg RQ	avg PQ
Not at All	0	—	—	—
Somewhat	5	3.0	3.0	2.0
Well	71	3.8	3.3	3.1
Excellently	24	3.8	3.8	4.4

of human-computer interaction, completion of a large software project, and participation in a project team.

4 Conclusions and Recommendations

The 2008 review of Computer Science MQPs reflects data and evaluations for 28 MQPs, involving 52 computer science students, that were completed between the Summer of 2007 and the Spring of 2008. In this section, we attempt to draw some conclusions from the data collected during the evaluation process.

4.1 Quality of Project

The overall project quality shows that fewer projects were judged as at least adequate (83%) in this year's review compared to the previous two reviews of 88% in 2006 and 92% in 2001. The grades for this year's projects are a bit lower than the grades for 2006 projects, although the grades are still higher than most previous reviews. Again the difference between project grade and evaluation was most apparent for projects done on campus.

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.

4.2 Quality of Report

The overall quality of the reports themselves was better than in 2006 and comparable to reviews done before that time. All aspects of the report were judged to improve from the 2006 review.

4.3 Students per MQP

The number of single student CS MQPs was 29%, which is similar to previous MQP reviews. The average number of students per project dropped to 1.6 from 2.0 in 2006. These numbers reflect fewer undergraduate students. On the order of 30% single-student projects seems to be the norm for CS MQPs.

4.4 Distribution of CS Faculty over MQPs

After the 2006 review found the highest concentration of MQPs amongst a relatively small number of faculty, this 2008 review an even higher concentration. These results yielded in the highest Gini coefficient for the years it has been computed. Part of this result is because of off-campus project centers, but it also indicates some amount of specialization of faculty in terms of where they are expending effort. There is also the aspect that not all faculty areas are equally attractive to all students.

4.5 Off-Campus Projects

The sponsorship of projects by off-campus companies and organizations increased dramatically to 54% of projects up from 43% in 2006 and 23% in 2001. These projects also received much higher evaluations—both by grade and evaluation—than projects done on campus.

4.6 Project Resources

The project data show that the use of Java for projects was over 50% with a variety of other languages getting smaller amounts. The hardware resource usage indicates clearly that students are less concerned about the particular hardware environment that is used compared to previous years.

4.7 Learning Outcomes

The results show that demonstration of department learning outcomes is mixed amongst the projects reviewed. Correlation of learning outcome representation with project quality identifies a number of core outcomes of high quality projects for which good work on these outcomes translates directly into successful projects. These outcomes include: ability to analyze the behavior of computational systems, advanced knowledge of computer science topics, demonstration of independent learning, ability to locate and use technical information from multiple sources, ability to communicate in speech and the ability to communicate in writing.

4.8 Recommendations for the Next CS MQP Review

The evaluation process generally worked well. Extending the process to include online faculty advisor evaluations of learning outcomes is a positive development. This approach both allows the outcomes to be explicitly included in the evaluation as well as having only one evaluation form for advisors to fill out.

One aspect that could be improved with the online form is to gather information best provided by advisors such as the type of work on projects, project area as well as software and hardware resources used.

There are two other ongoing recommendations from previous reviews. The first includes including evaluations of MQP presentations. While desirable to consider this aspect of the projects, there are difficulties in relating the volume of data gathered with individual project.

The second ongoing recommendation is for the department to consider involving external (to the department) professionals, such as alumni, in the review process. Again this is desirable, but has not happened.

References

[1] 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.

[2] George T. Heineman and Craig E. Wills. 2006 computer science department MQP review. Technical Report WPI-CS-TR-06-15, Worcester Polytechnic Institute, August 2006.

[3] Micha Hofri and Craig E. Wills. 1999 computer science department MQP review. Technical Report WPI-CS-TR-99-24, Worcester Polytechnic Institute, August 1999.

[4] Micha Hofri and Craig E. Wills. 2001 computer science department MQP review. Technical Report WPI-CS-TR-01-18, Worcester Polytechnic Institute, August 2001.

[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 Reviewer-Only Form

The following form was used by the review team to evaluate MQP projects.

1. Number and department of MQP student(s)__________________
2. Final grade given to report
3. Terms to complete MQP__________________ Units Earned__________________
4. Report length in pages (excluding appendices and code)__________________

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
Quality of Background/ Literature Review	______ no. refs	2 poor	3 adequate	4	5 excellent
Style, grammar, spelling	1 poor	2	3 adequate	4	5 excellent
Project Methodology Issues/Problems Discussed	1 unknown	2 poor	3 adequate	4	5 excellent
Quality of report	1 poor	2	3 adequate	4	5 excellent
Quality of project	1 poor	2	3 adequate	4	5 excellent
Project Objective met	1 unknown	2 no	3 mostly	4 yes	5 exceeded
Overall Effort Level (worth one unit/student)	1 too little	2	3 about right	4	5 too much

1. Project strengths: Project weaknesses:

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. Did this project involve any interdisciplinary work? What departments or organizations were involved? Off-campus or on.

4. Circle the following types of work and areas of computer science that are relevant for this project.

Analysis	AI	Theory/Foundations
Data Collection (Empirical)	Architecture	Networks
Design	DataBase	Webware
Implementation	Graphics	
Performance Evaluation	HCI	
Research	Languages/Compilers	
Simulation	Software Engineering	
Survey	Operating Systems	
Testing	Distributed Systems	
Other________________	Other________________	

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

C	Macintosh
C++	PC/Windows
Assembly Lang.	PC/Linux
Lisp/Scheme	CS Unix
Java	CCC Unix
Perl/Python/PHP/Tcl/Tk	Other________________
Ruby	
Other________________	