WPI-CS-TR-06-15 August 2006

2006 Computer Science Department MQP Review

by

George T. Heineman
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 2006 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.

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 44 MQPs, involving 85 computer science students, that were completed between the Summer of 2005 and the Spring of 2006. 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 88% of the projects were evaluated with an overall quality of at least adequate with 47% better than adequate. However, these measures are down from similar measures in the 2001 MQP Review. In contrast, the grades for this year's projects are significantly higher than the grades for the 2001 projects.

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. 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 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 2006 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 44 MQPs, involving 85 computer science students, that were completed between the Summer of 2005 and the Spring of 2006. 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
2001	47	104
2006	44	85

1.2 Procedure

The peer review was conducted during the Summer of 2006 by George T. Heineman and Craig E. Wills, each an associate professor. The review was to be for projects completed during the 2005-06 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, the project advisor or from the Gordon Library. Additional project information was gathered from CDR (Completion of Degree Requirement) records.

Based on records from the WPI registrar, there were 52 registered MQPs during the 2005-2006 academic year. Three were removed from consideration because the work had been completed before the 2005-2006 academic year, and the students on these projects had registered for minimal credit only to deposit the project with the WPI registrar. Five MQPs were not completed. Thus we reviewed a total of 44 projects.

As in the 2001 review process, this review process included faculty advisor input on some aspects of each project. The peer review team used a similar process from previous years [4, 5, 6, 1, 2, 3]. This approach was used to ensure longitudinal analysis of results with previous years.

The project review form in Appendix A was completed by the faculty advisor of the project. In past reviews, the MQP review team also completed these forms; the review team felt there was no added benefit to completing the form. 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. 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. A project evaluation form was received back from a faculty advisor for most projects reviewed.

The MQP review team reviewers filled out a one page form shown in Appendix B. This form gathered additional information 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 MQP reports were divided between the two reviewers for evaluation. After all evaluations were completed by the reviewers and the faculty advisors, 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 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., 1/44 is represented as 2%), 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 1 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. 85 CS students completed MQPs, working with 10 non-CS students (3 IE, 2 MG, 1 MGE, 2 ECE, 1 MAC, 1 MGE).

The average number of students per project was 2.2. The average number of faculty per project (including faculty from other departments that co-advised projects) was 1.2.

Comment: The results show that 13 (30%) of the projects were done by a single student, which matches the number of single-student projects observed in the 2001 MQP review.

The average number of students per project is down only slightly to 2.2 from 2.3 in 2001 with the number of three- and four-student team projects falling from 51% to 40%. The number of projects advised by a single faculty member was 37 (84%), an increase from 2001.

2.2 Faculty Project Load

Table 2 shows the distribution on the number of projects (co-)advised by each faculty member. There were 22 full-time faculty in Computer Science during AY05-06 (one faculty was on sabbatical) plus one associated faculty member and two professors of practice who advised projects. Table 3 shows the same data, but with per-advisor weights of 1/2 or 1/3 for projects with

Table 1: Percentage of projects with the given number of students and faculty

	Students				
Faculty	**1**	**2**	**3**	**4+**	**Total**
1	27	23	34	0	84
2	2	7	2	2	14
3+	0	0	0	2	2
Total	30	30	36	5	100

two and three advisors. Note: Loads for co-advisors from other departments are not shown in the tables.

Table 2: Distribution of Projects Advised or Co-advised

Number of Projects (Co-)Advised	Number of Faculty
0	5
1	8
2	4
3	1
4	1
5	1
6	1
7	1
>8	1

avg: 2 projects/faculty

Comment: The average project load has decreased to nearly 2 from 2.8 in 2001. The comparable average loads shown in Table 3 decreased to 2.1 projects/faculty from 2.3 projects/faculty in 1999. These numbers were expected given the shrinking CS undergraduate population.

The Gini Coefficient, a number between 0 and 1 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

Table 3: Distribution of Load of Projects Advised

Load of Projects Advised	Number of Faculty
0	4
0.5	1
1	8
2	3
3	2
4	1
5	1
6	1
10	1

avg: 2.1 projects/faculty

single faculty member. Table 4 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 in the past four reviews, for which these coefficients were calculated.

Table 4: Gini Coefficient for Project Advising Load Amongst Faculty

Year	Number of Projects (Table 2)	Load of Projects (Table 3)
1997	0.484	0.554
1999	0.406	0.453
2001	0.402	0.399
2006	0.560	0.544

2.3 Off-Campus Projects

Nineteen (43%) of the projects were sponsored or involved off-campus companies and organizations. The sponsors were:

Autonomous Undersea Systems Institute, Goddard/NASA, Lehman Bros., JP Morgan, Lincoln Labs, Deutsche Bank, Morgan Stanley, EBay, NVidia, SRI, EMC, Harmony Line, General Dynamics, MTA-SZTAKI, and Cisco.

Comment: The number of off-campus sponsored projects (16 of 44) shows a continued increase.

2.4 Project Grades

In the projects reviewed, 84% of the projects (82% of the CS students) received a final grade of A, 7% of the projects (10% of the CS students) received a final grade of B, and 9% of the projects (8% of the CS students) received a final grade of C.

Four projects resulted in members on a given project receiving different individual grades. In one two-student project, one student received an A while the other student received an NR grade; for the purpose of this review, this project was classified as an A-grade project. In the other three split-grade projects, the closest letter grade of the average of the individual student grades was used as the project grade. For example, a two-student project with grades of A and B (ave. of 3.5) was rated as an A project, while a three-student project with grades C, B, and C (ave. of 2.33) was rated as a C project.

Comment:

The numbers reflect the highest percentage of A's across all MQP reviews. The number of C grades showed a slight increase. Table 5 shows the distribution of project grades found during each MQP review.

2.5 Project Continuation

Twelve projects (27%) were continuations of prior MQPs, three projects extended existing CS graduate research projects (either MS or PhD), and three projects depended heavily on external non-WPI projects (available freely on the Internet).

Comment: These numbers are consistent with the 1999 and 2001 reviews. In 1995 and 1997 only 14% and 7% of the projects were continuations. The results indicate that faculty have continued to integrate new projects with previous work.

Table 5: Percentage of MQP Project Grades for Each MQP Review

Year	Project (Student) Grades		
	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)

2.6 Project Duration

Table 6 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 project centers (Wall Street and Silicon Valley) have lifted B and C terms to contain the highest percentage of project work. 70% of the projects were completed in a single unit worth of work, which is comparable to 1997, but lower than the high of 87% in 1999. Many of the off-campus projects require either a PQP or a 1/6th follow-up credit after the project, which decreases the number of projects that complete in a single unit.

2.7 Project Report Size

The average size of the project reports was 48 pages (with a range of 13–221), which excludes appendices and code.

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

Table 6: Percentage of Projects with the Given Duration in Terms and Registered Units

	Units			
Terms	**1**	**1 1/6**	**1 1/2+**	**Total**
1	16	7	1	24
2	20	8	2	31
3	21	5	2	28
4	14	3	1	18
Total	70	23	7	100

2.8 References

The average number of references was 11 (with a range of 0–34) for each report. Results reported in Table 12 show 25% of the project reports were less than adequate with 27% better than adequate in terms of the quality of the background and literature review.

Comment: This number is down slightly from past years. Without question, there should be no projects with zero references. In this review, two projects had no references, while a total of five projects had two or fewer references.

2.9 Type of Projects

Table 7 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%).

Comment: As in previous years a significant number of the projects involved a design component and in most cases implementation of a program (although the 70% is down from 87% in 2001). The number of projects involving design and implementation of a piece of software is comparable to previous reviews.

Table 7: Types of Work on Projects by Percentage

70	Design/Implementation
44	Design
40	Analysis
33	Performance Evaluation
33	Research
19	Data Collection (Empirical)
7	Simulation

2.10 Project Area

Table 8 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: There is a variety of Computer Science sub-areas covered by the projects, with Human-Computer Interaction and Software Engineering being involved in an increasing number of projects.

Table 8: Project Areas by Percentage

42	Software Engineering
26	HCI
21	Networks
14	Webware
12	Database
9	Artificial Intelligence
9	Distributed Systems
7	Languages/Compilers
7	Operating Systems
5	Graphics

2.11 Software Used

Table 9 shows the relative use of different programming languages and other software in the projects. Not all project reviews included answers to these questions, so the reported numbers seem smaller than they likely are.

Comment: The use of the Java programming language is the highest, but dropped from a value of 48% in 1999. The C++ language showed the largest drop (from 30% in 2001). Visual Basic was not used on any projects (down from 13% in 1999 to 4% in 2001).

Table 9: Software Used by Percentage

42	Java
14	C
12	Perl/Python/PHP/TCL/Tk
9	C++

2.12 Hardware Used

Table 10 shows the percentage of projects that used different types of hardware platforms for their work. The numbers do not add to 100% because, once again, the surveyed responses did not include this information.

Comment: Just over half of the projects actually reported the machines on which they executed. The trend is clear, namely, that students are less and less concerned with the actual machines being used. This is a sign of, perhaps, the open source movement, but also that student projects are being performed at a higher level of abstraction where the underlying machine does not matter.

Table 10: Hardware Used by Percentage

37	PC/Windows
21	PC/Linux
7	CCC Unix

2.13 Advisor Project Evaluations

Numerical evaluations of the projects are shown in Table 11 based on the questions from the form in Appendix A. This form was completed by the faculty advisor, and we received 36 out of 44 surveys. Thus, in this section, all percentages are based on 36 projects. Because we only have a subset of respondents, we suggest that the reported average numbers may actually be higher because it is possible that some faculty may have chosen (from indifference) not to complete and submit the brief MQP evaluation form. Specifically, of the eight unrated projects (for 16 students), 7 project teams received an A, and the eighth team received a split assessment of A/B/A.

Table 11: Advisor Evaluations by Percentage

	1	**2**	**3**	**4**	**5**	**avg.**
CS Level	1000	2000	3000	4000	grad	
Demonstrated	0	3	25	64	8	3.8
Math Level	none	lin alg	prob/stat	4000	grad	
Demonstrated	53	6	31	11	0	2.0
Demonstrated	poor		adequate		excellent	
Professional Ethics	3	3	36	25	22	3.3
Software Project Size/	none		moderate		large-scale	
Programming Effort	3	11	36	42	8	3.4
Learned New	none		some		considerable	
Tools/Techniques/Info	0	6	28	22	44	4.1
Project Objective met	unknown	no	mostly	yes	exceeded	
	0	8	22	44	25	3.9
Overall Effort Level	too little		about right		too much	
(worth one unit/student)	8	11	42	36	3	3.1

Comment: In the 2001 MQP assessment, the questions from Table 11 were answered by both MQP advisors and the MQP review team. This was done to try to validate the overall MQP review process, by showing correlations between the independent evaluations and the advisor reviews. During this review, the MQP review team decided that there was no substantial benefit to duplicating the questionnaire.

There were some declines. The demonstrated math dropped to 2.0 (which is still higher than the 1.6 score witnessed in 1997) and the demonstrated professional ethics dropped from 3.7 to 3.3. Project objectives showed a noticeable increase to 3.9 (up from 3.6) and is at its highest level in four reviews. Other metrics remained the same.

2.14 Reviewer Project Evaluations

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

Comment: When compared with last year's numbers, each of the average values from Table 12 is down from the last review in 2001. The highest dropoff is present in the quality of background/literature review (down from 3.6 in 2001).

One observation that we make is that the quality of the project reports dropped quite a bit with the percentage of better-than-adequate reports dropping from 57% in 2001 to 34% this year. Similarly, the percentage of less-than-adequate reports increased from 13% in 2001 to 19% this year. The average quality of the report dropped from 3.6 in 2001 to 3.2 this year. The quality of the projects dipped a bit from an average of 3.6 in 2001 to 3.5 this year.

2.15 Project Strengths

Table 13 contains specific advisor and 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.

One project had no strengths listed.

Table 12: Reviewer Project Evaluations by Percentage

	1	2	3	4	5	avg.
Abstract accurate and complete	missing	poor	adequate		excellent	
	2	16	48	27	2	3.1
Clearly stated project objective	poor		adequate		excellent	
	5	7	55	25	9	3.3
Quality of Background/	N/A	poor	adequate		excellent	
Literature Review	0	25	43	25	2	3.0
Style, grammar, spelling	poor		adequate		excellent	
	0	9	52	30	7	3.3
Project Methodology	unknown	poor	adequate		excellent	
Issues/Problems Discussed	7	16	32	32	11	3.3
Quality of report	poor		adequate		excellent	
	5	14	45	25	9	3.2
Quality of project	poor		adequate		excellent	
	5	7	36	36	11	3.5

Table 13: Project Strengths

Interesting problem
independent work, simulator construction, evaluation
group work with NASA and SEU
complete software development cycle
analysis
real-world experience
Difficult project, done through work
solid plan and performance
built a real device
use of off-the-shelf components
solid prototyping and experience with Internet Technologies
well written report, nice organization, clear objective, good evaluation
good observations, hypotheses, and evaluation
good performance study and results
evaluation and user study
experience, multiple prototypes
solid design and implementation
solid experience in system design, networking. Nice integration with quake game engine
Tough assignment and reasonable accomplishments, evaluation (while imperfect) is described nicely
iterative, agile development and strong communication experience
Research
contribute to a bigger project
nice evaluation and tie in with research
experience with different technologies
applied research, evaluation nice
user evaluation
evaluation and system performance design
Serious HCI design worksome functionality
understanding and enhancement of existing tool, user testing
System building
Generally successful
real company, addresses real need
Working with existing project
tackle interesting and hard problem
plugin development
Solid compiler and languages effort
solid work and contributes to larger project
Working demonstration, neat idea
Integrated project with real benefit to Cisco

2.16 Project Weaknesses

Table 14 contains specific advisor and 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. The most common problem were issues with the evaluation and testing portion of the projects.

In some cases, the reviewers felt the material was not “core CS”, perhaps a testament to the projects completed at the Wall Street project center, where projects are, at times, distinctly Information Technology.

2.17 Interdisciplinary Work

There were four projects involving other departments (ECE and MG). One project was done in conjunction with a faculty member on the Boston College Computer Science Department.

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 15, suggesting that an A should never be rated less than a 4, a B should receive an evaluation of

Table 14: Project Weaknesses

mixed success
oversight management of the project
report not well organized and need better motivation
not much development
no development work
no evaluation, didn't meet all objectives because of time constraint
training data for neural network flawed, evaluation section seriously weak
could have (perhaps) a bit more evaluation.
not enough CS material
no specific description of game/tour
limited evaluation, more feedback on ebay reaction
need more analysis. Evaluation needs more detail. Conclusion is nearly empty
much of the Maude part of the MQP report is taken right from the Maude 2.0 primer
lack of independent evaluation
no evaluation or design consideration. Project requirements vague or simply missing. What were the students trying to do?
This is strictly an extended version of a conference paper. More work should have been done on extending the write-up to be suitable for an MQP report organization
not clear on criteria for choosing tracker
alternative configurations of network
Background on HCI presented in an incohesive manner; unclear why team didn't attempt at least some functionality
Without callibration, this project is clearly ineffective, therefore evaluation is non-existent
report not well organized
not enough discussion on alternatives, how project developed, evaluation
not clear motivation for why particular features added
report not well organized, lack of evaluation, not a good discussion of organization no evaluation and small project no real evaluation, no clear set of objectives over the web Too much on-the-job learning, report has highly redundant text as problem is described repeatedly

Table 15: 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

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 16 shows the correlation between the project evaluation and the project grade assigned by the advisor. The projects were evaluated before obtaining the project grade.

Table 16: Correlation of Project Grade with Quality of Project by percentage

	Project Quality					
Grade	1	2	3	4	5	Total
C	5	0	5	0	0	9
B	0	0	7	5	0	11
A	0	**7**	**27**	32	14	80
Total	6	9	42	41	19	100

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.
- 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 63% (versus 73% in 2001) of the projects have correlating evaluations (C1), 34% (versus 25% in 2001) fall into case

C3, and 0% (versus 2% in 1999) fall into case C2. In Table 16, cases C2 and C3 are represented by **bold** faced entries in the lower left triangle of the array. These numbers indicate that the reviewers believe there is an increase in 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. Three projects (7%) received an A grade although they were assessed to be less than adequate. 27% 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 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 17 shows the correlation between the number of faculty and the project evaluations. The two indicators are report quality (RQ) and project quality (PQ).

Table 17: Correlation of Faculty Team Size and Evaluation

Faculty Team Size	% of Projects	avg Grade	avg RQ	avg PQ
1	81%	3.74	3.29	3.47
2+	18%	3.75	3.0	3.5

Comment: There is no real difference when separating individually advised projects (81% of the total) from group-advised projects.

3.3 Correlation of Student Team Size and Evaluation

Table 18 shows the correlation between the number of students and the project evaluations.

Table 18: Correlation of Student Team Size and Evaluation

Student Team Size	% of Projects	avg Grade	avg RQ	avg PQ
1	30%	3.77	3.15	3.61
2	30%	3.57	2.69	2.85
3	36%	3.81	3.72	3.88
4	4%	4.0	3.5	3.5

Comment: There was a noticeable gap in performance for projects with two students. While the grade was just slightly lower than the other groups, the average report quality and project quality was considerably lower than the other project groups.

3.4 Correlation of On/Off-Campus Projects and Evaluation

Table 19 shows the correlation between projects that were sponsored on/off-campus and the project evaluations. While nineteen (43%) of the projects were sponsored or involved off-campus companies (see Section 2.3), only 16 (or 36%) of the projects were completed off campus.

Table 19: Correlation of On/Off-Campus Projects and Evaluation

Type	% of Projects	avg Grade	avg RQ	avg PQ
On	63%	3.64	2.84	3.16
Off	36%	3.92	3.94	4.03

Comment: In contrast with the 2001 survey which showed little difference in evaluated quality between off-campus projects and on-campus projects, the off-campus projects for this review appear to do better than on-campus projects in all categories. Most notably, the report quality for off-campus projects is over one point higher, and the project quality about .85 points. We expect that the added rigor of having an external sponsor helps motivate students to perform at a higher level.

3.5 Correlation of Project Duration and Evaluation

Table 20 shows the correlation between the registered units for a project and the project evaluations.

Table 20: Correlation of Registered Units and Evaluation

Registered Project Units	% of Projects	avg Grade	avg RQ	avg PQ
1	75%	3.76	3.44	3.67
1 1/6	20%	3.83	2.78	3.0
1 1/2+	4%	3.0	2.0	2.5

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

Table 21: Correlation of Project Report Size and Evaluation

Project Report Size	% of Projects	avg Grade	avg RQ	avg PQ
0–39 pgs.	48%	3.61	2.67	2.98
40–69 pgs.	39%	3.88	3.62	3.85
70+ pgs.	14%	3.78	4.17	4.17

Comment: The results of this correlation show that the quality of both the report and project track with the size of the project. This result has generally been the case in previous reviews as shorter reports indicate that students did not accomplish much or that they did not allocate enough time to write an adequate report. The worrisome part of this data is the average grade for those projects with a short report. ¿From the subset of Table 21, there were 10 projects with less than 28 pages. The average grade of these projects was 3.28 with more than half receiving an A grade.

3.7 Correlation of Computer Science Level and Evaluation

Table 22 shows the correlation between the Computer Science level and the project evaluations. The level provided by project advisors is used for this correlation.

Table 22: Correlation of Computer Science Level and Evaluation

Computer Science Level	# (%) of Projects	avg Grade	avg RQ	avg PQ
< 3000	1 (3%)	4.0	3.0	3.0
3000	9 (25%)	3.1	2.7	2.9
4000	23 (64%)	3.9	3.3	4.3
grad	3 (8%)	4.0	3.6	4.0

Comment: There is a single project rated by the reviewers as having insufficient Computer Science content. In general, the CS effort correlates with the average grade received and the average report quality. But there is an aberration with the three projects rated to contain graduate-level effort; these projects scored lower in average project quality. This unexpected drop-off might occur because the difficulty of the project work might have made it hard for the students to complete or properly evaluate their work on the project.

3.8 Correlation of Math Level and Evaluation

Table 23 shows the correlation between the math level and the project evaluations again using the evaluations from project advisors.

Comment: We observe that all projects with at least some math content received an A. This led us to conclude that it is hard, perhaps, for Computer Science advisors to use math level to differentiate project grades.

Table 23: Correlation of Math Level and Evaluation

Math Level	# (%) of Projects	avg Grade	avg RQ	avg PQ
none	19 (53%)	3.4	2.8	3.1
lin. alg.	2 (6%)	4.0	2.0	2.5
prob/stat	11 (31%)	4.0	3.4	3.7
4000	4 (11%)	4.0	4.2	4.5
grad	0 (0%)			

The MQP reviewers consistently rated the report quality higher for those projects that incorporated math within the report. With the exception of two math-2 projects whose project quality seems pointedly low, the project quality appears to also correlate with the math level.

3.9 Correlation of Overall Effort Level and Evaluation

Table 24 shows the correlation between the overall effort level and the project evaluations again using evaluations of the project advisors.

Table 24: Correlation of Overall Effort Level and Evaluation

Overall Effort Level	# (%) of Projects	avg Grade	avg RQ	avg PQ
1 too little	3 (8%)	2.1	1.3	1.7
2	4 (11%)	3.3	2.3	3.0
3 about right	15 (42%)	3.8	3.2	3.3
4	13 (36%)	4.0	3.7	4.0
5 too much	1 (2%)	4.0	3.0	4.0

Comment: One project, rated by the advisor as requiring too much work, received a lower report quality score, likely because the final report produced by the student wasn't an accurate reflection of the true effort involved. In all other cases, the effort correlated well both the report and project quality.

4 Conclusions and Recommendations

The 2006 review of Computer Science MQPs reflects data and evaluations for 44 MQPs, involving 85 computer science students, that were completed between the Summer of 2005 and the Spring of 2006. 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 (88%) in this year's review compared to the previous year (92%). This slight drop is in contrast with the percentage of A grades assigned by project advisors, which is at its highest-ever level of 82%. The increased grades with a drop in project quality is a cause for concern. The difference between project grade and evaluation was most apparent for projects done on campus.

81% (down from 85% in 2001) 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.

4.2 Quality of Report

The overall quality of the reports themselves was worse than the previous review with a bigger drop than in the quality of the project. Aspects of the report that were judged to drop the most in quality were the background/literature review and the discussion of project methodology and issues.

4.3 Students per MQP

The number of single student CS MQPs was 30%, which is similar to the 28% in 2001. The average number of students per project dropped to 2.2 from 2.3 in 2001. These numbers are generally better than previous years and indicate that faculty continue to group students together on projects. Unlike previous years, the most problematic projects in terms of grades and evaluation were the two-student projects, which earned the lowest grades and significantly lower evaluations than other team sizes. We have no clear explanation for this result.

4.4 Distribution of CS Faculty over MQPs

After an improvement in the distribution of the MQPs over faculty in past reviews, the current review found the highest concentration of MQPs amongst a relatively small number of faculty. 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 43% of projects up from 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 dipped a bit compared to 2001, but its use dropped much less than other languages such as C, C++ and Perl. 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 Recommendations for the Next CS MQP Review

The evaluation process generally worked well. Extending the process to included faculty advisors continues to a success as all faculty returned their project evaluation forms and these forms allowed more accurate information about the projects to be included in this report.

Two aspects of the process that could be improved are better integration with an existing procedure for evaluating MQP presentations as well as one for evaluating projects in meeting department-defined student learning outcomes. The annual timeframe of these procedures does not match the periodic nature of this MQP evaluation.

While the review forms worked well and provided a substantial amount of data, some revisions are needed for future reviews. These include:

- The question concerning "Math Level Demonstrated" needs to include a response for "some" math, which was not reflected in the current set of possible responses.

- The question about the types of work encompassed by the project needs work. For example, there was not an explicit response for "testing." There was confusion with responses for both "Design" and "Design/Implementation." The types of project activity also needs to be better aligned with the set of student learning outcomes.

- The department needs to consider involvement of external (to the department) professionals, such as alumni, in the review process.

4.8 Recommendations for Improving CS MQPs

The following list of recommendations are drawn from the analysis and conclusions of this Computer Science MQP Review. They are similar to those recommendations from previous reviews.

- Increase student team size and avoid single-student projects when possible. 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.

- Ensure that project reports are complete and encompass all aspects of a project including proper background research. A decrease in the quality of the project report was a negative result of this MQP review and needs to be a point-of-emphasis moving forward.

- 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.

- Emphasize the need for students to indicate why the MQP was a good experience, what was difficult about the project 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.

- 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.

- Continue to work with external companies and organizations to sponsor MQPs. Externally-sponsored projects are both beneficial for students and generally lead to better quality projects.

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

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

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

[5] 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.

[6] 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 Advisor and Reviewer Form

The following form was used by advisors to evaluate all MQP projects. This form was also used by the reviewers.

	1	2	3	4	5
CS Level Demonstrated	1000	2000	3000	4000	grad

	1	2	3	4	5
Math Level Demonstrated	none	lin alg	prob/stat	4000	grad

	1	2	3	4	5
Demonstrated Professional Ethics	poor		adequate		excellent

	1	2	3	4	5
Software Project Size/ Programming Effort	none		moderate		large-scale

	1	2	3	4	5
Learned New Tools/Techniques/Info	none		some		considerable

	1	2	3	4	5
Project Objective met	unknown	no	mostly	yes	exceeded

	1	2	3	4	5
Overall Effort Level (worth one unit/student)	too little		about right		too much

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

   Analysis
   Data Collection (Empirical)
   Design
   Design/Implementation
   Performance Evaluation
   Research
   Simulation
   Survey
   Other________________________

   AI
   Architecture
   DataBase
   Graphics
   HCI
   Languages/Compilers
   Software Engineering
   Operating Systems
   Distributed Systems
   Other________________________

   Theory/Foundations
   Networks
   Webware

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

C
C++
Assembly Lang.
Lisp/Scheme
Java
Perl/Python/PHP/Tcl/Tk
Other________________________

Macintosh
PC/Windows
PC/Linux
CS Unix
CCC Unix
Other________________________

3. Project strengths/weaknesses/publications/other comments?

B Reviewer-Only Form

The following form was used only by the reviewers to evaluate all MQP reports.

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

1. Was this project a good learning experience? What was learned by the student(s)?
2. Project strengths: Project weaknesses:
3. Was this project a continuation of an earlier project, and if so, did the students indicate the part of the work that is theirs?
4. Did this project involve any interdisciplinary work? What departments or organizations were involved? Off-campus or on.