WPI-CS-TR-15-04 August 2015

2015 Computer Science Department MQP Review

by

Robert E. Kinicki

Hugh C. Lauer

Computer Science Technical Report Series

WORCESTER POLYTECHNIC INSTITUTE

Computer Science Department

100 Institute Road, Worcester, Massachusetts 01609-2280

2015 Computer Science Department MQP Review

Robert E. Kinicki and Hugh C. Lauer
Summer 2015

Abstract

This report presents results of a Summer of 2015 peer review of Major Qualifying Projects (MQPs) completed to satisfy the Computer Science major during Academic Year 2014-2015. The intent of this study is to assess whether the MQPs in the department are accomplishing their educational goals and fulfilling department-defined student learning outcomes.

The report identifies problems that need to be addressed and trends that should be continued to make the MQPs a worthwhile learning experience. It reflects data and evaluations for 64 MQPs involving 114 computer science students who submitted CS projects during the academic year 2014-2015. Additionally, the report compares its findings with similar reviews done in the past.

This year, a much higher proportion of projects involved students with dual majors and students with other majors than in previous years. The 64 MQPs in this review involved a total of 163 students, meaning that 49 of the students did not have Computer Science as one of their majors at all. It also means that many projects were co-advised by faculty members not in or affiliated with the Computer Science Department. The reviewers expect this trend to continue.

This review indicates a wide disparity in the sizes, scopes, levels of difficulty, levels of effort, and qualities of the projects. One noticeable trend is that writing quality is often lower than the project quality. This suggests that a number of reports were hurriedly, and often poorly, written near their project deadlines. A consequence of the writing quality is that the reviewers' opinion of project quality, based only on the reports, is extremely inconsistent with the nominations for best project at Project Presentation Day. The reviewers also noted that some projects could hardly be considered capstone-level work in Computer Science.

This year's MQP review includes faculty evaluation of projects relative to department learning outcomes and 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. Among these recommendations are:–

- The department should have a full and frank discussion about its expectations of non-CS students participating in CS MQPs and of CS student participating in non-CS MQPs.
- The department should commission the writing of an "MQP Manual" or syllabus. This would provide guidance to both students and faculty members about what constitutes an MQP, what is expected of a report, what are its educational goals and learning outcomes, how to work with dual major students and non-CS students, and how to plan the work of the Project.
- The CS Advisor Survey should be updated to reflect the modern languages and platforms on which MQPs are carried out.

1 INTRODUCTION

1.1 PURPOSE

The Major Qualifying Project (MQP) is a fundamental component of Worcester Polytechnic Institute (WPI)'s project-based undergraduate curriculum. WPI requires all undergraduates to successfully complete an MQP in each of their majors. The MQP for the Computer Science (CS) major is a capstone experience (or capstone-like experience)[1] that gives students an opportunity to apply learned fundamentals and skills to relevant problems in the field of Computer Science. The project may involve original research, data collection, analysis, or design of a system that often includes a software implementation. The student/advisor team determines an approach for an MQP that enables students to study an area of Computer Science in depth or to combine areas into a single project. MQPs for students majoring in Computer Science require one unit of work.[2] Students pursuing double majors must either complete two separate MQPs (one for each major) or a single 4/3 unit project that encompasses substantial effort in relevant areas of both majors.

This report, the ninth in a series, discusses the results of a peer review of MQPs conducted within the Computer Science Department during the summer of 2015. The goal of the report is to assess whether the MQPs for the CS major are accomplishing their educational goals. The report identifies problems that the CS Department needs to address and trends the Department should continue to make CS MQPs worthwhile learning experiences. It reflects data and evaluations for 64 Computer Science MQPs that CS majors completed between summer 2014 and spring 2015. These MQPs involved 163 students in all, of whom 114 were declared Computer Science majors or were double majors, with one of the majors being Computer Science. This review does *not* include non-CS MQPs advised by CS faculty members — for example, Interactive Media and Game Development (IMGD) and Robotics Engineering (RBE) MQPs. Table 1 provides numerical totals for student involvement in CS MQPs for this and the eight prior reviews conducted by the WPI CS Department.

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
2015	64	116

The Computer Science Department did not conduct any department-wide MQP reviews between 2008 and 2015, but it has collected advisor survey information about CS MQPs on a continuing basis since academic year 2009-2010.

2 REVIEW SCOPE AND PROCESS

The CS Department Head appointed Professors Robert E. Kinicki and Hugh C. Lauer as the CS review team to conduct this study during the summer of 2015. To calibrate the process, both members of the review team

[1] The term *capstone* is used informally throughout this report to indicate a level of advanced undergraduate work that embodies a broad spectrum of skills and knowledge accumulated during the undergraduate education in Computer Science. By contrast, *capstone* is a formal term — and requirement — in many of the engineering disciplines at WPI. The use of the term *capstone* in this report, as in previous MQP reviews in Computer Science, is not intended to suggest any sort of similar formality or requirement of Computer Science students.

[2] One unit at WPI is equivalent to one academic quarter of work full-time or three academic quarters of work at one-third time (the same as most WPI undergraduate courses).

began by examining the same small subset of five project reports. Review in the context of this report does not imply the reviewer read the complete report in every detail, but rather that the reviewer inspected an MQP report at a sufficient level to conscientiously rate the criteria specified in the 2015 CS MQP Review Form. For each project review, the reviewer completed a Review Form (Appendix A) in the form of an individual Excel spreadsheet. Starting from the 2008 review form, the review team adjusted and modified the criteria in the 2015 Review Form to take account of new factors such as MQPs representing a student's effort in more than a single major while remaining consistent with previous reviews. In addition to criteria used in past CS Department MQP reviews, the 2015 Review Form added eight questions that the Undergraduate Outcomes Assessment Committee (UOAC) requested as part of a university-wide study. Once the reviewers completed all the individual MQP Review Forms, the next step in the review process was to transcribe the individual Review Forms into a single combined spreadsheet for the purpose of analyzing and presenting the results of this investigation.

2.1 SCOPE OF THE REVIEW

The intent of the review team was to include all MQPs submitted by students with majors or double majors in Computer Science during the past academic year. Finding and identifying these students and their MQP reports proved to be non-trivial. The team took two parallel approaches. In one, after identifying and enumerating all the CS students who completed MQPs during the target interval, the team attempted to match each student with an MQP. In the second approach, the Registrar's Office provided a list of MQPs submitted during the academic year, filtered to include only those who list Computer Science as a major or as one of a dual major. The benefits of this strategy were that the review team uncovered a small number of students who fell through the cracks in one or the other of the approaches while additionally eliminating a small number of non-CS MQPs completed to satisfy a CS student's other major.

2.1.1 Enumerating Students

The Office of the Secretary of the Faculty provided a spreadsheet that listed by major all WPI undergraduates who graduated in Academic Year 2014-2015. The review team filtered this list to include only the students with a major or double major in Computer Science. Table 2 shows the filtered results. Three students graduated in October 2014, eight graduated in February 2015, and 107 graduated in May 2015. The team then merged this list with the Computer Science Department's own list of seniors to identify an additional six seniors who did MQPs but did not graduate yet, bringing the total of possible students up to 124.

Table 2: CS Students in 2015 Review

Students	*Count*
Graduated 10 Oct 2014	3
Graduated 27 Feb 2015	8
Graduated 16 May 2015	107
Did not graduate yet	6
SUBTOTAL	124
MQPs done last year	–10
CS MQP STUDENTS IN 2014-15	114

The team then searched the Library database for MQPs by these students. As ten students completed their MQPs in an earlier academic year, the review team eliminated these MQP from this study. At least two, possibly more, students completed *two* distinct MQPs, one in each of two majors. Based on available information, the review team decided these students completed a non-CS MQP in an earlier academic year, leaving only the CS MQP as the subject of this review. Additionally, the review team looked for, but did not find, any juniors who completed MQPs in Computer Science in 2014-2015.

2.1.2 Enumerating MQPs

The Review team then attempted to enumerate all of the MQPs completed by the 114 students identified in Section 2.1.1. This involved a combination of consulting the Registrar's records, searching the Library's online database of projects, and polling CS Department faculty advisors. Carla Mararian of the Registrar's office kindly ran an SQL query of the Registrar's database for all MQPs submitted by students with Computer Science listed as a major. This produced 158 records, many of which were duplicates (because of more than one CS team member per project and/or because some students had multiple registrations for the same project under separate numbers or advisors).

In parallel, the review team searched the Library database for MQPs by students listed as CS majors. This search was incomplete because several MQPs are confidential at this time. A few of these appear in the database with abbreviated reports declaring them to be confidential, but others do not show up at all in the Library database.

In addition, between the Registrar's SQL query and the search of the Library database, the team identified six additional MQPs as *not* in Computer Science, even though the students are CS double majors. However, the review team is confident of correctly identifying the MQPs of all of the graduating CS students and the six non-graduating CS students, even if their reports are not all available. Table 3 is a list of the CS MQPs included in this review sorted by advisor.

Table 3: Computer Science MQPs in 2014-2015

No	CS Advisors	Title
1.	Agu	‡‡ General-Purpose Computing on GPUs for Storage Networks
2.	Agu	Smartphone Gait Inference
3.	Beck	Racket Programming with ASSISTments
4.	Berenson	Amazon Picking Challenge
5.	Berenson	Exploring Human-Robot interaction in Collaborative Tasks
6.	Berenson	Hydro Muscle Control System
7.	Berenson	‡‡‡ Improving Soft Pneumatic Actuator Fingers through Integration of Soft Sensors, Position and Force Control, and Rigid Fingernails
8.	Chernova	A Comparison of Genetic Algorithms Using Super Mario Bros.
9.	Chernova	Evaluating Multiple Caching Strategies for Semantic Network Applications
10.	Chernova	Human Supervision of Multi-Robot Systems
11.	Chernova	WALRUS Rover
12.	Ciaraldi	Augmented Unix Userland
13.	Ciaraldi	Hermes Blue Sheet Implementation
14.	Ciaraldi	MBUSuite: Registration and Scheduling System for Merit Badge University
15.	Ciaraldi	Optimizing the Control of a Wi-Fi based Teleoperated Mobile Wheelchair
16.	Ciaraldi	TAR Browser: TAR Archives as File Systems
17.	Ciaraldi	WPI Student Organization Management System
18.	Claypool	Vir and the Army of Tenebrax
19.	Dougherty	A Formalization of Strand Spaces in Coq
20.	Dougherty	Interactive Model Finding with Hominy
21.	Dougherty	Scenarios for Description Logic

No	CS Advisors	Title
22.	Eltabakh	† Airline Analytics and Services
23.	Eltabakh	Discovering Correlations in Annotated Databases
24.	Finkel	Check-In Testing Framework for iOS
25.	Finkel	Conquest of the Verse -- Conquering the Galaxy one Ship at a Time
26.	Finkel	Endicia Interactive Style Guides for Future Web Development
27.	Finkel	† Modifying the Jetson TK1 Board Support Package for Multiple Linux Distributions
28.	Finkel	NVIDIA Bug Services: Synchronization and Statistics
29.	Finkel	Optimizing an Endicia Web Service
30.	Gennert	Atlas in the Cloud
31.	Gennert	HydroDog Quadruped Robot
32.	Gennert	Project Squirrel 2.0: A Tree Climbing Robot
33.	Heffernan	Adding Interoperability to ASSISTments
34.	Heffernan	ASSISTments Smart Reporting
35.	Heffernan	Enhancing Google Docs for Essays and Peer Review
36.	Heineman	NavPro: Network Analysis and Visualization Using Provenance Data
37.	Hofri	†† Data Analysis of Residential Mortgage-Backed Securities
38.	Hofri	Foreign Future Variation Margin Reconciliation
39.	Hofri	JPMorgan Web Service Registry
40.	Krishna	An Identification System for Head-mounted Displays
41.	Krishna	Wearable Honeypot
42.	Lauer	Program Similarity Detection with Checksims
43.	Lindeman	Errant: An iOS Fitness RPG
44.	Lindeman	Hikari Michi: Studying Immersion in Games Using the Oculus Rift and Leap Motion Controller
46.	Lindeman	The EcoKids & the Paper Pests
47.	Michalson	Mobile Motion Capture
48.	Pollice	Design Pattern Generator for Python
49.	Pollice	Git Analytics Tool
50.	Pollice	Leap Motion Presenter
51.	Pollice	Map My Trip: A Leap Motion Web App
52.	Pollice	WanderLog Travel Application
53.	Rich	The Urban Whale: A Serious Game About the North Atlantic Right Whale, and a Template for Future Endangered Species Games
54.	Rich	Whiskey2D: A 2D Game Creator
55.	Ruiz	Data Preprocessing for Advanced Analytics
56.	Ruiz	Gemini: The Genomic Search Engine
57.	Ruiz	Predicting Clicks on Mobile Ads
56.	Ruiz	Trading in the Financial Market using Data Mining
58.	Rundensteiner	Advancing MATTERS
59.	Rundensteiner	†† Fraud Detection using Storm
60.	Rundensteiner,	SAM Fitness -- An Android Wellness-Application

No	CS Advisors	Title
	Agu	
61.	Sarkozy	Automated Building of Classic Chinese-English Dictionary and Chinese-Hungarian Dictionary
62.	Sarkozy	UCT-Enhanced Deep Networks for Move Recommendation in Go
63.	Shue	Virtual Machine Server Management Tools
64.	Ward, Ryder	Simworm: Simulation of Early C. elegans Embryogenesis

In this list, two of the MQP reports (#22 and #27, marked with †) are confidential, so that the Library database contains only the titles and abstracts. Three other MQP reports (#1, #37, and #59, marked with ††) are confidential and were missing entirely from the Library database at the time of the review. Finally, one report (#7, marked with †††) was in the form of a paper submitted for publication, jointly authored by students and faculty members of Worcester Polytechnic Institute and Carnegie-Mellon University. Therefore, it could not be reviewed and evaluated under the same criteria as the other reports and projects.

Therefore, of the 64 Computer Science MQPs completed in 2014-2015, the review team was able to examine 58 reports for this study. Note that when a comparative result does not require reviewing the full report, the results can include all 64 completed CS MQPs.

2.2 ADVISOR SURVEYS

An online system for collecting surveys from CS MQP advisors has been in place for a number of years. Although in principle, CS project advisors should complete these survey forms at the same time as submitting project grades, the review team had to prod CS faculty considerably at the beginning of this investigation to get a sufficient number of surveys to be useful. In the end, advisors submitted surveys for 48 of the 64 MQP identified in Table 3, including four of the confidential MQPs (for which the review team has no report).

Appendix B provides the complete Advisor Survey Form.

2.3 REVIEW QUESTIONS AND REVIEW PROCESS

The reviewers of the CS projects completed during the 2014-15 academic year executed a methodology consistent with past MQP review practice. Section 2.1.2 details the enumeration and collection process for the MQPs assessed in this review. While the review team obtained most of the MQP reports from the Library database, the CS advisors provided copies of a few reports that were not yet part of the Library database by the summer of 2015.

The two team members jointly selected five reports to be read by both members. As one of these five reports turned out to be confidential, the set of MQPs both reviewers examined was only four reports. Using a random number generator, the team initially assigned the remaining equally between the two review team members. As the review process filtered out several additional projects from the initial list, Professor Kinicki reviewed 29 project reports (including the four in common), and Professor Lauer read 34 project reports (including the four in common).

For each MQP report, the reviewer filled out a copy of the 2015 CS MQP Review Form, using a scale of 1 to 5 to rate the MQP according to each criterion in the top section of the form. The following outlines the guidelines both reviewers used to complete the Review Form for each MQP report examined:

- Assume a base rating of 3 for each criterion.[3]
- If a report was better than expected for that criterion, increase the rating to 4. In exceptional cases, increase the rating to 5.
- If a report was somehow deficient in a particular criterion, decrease the rating to 2 or even to 1 in particularly bad cases.
- The rating of zero was reserved for omitting something entirely that should have been in the report — for example, a missing bibliography.
- The rating of -1 (negative 1) was reserved for situations where the criterion was not relevant — for example, a criterion involving ethical standards or human factors might not apply to the proof of a mathematical theorem or to measuring the forces exerted by a robotic "hand" in some particular cases.

Each reviewer also had the option of adding comments to any criterion on the form.

The 2015 CS MQP Review Form also includes eight university-wide questions requested by the Undergraduate Outcomes Assessment Committee (UOAC) as a result of a vote by the Faculty. As per the UOAC request, the review team rated these questions on a scale of 1 to 3, and used the -1 rating only when a particular question was not relevant to a particular project.

After both reviewers completed all project review forms, Professor Lauer transcribed the recorded ratings to a single combined spreadsheet for collecting results. While this process preserved reviewer comments in the archive of the reviews, none of the comments were transcribed into the merged spreadsheet. For the four MQP reports jointly examined by both reviewers, the combined spreadsheet contained the average of both reviewers' assessments for each criterion.

2.3.1 Calibrating the Reviewers

When a review team consists of more than one reviewer, the question naturally arises whether the reviewers are reviewing according to consistent standards or whether there might be a built-in bias. The members of this team do not have a good answer to this question. Appendix C shows a comparison of the distribution of the reviews on a criterion-by-criterion basis. Based on this and on numerous discussions during the period of the review, the two reviewers do not see the differences between them as significant enough to prevent combining their assessments into a single report and set of statistics.

[3] Note that this rating scale is different from the *de facto* criteria for grading projects at WPI. Despite the guidance offered by the University, faculty members overwhelmingly assume a grade of A and then reduce it when circumstances warrant.

3 RESULTS

This section provides the results of the 2015 Computer Science MQP evaluations. Note: the display of most of the resultant data is on a per-project basis, not a per-student basis.

3.1 FACULTY/STUDENT RATIO

3.1.1 Computer Science only

Figure 1 shows a histogram of the number of Computer Science students per MQP in Table 3. The vertical axis indicates percentages of the total. Of the 64 MQPs in this study,

- 28 had only one Computer Science student,
- 22 had two Computer Science students, and
- 11 had three Computer Science students.

The vast majority of these projects had one advisor in the Computer Science Department. Only one project had two advisors from the Computer Science faculty.

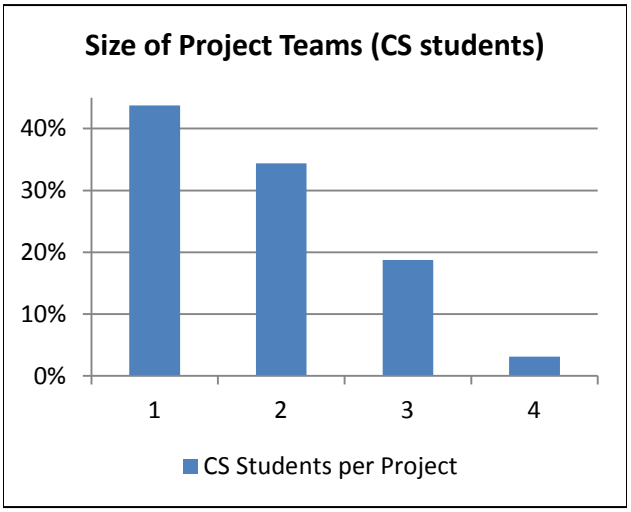


Figure 1: Number of CS Students per Project

3.1.2 Multi-disciplinary projects

Figure 1 and Section 3.1.1 do not tell the whole story. Of the 114 MQP students who listed Computer Science as (one of) their majors, 33 were dual majors from the following departments and programs:–

- BCB — Bioinformatics and Computational Biology (1)
- ECE — Electrical and Computer Engineering (2)
- ECS — Economic Science (1)
- IMGD — Interactive Multimedia and Game Development (10)
- MA — Mathematics (7)
- MIS — Management Information Systems (2)
- RBE — Robotics Engineering (10)

In addition, 49 other students, none of whom declared Computer Science as a major, joined them in 25 of the 64 MQPs of this review, to bring the total number of participating WPI students to 163. Figure 2 below shows a histogram of the total number of students and advisors.

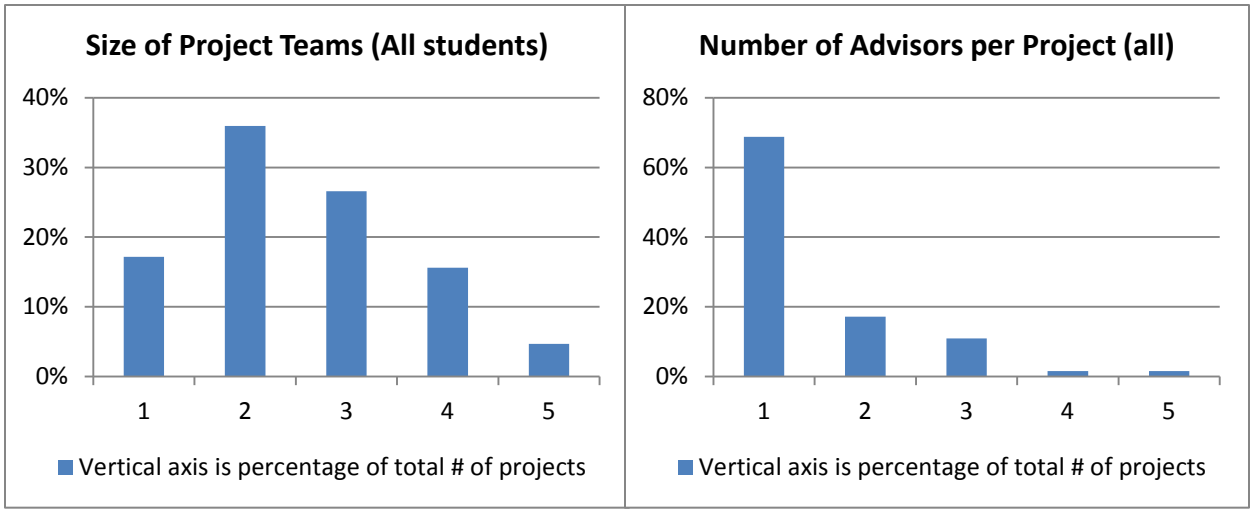


Figure 2: Numbers of students and advisors per project (all departments and programs)

The left side of Figure 2 shows a histogram of the numbers of students (of all majors) per project of Table 3, and the right side shows a histogram of the number of advisors (of all departments and programs). As can be seen, the number of true single-person projects is very small — 11 projects of the 64 considered (17%). This is significantly lower than in previous reviews. The majority of the projects are multi-person projects, often with one or more students from other majors. A non-trivial number are four- and five-person projects. The average number of students per project is 163 ÷ 64 = 2.5. This is higher than in previous reviews. The fraction of projects with a single advisor is 67% (i.e., 43 of 64), somewhat lower than in previous reviews. This is as expected given the introduction of the 4/3 unit dual MQP and the increased interdisciplinary nature of CS faculty research since the last review in 2008.

3.2 Faculty CS MQP Load

According to the CS Department Head, the Computer Science Department had 27 faculty members during the Academic Year 2014-2015, including tenured and tenure-track faculty, Professors of Practice, and other full-time faculty members for whom MQP advising is a normal part of their responsibilities.

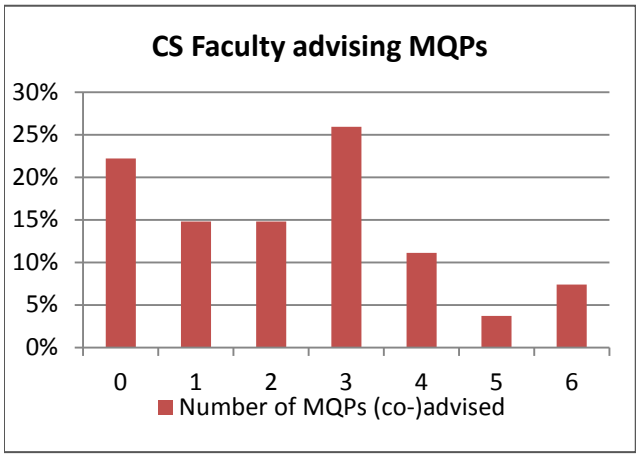


Figure 3: Percent of CS faculty members who advised or co-advised CS MQPs

Figure 3 shows the distribution on the number of CS MQPs advised by full-time CS faculty members. This data does not take into account the three projects that were advised by non-full-time faculty members in or affiliated with Computer Science (Professors Lauer, Michalson, and Ryder). It also does not take into account the advising responsibilities of Computer Science faculty members who also advise non-CS MQPs in areas such as RBE and IMGD.

Thus, the average CS MQP advising load per full-time CS faculty member was 2.2. This is considerably higher than the number reported in 2008 (1.6) and slightly higher than reported in 2006 (2.0). The change in the numbers may be reflected, in part, by the increase in dual-major MQPs.

3.3 SPONSORED AND OFF-CAMPUS PROJECTS

According to the Registrar's list of MQPs of CS and dual majors, twenty-two (34%) of the 64 projects in this review were sponsored by or involved off-campus companies and organizations. Table 4 shows the list of sponsors.

Sixteen (25%) projects in the scope of this review, involving 41 students (not all of whom were CS students), were done by students at WPI Project Centers listed in Table 5. The review team did not have sufficient information to determine whether other projects, sponsored or not, were carried out on campus or off campus.

Table 4: Sponsoring organizations for 2014-15 MQPs

ACI Worldwide, Inc.
Amadeus North America, Inc.
ASSISTments project at WPI
EMC Corporation
Endicia, Inc.
Huazhong University of Science and Technology, China
Institute of Computer Science and Control, Hungarian Academy of Sciences
J.P. Morgan & Co.
InterSystems Corporation
Massachusetts High Technology Council
Microsoft
MIT Lincoln Laboratory
New England Aquarium
NVIDIA, Inc.
Osaka University, Japan
QEUBIT Consulting LLC
Ritsumeikan University, Japan

Table 5: Project Centers for 2014-15 MQPs

Budapest
China
Japan
Microsoft
MIT Lincoln Laboratory
Silicon Valley
Wall Street (New York and Glasgow)

3.4 Project and Student Grades

In previous MQP reviews conducted by the Computer Science Department, project grades and student grades were reported separately. This practice seems to have stopped, and only student grades were available from the Registrar. However, it is possible to infer from the Registrar's data that in five projects, some Computer Science students earned different grades from their Computer Science partners.

Figure 4 shows the distribution of grades by project. Forty-eight (i.e., 75%) of the projects earned grades of A for all students, five (7.8%) had split grades between A and B, nine (14%) earned grades of B, and two projects (3%) earned grades of C.

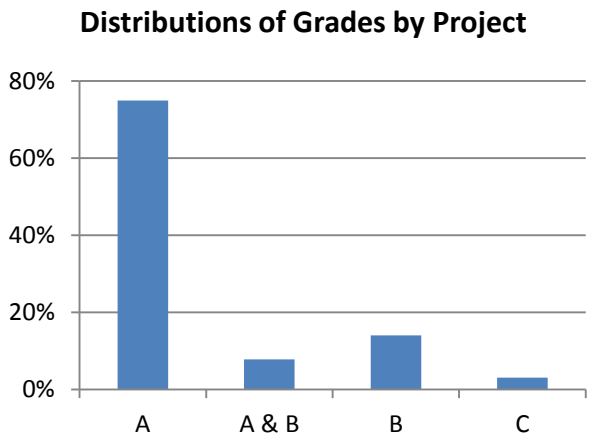


Figure 4: Distribution of Grades by Project

Figure 5 shows the distribution of MQP grades for all CS students enumerated in Table 2. As can be seen, the vast majority of projects and students earned grades of A — nearly 80% in both cases.

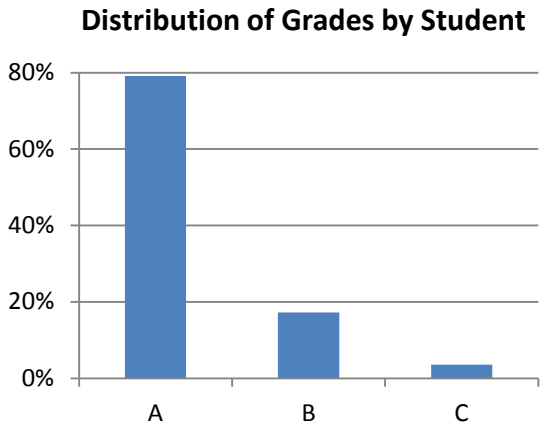


Figure 5: Distribution of Grades by Student

3.5 Project Continuations

At least seven[4] projects appeared to be continuations of previous projects. In some cases, these were continuations of previous MQPs. In other cases, they were continuations of Master's thesis work.

Three of the projects were part of larger, on-going efforts:–

- One project participated in the *Amazon Picking Challenge*.
- One project was a contributing component to a joint WPI-CMU entry in the DARPA Robotics Challenge.
- One project produced and integrated a piece of software for the on-going software platform of the CS-3733 course (Software Engineering).

3.6 Project Duration

Previous MQP reviews in the Computer Science Department reported the durations of projects in terms of units of credit earned. The review team did not have any such data for Academic Year 2014-2015.

[4] This number (7) may be understated. Discovering whether or not a project is a continuation of another project requires careful reading of the report. In at least one case, one member of the review team missed a continuation that the other found. Therefore, it is possible that some project continuations were overlooked.

One factor that would be useful to report and analyze is whether projects that earn 1-1/3 units for dual majors actually involve 33% more work and 33% more outcomes than single-major projects. The review team discussed this issue but had no information as to which projects earned the dual-major credit and also no way to evaluate the level of effort, difficulty, or outcomes of a dual-major project over a single-major project. Moreover, there is no way to evaluate from the data and the reports whether dual-major students did more work within a team than single-major students did within the same team.

3.7 PROJECT REPORT SIZE

The average size of the 58 project reports that the review team were able to read is 47 pages (with a range of 8–116), excluding appendices, code, front matter, and other pages before the body of the report, but including the list of references. Figure 6 shows a histogram of report lengths.

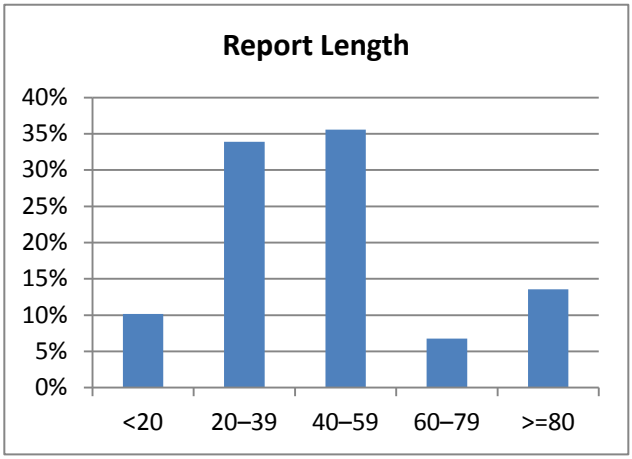


Figure 6: Distributions of Lengths of Reports

The shortest report (8 pages) was a copy of a paper submitted for publication with no mention of its being a report about an MQP.

The average length of reports is about the same as previous MQP Reviews in Computer Science:– 45 (1991), 49 (1993), 50 (1995), 59 (1997), 50 (1999), 58 (2001), 48 (2006), and 44 (2008).

3.8 REFERENCES

The review team considered both the number and quality of the references in each report. The average number of references per report was 20 (with a range of 0–51). Results are shown in the histogram in Figure 7.

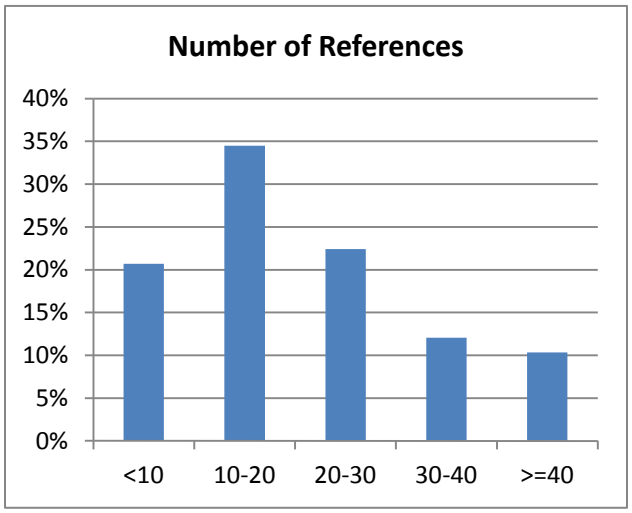


Figure 7: Number of References

When evaluating the quality of the references, the Review team considered that a reference in the format required for a scholarly publication would be considered to be the highest quality. A reference in the form of a URL to an established web site of an official organization would be considered medium quality if there were no other information about how to find the document, and a reference with that does not reliably direct a reader to the document (or only contains a URL of unknown provenance and security) would be considered the lowest quality. The reviewers then considered the entire list of references as a whole when assigning a value for this metric. Reports with mostly high quality ratings would earn a rating of 5, whereas a report with a mix of qualities of references might earn a rating of 3 or so.

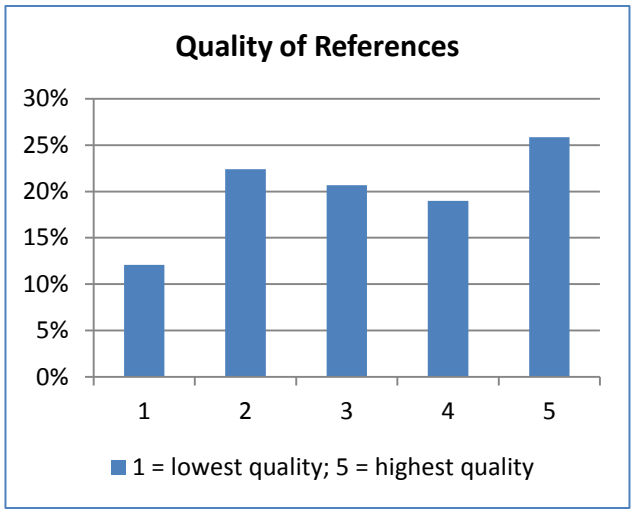


Figure 8: Quality of References

Figure 8 shows a histogram of the quality of references across the 58 MQP reports read by the review team.

3.9 Data from Advisor Surveys

For a number of years, the Computer Science Department has had an online system for surveying MQP advisors about the projects that they have advised. This year, MQP advisors submitted online surveys for 48 (75%) of the 64 projects in the scope of this review. Five of these 48 surveys pertained to projects for which the review team had no reports.

Four of the questions on the survey are about the type of work involved in the project, the areas of Computer Science that the project involved, the programming languages and tools used in the project, and the hardware platforms used to carry out the project.

In earlier MQP Reviews, the data reported was taken directly from the Advisor Surveys. In the 2008 Review, the review team determined this information itself directly from the project reports. In this review, the review team again attempted to determine the information from the reports. However, team members felt that that the reports had too little information to provide an accurate picture and that determination of this information from those reports was more often speculation rather than informed knowledge. Therefore, the following information was derived from the 48 Advisor Surveys and is expressed as percentages of 48.

3.9.1 Type of Work

Figure 9 shows the type of work involved with the MQPs in 2014-2015. As can be seen, 100% of the projects involved some kind of implementation and most also involved some kind of design.

Note that the *Testing* category was added to the Advisor Survey Form at the suggestion of a member of the review team. Hence it was not available in surveys returned earlier than mid-May 2015. The advisors checked the Testing category on all but one of the 11 surveys submitted after it was added.

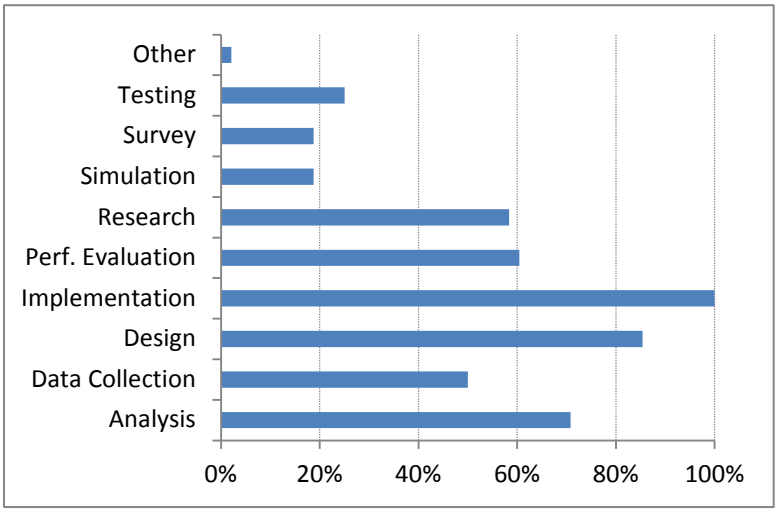


Figure 9: Types of Work

3.9.2 Project Area

The chart in Figure 10 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%). This information is drawn from the 48 Advisor Surveys received.

From the chart, it is apparent that the majority of projects involve Software Engineering. The percentages of projects involving HCI and Artificial Intelligence are up dramatically from 2008, and the percentages involving Webware and Graphics have declined slightly.

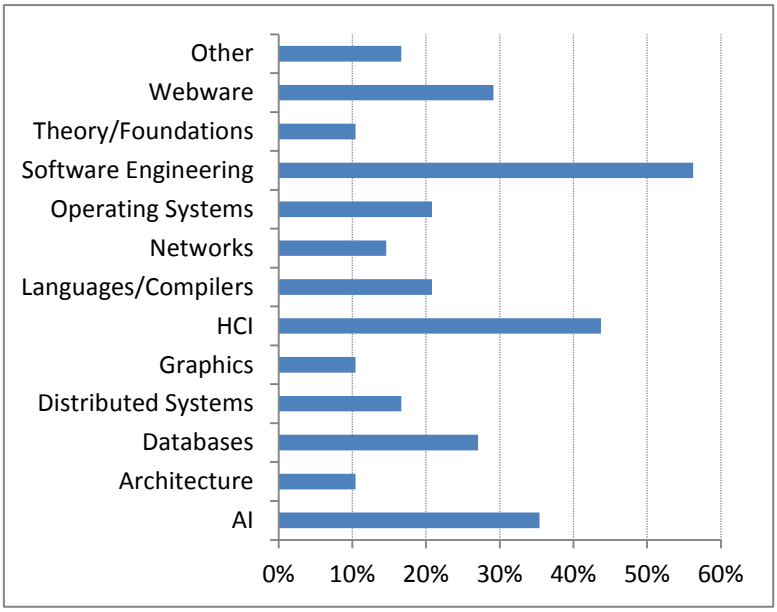


Figure 10: Areas of Computer Science

3.9.3 Software Used

The section on the Advisor Survey form regarding Languages and Tools is outdated and does not reflect the reality of modern Computer Science and modern MQPs. Figure 11 below shows that two options — LISP/Scheme and Assembly Language — were not used at all. The largest percentages of projects were implemented in scripting languages such as PERL, Python, PHP, Tcl/Tk, and Javascript. Only one project was implemented in Ruby, and the "Other" category accounted for one third of the projects. For example, several MQPs involving the development of games reported implementation in the Unity Game platform without mention of any programming language. Likewise, several projects cited the ROS (Robot Operating System) platform rather than any particular language, and others cited Arduino.

This section of the Advisor Survey should be revisited and revised for the next academic year.

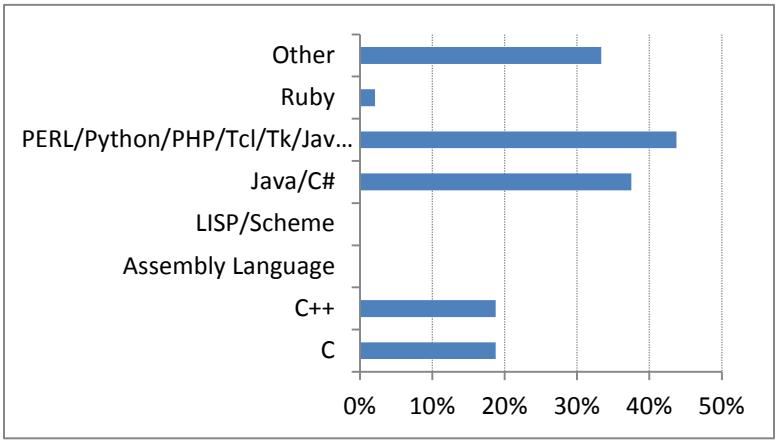


Figure 11: Languages and Programming Environments

3.9.4 Hardware Used

The section of the Advisor Survey regarding hardware is also outdated and does not reflect the wide variety of platforms available to projects.

Figure 12 shows the percentage of projects that used different types of hardware and operating system platforms, as reported by advisors in the Advisor

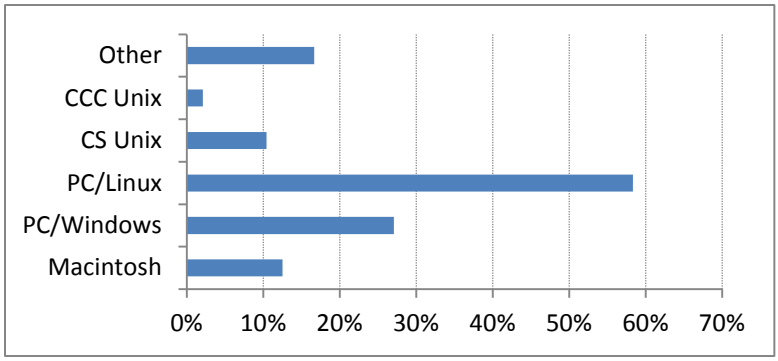


Figure 12: Hardware and Platforms

Surveys. The responses are suspect because the options available on the Survey form do not include platforms such as Android phones and embedded processors such as Arduinos and Raspberry Pi devices.[5]

For some MQPs, this project aspect is almost irrelevant, because student projects are being carried out at higher level of abstractions where the underlying machines and operating systems do not matter. On the other hand, for projects where the platforms are highly relevant, they rarely involve the traditional ones listed in this figure. The Advisor Survey should include an open-ended option to specify any platform and/or multiple platforms.

3.10 Reviewer Evaluations

Table 6 (split across two pages) shows a summary of review team member evaluations of the 58 reports that the team members were able to read, based on the criteria in the 2015 CS MQP Review Form of Appendix A The entries represent the numbers of projects earning each evaluation. Appendix C provides histograms of this information, organized by review team member.

Table 6: Review Team Evaluations

Categories	NA	0	1	2	3	4	5	Average
Abstract accurate and complete	0	5	0	7	17	19	9	3.3
Clearly-stated Objectives	0	10	7	10	11	10	10	2.6
Background	0	2	7	6	15	13	15	3.3
Literature Review	14	9	7	4	10	6	6	2.4
Quality of References	1	1	5	13	12	11	15	3.3
Grammar, Spelling, Typos	0	0	0	9	31	12	6	3.3
Quality of Visuals	0	1	4	8	16	19	10	3.4
Writing Style	0	0	6	8	25	9	10	3.2
Report Organization	0	0	8	10	22	11	7	3.0
Methodology Discussed	1	5	6	8	17	7	13	3.0
Project Problems Discussed	3	7	3	5	16	13	11	3.1
Implementation	2	4	5	7	19	7	14	3.1

[5] The 2008 MQP Review reported only PC/Windows and PC/Linux, and the review team for that review explained that it was impossible to determine most of the platforms from the reports.

Categories	NA	0	1	2	3	4	5	Average
Evaluation Done	2	17	3	3	11	8	14	2.6
Product Testing	10	14	5	2	12	6	9	2.4
Data Analysis	21	12	1	1	7	6	10	2.6
Report Quality	0	0	8	8	24	9	9	3.1
Project Quality	0	0	5	7	14	16	16	3.6
Objectives Met	0	17	5	5	12	9	10	2.4
Overall Effort	0	0	4	5	16	15	17	3.6

Several things stand out in this data.

- The average quality of projects (as rated by the reviewers) is 3.6 — i.e., similar to the previous review at 3.5.
- Average report quality deteriorated from 3.5 in the previous review to 3.1 in this review. This is one half of an evaluation point.

  This may reflect the reviewers' own biases, project advising standards, and writing standards. However, both reviewers came away with the impression of a significant subset of projects in which the students worked hard and then hastily wrote up their reports in the final days before submission.
- 17 of the projects (29%) appeared not to have met their objectives — at least so far as the review team could tell from the reports. This would be considered a negative in the case where objectives are reasonable and students are weak. On the other hand, if the objectives were extremely ambitious, then partial achievement of the objectives might indicate an extraordinary and worthy accomplishment.

  However, more than one quarter of the projects did not state their objectives adequately. This is a matter of coaching and guidance on the part of the advisors.
- In reviewing the reports, the review team evaluated the Background independently from the Literature Review. In retrospect, these should probably be considered together. In at least some reports, they were presented together. On the other hand, 14 of the reports reviewed (24%) had no literature review at all. We can imagine that in some projects, a Literature Review might not be relevant; however, it is surprising that the number is so many.
- Note that the Advisor's assessments of *Objectives Met* and *Level of Effort* disagree with the review team's assessments of the Objectives Met and *Overall Effort* as determined from the reports. Obviously, the advisors are in a better position to assess the actual outcomes of the projects. However, the report is sometimes the only lasting record of the project and conveys a different outcome.

3.10.1 Criteria from Undergraduate Academic Outcomes Committee

In academic year 2014-15, the WPI Faculty voted to survey all MQPs according a university-wide set of criteria. The Undergraduate Outcomes Assessment Committee (UOAC) developed the criteria, and the Dean of Undergraduate Studies directed all departments to evaluate their MQPs accordingly. The result was eight specific questions that the CS reviewers added to the 2015 CS MQP Review Form.

Table 7 shows the review team's evaluation of the MQPs of this study according to these criteria using the UOAC-specified scale of 1, 2, and 3 which roughly correspond to a C, B and A grade respectively. The evaluation of NA was used to indicate that the criterion did not apply to a project. In particular, the team members felt that the question

> "Understanding and applying ethical standards (for example, human and animal rights in research, respect for intellectual property, social and environmental responsibility, honest reporting of data, sensitivity to conflict of interest)"

is only relevant to a project that has human subjects, privacy concerns, social responsibilities, etc. In 43 of the 58 reports reviewed (74%), these were not concerns.

Table 7: UOAC Survey Questions[6]

Criteria	NA	1	2	3	Averages
1. Fundamentals	0	7	25	26	2.4
2. Skills & knowledge	0	8	22	28	2.4
3a. Writing	0	9	31	18	2.2
3b. Visual	0	8	31	19	2.2
4. Problem solving	0	8	21	28	2.3
5. Finding info	0	10	18	29	2.3
6. Ethical/Social	43	6	4	5	2.0
7. Lifelong learning	0	8	26	24	2.3

3.11 PROJECT STRENGTHS AND WEAKNESSES

While reading the 58 project reports, the review team members noted with some disquiet that some of the projects seemed very weak — so weak, in fact, that it called into question whether they could genuinely be considered capstone-level projects in Computer Science. This is reflected more in the comments than in the ratings according to individual categories. Comments include:–

> This seems like a simple, lightweight project that would befit a summer intern at a lower level, not a senior at WPI.
>
> A little lightweight for an MQP. Not a formal report. No background or analysis done.
>
> BIG problem is they never indicate game was finished. Also no discussion of how game was built; also no conclusions.
>
> This felt like a lightweight project. The students should have learned a lot about *xxx* from CS-yyyy, but they rejected those lessons and did not draw any analogies.
>
> Lightweight computational challenges.
>
> This is a really lightweight project, not up to the weight of an MQP. It is more like a summer internship.
>
> Terrible project — not worthy of a passing grade for these students; worst one I have read so far.
>
> A very lightweight performance assessment of two alternatives for programming the game platform.
>
> Very lightweight web server project.

In all, about nine of the 58 projects (15%) struck the reviewers as marginal quality for the educational goals and requirements of a WPI Major Qualifying Project.

[6] Note that the averages in this table do not include project evaluations marked NA.

3.12 INTERDISCIPLINARY WORK

As reported in Section 3.1.2, many projects involved students of other majors. In fact, 25 of the 64 projects reviewed (39%) included team members who were *not* Computer Science majors or double majors. Thus, this study infers that at least these projects were interdisciplinary. By contrast, the 2008 Review cited only three interdisciplinary projects and did not report how many non-CS students participated.

In addition, 27 of the 114 graduating and non-graduating seniors in Table 2 (24%) declared double majors including Computer Science. In some cases, these students did two separate MQPs, but in most cases they did joint, interdisciplinary MQPs.

However, the review team noted that MQPs handful of dual major students (mostly in RBE or IMGD) were primarily in the non-CS field, with very little Computer Science content. This issue is worthy of a departmental discussion about how to ensure adequate Computer Science content is such projects.

3.13 ADVISOR EVALUATION OF PROJECT LEARNING OUTCOMES

Table 8 (split across two pages) shows the Learning Outcomes of the 48 MQPs for which Advisor Survey were received. The data in Table 6 reflect the Advisors' assessments of the Departmental Learning Outcomes, not assessments by the review team.

Table 8: Advisors' Assessments of Learning Outcomes

Outcome	Not at all (1)	Somewhat (2)	Well (3)	Excellently (4)	Averages
Demonstrated an understanding of programming language concepts	3	10	18	17	3.0
Demonstrated knowledge of computer organization	14	21	7	6	2.1
Demonstrated an ability to analyze the behavior of computational systems	9	16	15	8	2.5
Demonstrated knowledge of computer operating systems	13	18	14	3	2.1
Demonstrated an understanding of the Foundations of Computer Science	29	12	6	1	1.6
Demonstrated an understanding of Software Engineering principles and the ability to apply them to software design	3	14	17	14	2.9
Demonstrated an understanding of Human-Computer Interaction	16	7	15	9	2.3
Completed a large software project	1	14	14	19	3.1
Demonstrated advanced knowledge of Computer Science topics	1	12	20	15	3.0
Demonstrated an understanding of mathematics appropriate for Computer Science	21	14	9	4	1.9
Demonstrated knowledge of probability and statistics	25	11	7	5	1.8
Demonstrated an understanding of scientific principles	13	18	13	4	2.2
Demonstrated the ability to design experiments and interpret experimental data	10	18	10	10	2.4
Demonstrated an ability to undertake independent learning	0	7	16	25	3.4

Outcome	Not at all (1)	Somewhat (2)	Well (3)	Excellently (4)	Averages
Demonstrated the ability to locate and use technical information from multiple sources	0	7	19	22	3.3
Demonstrated an understanding of professional ethics	17	16	10	5	2.1
Demonstrated an understanding of the links between technology and society	13	21	10	4	2.1
Participated effectively in a class or project team	6	7	20	15	2.9
Demonstrated the ability to communicate effectively in speech	0	11	22	15	3.1
Demonstrated the ability to communicate effectively in writing	0	11	23	14	3.1

4 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.

4.1 Project Quality and Report Quality

When reading the MQP reports, the review team sensed that report quality tended to be less than project quality. Table 9 shows the correlation of the review team's evaluation of report quality (vertical axis) and project quality for the 58 project reports that they were able to read. The diagonal represents projects in which reports and projects were evaluated the same by the review team. The cells to the upper right represent projects for which the review team felt that the report was *not* as good as the project. 27 of the reports (47%) did not do justice to the projects that they described. Only 4 of the reports (to the lower left of the diagonal) were rated better than their projects.

Table 9: Project Quality vs. Report Quality

Report Quality \ Project Quality	1	2	3	4	5
1	4	3	1	0	0
2	0	3	3	2	0
3	1	1	10	9	4
4	0	0	0	3	5
5	0	0	0	2	7

The review team also researched whether or not CS team members had previously taken ID-2050, a writing and project organizing course, during their IQPs (Interactive Qualifying Project) during the junior year). Figure 13 shows report qualities, as evaluated by the review team, as a function of ID-2050. The blue bars indicate projects for which at least one Computer Science team member earned credit for ID-2050, and the red bars indicate projects where none of the CS team members received ID-2050 credit. It is apparent from the chart that ID-2050 makes a difference in the qualities of the project reports.

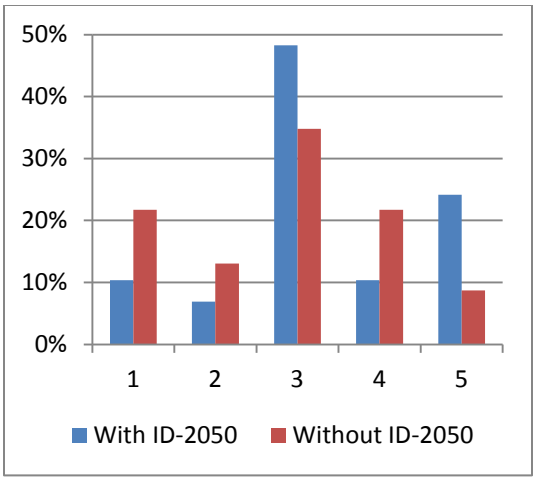


Figure 13: Report Quality vs. ID-2050 (5 is the highest quality)

4.2 Correlation of Reviewers' and Advisors' Assessments

Both the project advisors and the CS review team independently addressed two questions. These had to do with (1) whether and how well project teams met their objectives and (2) the level of effort of the project. Advisors were clearly basing their assessments on their knowledge of the projects and students themselves. The review team could only infer this information from project reports.

4.2.1 Objectives Met

Table 10 shows the correlation of the assessments of how well project teams achieved their objectives. The horizontal axis is the assessment by review team and the vertical axis is by the project advisors. The review team used a value of zero when it could not tell from the report whether the objectives had been met. This may be because a report did not clearly state the project objectives or because the report did not make clear that they really were accomplished.

Table 10: Assessment of Objectives Met (5 is high)

Advisor Assessment	Reviewer Evaluation 0	1	2	3	4	5
No	1	1	0	0	0	0
Mostly	2	1	1	3	1	4
Yes	6	1	3	3	5	2
Exceeded	3	0	0	1	0	2

As can be seen from this table, project advisors rated gave higher ratings to projects than the review team did regarding objectives met. This is indicated by the large number of entries to the lower left of the diagonal. It seems to be an issue that requires broader Department discussion about MQPs. (Note that the number of project entries in the table is 40. This reflects the projects for which the review team has both copies of the report and Advisor Surveys.)

4.2.2 Level of Effort

Table 11 shows the correlation between the advisor's assessment of the level of effort of each project and the review team's assessment of the same information from reading the report.

As can be seen from the table, most of the entries are on the diagonal or clustered near the diagonal. This means that the review team's assessments (based strictly on the reading of the reports) were close to the project advisors' assessments (based on direct knowledge of the projects).

4.3 CORRELATION OF EVALUATIONS

4.3.1 Correlation between Grades and Project Quality

Table 11: Assessment of Level of Effort (5 is high)

Advisor Assessment	Reviewer Evaluation 1	2	3	4	5
too little	1	0	0	0	0
a little	0	0	2	0	0
about right	1	2	7	5	4
a lot	1	1	5	5	6
too much	0	0	0	0	1

A perennial discussion at WPI is about grading standards for projects. On the one hand, official policy specifies that a grade of A is reserved for projects that exceed expectations, B is the grade for projects that meet but do not exceed expectations, and C is for projects that do not meet expectations. On the other hand, the *de facto* practice of many faculty members is to award a grade of A unless there are clear deficiencies in the project. Similarly, students *expect* grades of A on major projects (MQPs and IQPs) if they work hard and do what was asked of them by their advisors and/or sponsors.

Table 12: Expected Correlation between Grade and Project Quality

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

Table 12 shows what the correlation *should be* between grade and project quality according to the official policy.

Table 13 below shows the actual correlations between grades (assigned by project advisors) and project quality (estimated by the review team). The five columns under "Project Quality" show the numbers of projects earning the grades for each project quality evaluation — 58 projects in all. The column labeled "No Review" represents the six projects for which the review team did not have reports or could not review. All of those projects were assigned grades of A. This table shows only the grades for the Computer Science students. The review team has no information regarding the grades for students who are not majors or double majors in Computer Science. In five of the projects, the grades were split, so that at least one CS student earned and A and at least one earned a B.[7]

As can be seen from Table 13, grades of A were awarded to 48 of the 64 projects (72%), even though there were only 28 projects (44%) rated by the reviewers with Project Qualities of 4 or 5. Although project grading is worthy of a departmental discussion, it is not likely that the *de facto* practices of awarding high grades to MQPs will change.

Table 13: Actual Correlation between Grade and Project Quality

	Project Quality (5 high)						
Grade	1	2	3	4	5	No Review	Totals
C	1	1	0	0	0	0	2
B	2	1	2	4	0	0	9
A & B	0	0	3	0	2	0	5
A	2	5	9	11	15	6	48

[7] There were no projects with split grades involving a grade of C.

4.3.2 Correlation of Faculty Team Size and Evaluation

It does not make much sense to try to correlate the project grades and/or evaluations with the number of CS faculty members because all but two of the 58 projects evaluated had precisely one CS faculty member. Of the other two, one project had two CS faculty members and one had no CS faculty members.

However, because of the large number of multidisciplinary projects, it is useful to correlate the total number of faculty members in all areas and departments advising projects within the scope of this review. Table 14 shows this correlation. Not much can be concluded from this correlation, but it appears that the average report quality goes up significantly when the number of faculty members is greater than one.

Table 14: Correlation of Faculty Team size (all majors) and Evaluations

Faculty team size	# of projects	Percentage	Avg. Grade	Avg. Rpt. Qual.	Avg. Proj. Qual.
1	39	67%	3.7	2.9	3.5
2	10	17%	3.5	3.5	3.7
3	7	12%	4	3.6	3.7
4	1	2%	4	3	3
5	1	2%	4	5	5

4.3.3 Correlation of Student Team Size and Evaluation

It is useful to consider correlations of both the numbers of CS students (i.e., majors and dual majors) and the numbers of students of all majors. Table 15 shows the correlation when considering only CS students. Conclusions from this correlation are tenuous. It appears that the average grade and average report quality go up slightly as the team gets larger. Table 16 shows the same correlation, but considering students of all majors in projects in the scope of this review. In general, it appears that the average project quality and the average grade go up with the project size. Interestingly, the average report quality and average project quality decline for actual single student projects.

Table 15: Correlation of # of CS Students per Team and Evaluations

# CS students in team	# of projects	Percentage	Avg. Grade	Avg. Rpt. Qual.	Avg. Proj. Qual.
1	25	43%	3.6	3	3.4
2	21	36%	3.9	3.1	3.7
3	10	17%	3.7	3.3	3.6
4	2	3%	4	3.3	5

Table 16: Correlation of # of Students of all majors per Team and Evaluations

# students (all majors) in team	# of projects	Percentage	Avg. Grade	Avg. Rpt. Qual.	Avg. Proj. Qual.
1	10	17%	3.6	2.8	3.2
2	22	38%	3.8	3	3.5
3	14	24%	3.7	3.2	3.8
4	9	16%	3.8	3.1	3.6
5	3	5%	4	5	4.7

4.4 CORRELATION OF PROJECT REPORT SIZE AND EVALUATION

Table 17 shows the correlation between the project report size and the project evaluations. The report size does not include code and appendices.

Table 17: Correlation of Report Length and Evaluations

Report Length	# of projects	Percentage	Avg. Grade	Avg. Rpt. Qual.	Avg. Proj. Qual.
0-39	25	43%	3.7	2.6	3.1
40-69	22	38%	3.7	3.4	3.7
70+	11	19%	4.0	3.7	4.2

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 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 results of this year are mostly consistent with those of the 2008 Review, except that average report qualities are significantly lower. This is an extension of the trend discussed in Section 4.1 above.

4.5 CORRELATION OF PROJECT EVALUATION AND AWARD NOMINATIONS

Thirty-seven (37) of the 64 projects listed in Table 3 were presented at *Project Presentation Day* (April 23, 2015) in the Computer Science Department.[8] Following these presentations, nominations for awards were solicited from both students and faculty. Students nominated 11 of the projects, and faculty members nominated five more. The nominated projects are:–

Table 18: Projects nominated for Awards

No	CS Advisors	Title
2.	Agu	Smartphone Gait Inference
3.	Beck	Racket Programming with ASSISTments
11.	Chernova	WALRUS Rover
19.	Dougherty	A Formalization of Strand Spaces in Coq
28.	Finkel	NVIDIA Bug Services: Synchronization and Statistics
29.	Finkel	Optimizing an Endicia Web Service
35.	Heffernan	Enhancing Google Docs for Essays and Peer Review
36.	Heineman	NavPro: Network Analysis and Visualization Using Provenance Data
40.	Krishna	An Identification System for Head-mounted Displays
49.	Pollice	Git Analytics Tool
50.	Pollice	Leap Motion Presenter
52.	Pollice	WanderLog Travel Application
56.	Ruiz	Gemini: The Genomic Search Engine
56.	Ruiz	Trading in the Financial Market using Data Mining
57.	Ruiz	Predicting Clicks on Mobile Ads
64.	Ward, Ryder	Simworm: Simulation of Early C. elegans Embryogenesis

[8] A 38th project is listed in the program for the CS Project Presentation Day, but neither of the students is listed in the Registrar's information as a CS major, and the advisors are all from the Mathematics Department. That project was not included in this review.

The chart in Figure 14 below shows the reviewers' ratings of Project Quality and Report Quality for the nominated projects. The vertical axis shows the number of projects, and the horizontal axis indicates the review team's evaluation of the project and report quality of those projects.

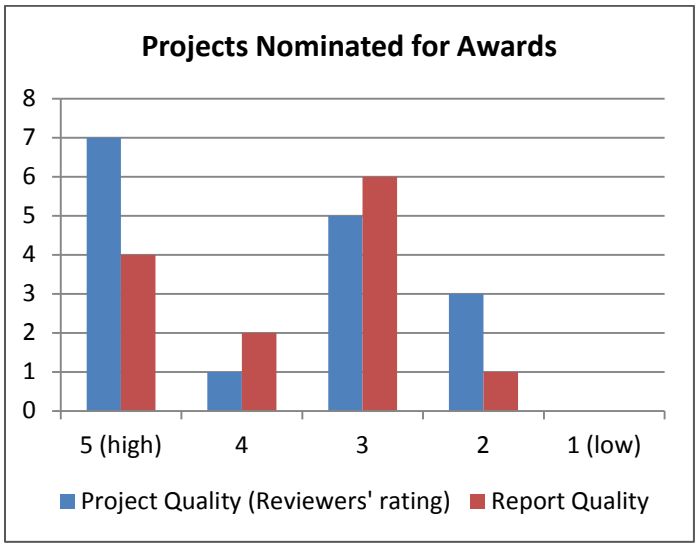


Figure 14: Project and Report quality for projects nominated for awards.

As can be seen from the chart, the reviewers rated seven of the eleven projects to be of high quality. However, the reviewers' ratings of reports of the nominated projects indicate that these were of lower quality.

One obvious conclusion is that it is very difficult for a third party (e.g., the review team) to assess the quality of a project from its report alone. The chart also reinforces a section observation — that report quality is less than project quality, even for projects that were deemed to be worthy of awards by both faculty members and students.

5 Summary and Recommendations

The 2015 review of Computer Science MQPs reflects data and evaluations for 64 Major Qualifying Projects completed between the summer of 2014 and the spring of 2015. These projects involved 114 students with majors or dual majors in Computer Science and 49 additional students of other majors. In this section, we attempt to draw some conclusions from the data collected during the evaluation process. The review team was able to read and evaluate 58 of the project reports. Five of the remaining reports were confidential. The last remaining report was in a form of a paper submitted for publication, but it was not suitable for evaluation as an MQP.

5.1 Quality of Projects and Quality of Reports

Approximately 85% were judged as at least adequate in this year's review, roughly the same fraction as in the previous review in 2008. The remaining 15% were judged to be below the level of expectations for capstone-level projects in Computer Science.

Forty-eight (75%) of the projects were awarded grades of A and an additional five projects (8%) were awarded split grades of A and B, including seven projects that were deemed by the review team to be inadequate. This suggests, at the very least, inconsistent grading practices across faculty members, but it also reflects the

tendency to award grades of A for work and project completion, rather than a dispassionate assessment of project quality by project advisors.

Report quality was conspicuously lower than project quality. Too many of the project reports had the appearance of hurried writing at the end of the project, very near a deadline. In additional, more than a few of the projects had the appearance of less than thorough reading by faculty advisors assigning grades.

5.2 INTERDISCIPLINARY PROJECTS AND STUDENTS PER MQP

Twenty-five of the 64 projects (39%) involved students in dual majors and/or majors other than Computer Science. This is significantly higher than reported in previous MQP Reviews in Computer Science. It clearly reflects a new reality in project advising and is worthy of one or more in-depth discussions by the Computer Science faculty.

As a result, only eleven of the projects (17%) were single-student Computer Science MQPs. This is substantially lower than reported in previous MQP reviews. When counting students of all majors, the average number of students per Computer Science project rose to 2.5, a jump of more than 50% from the 1.6 reported in 2008.

5.3 DISTRIBUTION OF CS FACULTY OVER MQPS

Whereas the 2006 and 2008 MQP Reviews reported very high concentrations among a relatively small number of CS faculty members, this review shows a much broader distribution. Moreover, at least five of the 64 MQPs were advised by faculty members who are not among the 27 faculty members in Computer Science for who MQP advising is a normal part of their responsibilities.

There is a growing trend for faculty advisors in other fields to contribute to the advising of CS MQPs. Counterbalancing this, however, is that a number of CS faculty members also have responsibilities for advising in other areas such as Robotics and Interactive Multimedia and Game Design. The review team was not able to quantify this diversity of advising responsibilities.

One effect of multiple faculty advisors is that Report Quality goes up noticeably with the number of faculty advisors. The review team attributes this to better and/or more thorough review of the reports before submission.

5.4 LEARNING OUTCOMES

The review team depended upon advisors' assessments of Learning Outcomes for the 48 projects for which advisors submitted surveys. However, the review team was not able to confidently discern any trends in this data from previous MQP reviews.

5.5 RECOMMENDATION — CS MQP MANUAL

The review team recommends that the Computer Science Department commission the preparation and writing of an *MQP Manual for Computer Science*, similar in spirit to the On-site Manuals that are prepared for each IQP Project Center. Such a manual should include a set of guidelines for what an MQP should be, what form it should take, what components it should have, and what faculty advisors and students should expect of each other.

Obviously, MQPs are not "one size fits all," but the review team detected many mismatches of expectations on the part of both faculty and students. The most egregious mismatch was the submission of a paper for publication as an MQP report, but there were many others.

5.6 RECOMMENDATIONS FOR THE NEXT CS MQP REVIEW

The evaluation process generally worked well, but the number of MQPs to review stretched the ability of a two-person review team to complete the review within the summer. The availability of the online faculty advisor evaluations is extremely important. This approach both allowed the outcomes to be explicitly included in the evaluation as well as having only one evaluation form for advisors to fill out. However, advisors submitted evaluations for only 48 of the 64 (75%) projects of this review, even after repeated prodding from the review team.

5.6.1 Advisor Evaluation Forms

The online Advisor Evaluation Forms are outdated in several respects. The form should be updated to include more modern options for languages/tools and hardware platforms. It should also be adapted to reflect the increasing numbers of multidisciplinary projects.

Currently, the Advisor Evaluation Form (also called the Advisor Survey) exists in the form of a web page that generates an e-mail when submitted. It has to be handled by two people before usable information is available to a review team member. This should be automated, or at least a program should be made available to review team members to gather late arriving surveys.

5.6.2 Data from Registrar

The review team did not have information regarding off-campus *versus* on-campus projects. It also did not have information as to how many and which students earned 1 unit, 1-1/6 units, or 1-1/3 units for their MQPs. While this information might be available, it was not apparent to the review team members until much too late during the review.

While the Registrar's office very kindly provided the information that the review team requested, it could not do so in a timely manner because of the needs of graduation and an immediate post-graduation recordkeeping. Moreover, it required the intervention of a third party before the Registrar's Office recognized the *bona fides* of the review team members.

5.6.3 Review Team Evaluation Form

Some fields should be added to the evaluation form used by the Review Team. One question that should be asked explicitly is whether, in the opinion of the review, the project is worthy to be considered a capstone-level experience in Computer Science.

Suggestions for minor changes include:–

- Include separate fields for the number of references and the quality of the references
- Include separate fields for project strengths and project weaknesses
- Recognize that sometimes a literature review and background are the same things and sometimes they are not.
- Be more clear about how to assess whether project objects are clearly stated and were met.

Appendices

Appendix A 2015 CS MQP REVIEW FORM

		N/A None Poor Adequate Excellent	
		-1 0 1 2 3 4 5	
Project Name goes here!		**Comments**	
Abstract accurate			
Objectives clear			
Background			
Literature Review			
References (number and quality)			
Grammar, spelling, and typos			
Quality of visuals			
Writing style			
Report organization			
Methodology discussed			
Project problems discussed			
Implementation discussed			
Evaluation done			
Product testing done			
Data analysis done			
Report Quality			
Project Quality			
Objectives met			
Overall effort			
Page count (excluding Appendices & code)			
Project strengths & weaknesses			
Was this project a continuation of an earlier project? If so, did the students indicate the part of the work that is theirs?			

Did this project involve any interdisciplinary work? What departments or organizations were involved? Off-campus or on?			
General Evaluation Comments			
Circle the following types of work and areas of computer science that are relevant for this project.			
		1. Analysis	1. AI
		2. Data Collection (Empirical)	2. Architecture
		3. Design	3. Databases
		4. Implementation	4. Distributed Systems
		5. Performance Evaluation	5. Graphics
		6. Research	6. HCI
		7. Simulation	7. Languages/Compilers
		8. Survey	8. Networks
		9. Other	9. Operating Systems
		10. Testing	10. Software Engineering
			11. Theory/Foundations
			12. Webware
			13. Other
Circle the following software languages, tools, and hardware resources used for this project.			
		1. C	1. Macintosh
		2. C++	2. PC/Windows
		3. Assembly Lang.	3. PC/Linux
		4. Lisp/Scheme	4. CS Linux/Unix
		5. Java/C#	5. CCC Linux/Unix
		6. Perl/Python/PHP/Tcl/Tk/JavaScript	Other
		7. Ruby	
		8. Other	

		Institute-Wide MQP Learning Outcomes **(rated on a scale of 1 – 3 as specified by the UOAC)**	
1. Applying fundamental and disciplinary concepts and methods specific to the major			
2. Skill and knowledge of current technological tools and techniques relevant to the major			
3a. Skill in written communication			
3c. Skill in visual communication (i.e., use of images and graphics to convey information, data, and ideas)			
4. Identifying, analyzing, and solving problems creatively through sustained critical investigation			
5. Finding, critically evaluating, and integrating information and ideas from multiple sources.			
6. Understanding and applying ethical standards (for example, human and animal rights in research, respect for intellectual property, social and environmental responsibility, honest reporting of data, sensitivity to conflict of interest)			
7. Skills, diligence, and commitment to excellence needed to engage in lifelong learning			

Appendix B ADVISOR SURVEY FORM

Project Advisor's Evaluation of MQP

Revised Version: Fri May 22 15:48:40 EDT 2015

Please use this web form to provide us with an evaluation of how well an MQP demonstrates each of the CS department's Outcomes. Only one response is needed for each MQP. Please provide values for *all* entries.

MQP Title:

Your Name:

Your Email:

MQP presented in: ○ December ◉ April

Grade given to project: ○ C ○ B ○ A

Number of Students: **Number of Advisors:**

The project objectives were met: ○ unknown ○ no ○ mostly ○ yes ○ exceeded

The overall student effort required was: ○ too little ○ a little ○ about right ○ a lot ○ too much

Based on the contents of this MQP we can claim that the student or students:

Demonstrated an understanding of programming language concepts.	-select-
Demonstrated knowledge of computer organization.	-select-
Demonstrated an ability to analyze the behavior of computational systems.	-select-
Demonstrated knowledge of computer operating	

systems.	-select-
Demonstrated an understanding of the Foundations of Computer Science.	-select-
Demonstrated an understanding of Software Engineering principles and the ability to apply them to software design.	-select-
Demonstrated an understanding of Human-Computer Interaction.	-select-
Completed a large software project.	-select-
Demonstrated advanced knowledge of Computer Science topics.	-select-
Demonstrated an understanding of mathematics appropriate for Computer Science.	-select-
Demonstrated knowledge of probability and statistics.	-select-
Demonstrated an understanding of scientific principles.	-select-
Demonstrated the ability to design experiments and interpret experimental data.	-select-
Demonstrated an ability to undertake independent learning.	-select-
Demonstrated the ability to locate and use technical information from multiple sources.	-select-
Demonstrated an understanding of professional ethics.	-select-
Demonstrated an understanding of the links between technology and society.	-select-
Participated effectively in a class or project team.	-select-
Demonstrated the ability to communicate effectively in speech.	-select-
Demonstrated the ability to communicate effectively in writing.	-select-

This project included the following types of work (mark all that apply) :

☐ Analysis ☐ Data Collection ☐ Design
☐ Implementation ☐ Performance Evaluation
☐ Research ☐ Simulation ☐ Survey ☐ Testing
☐ Other

This project involved the following areas (mark all that apply) :

☐ AI ☐ Architecture ☐ Databases ☐ Distributed Systems
☐ Graphics ☐ HCI ☐ Languages/Compilers ☐ Networks
☐ Operating Systems ☐ Software Engineering
☐ Theory/Foundations ☐ Webware ☐ Other

This project used the following languages/tools (mark all that apply) :

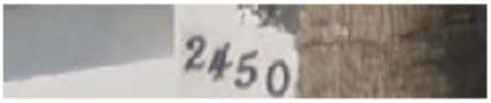
☐ C ☐ C++ ☐ Assembly Language ☐ LISP/Scheme
☐ Java/C# ☐ Perl/Python/PHP/Tcl/Tk/JavaScript ☐ Ruby ☐ Other

This project used the following hardware (mark all that apply) :

☐ Macintosh ☐ PC/Windows ☐ PC/Linux
☐ CS Unix ☐ CCC Unix ☐ Other

Comments - Project strengths/weaknesses/publications/other:

Complete the CAPTCHA challenge then select Submit Evaluation:



Type the text

Privacy & Terms

Submit Evaluation | Clear fields

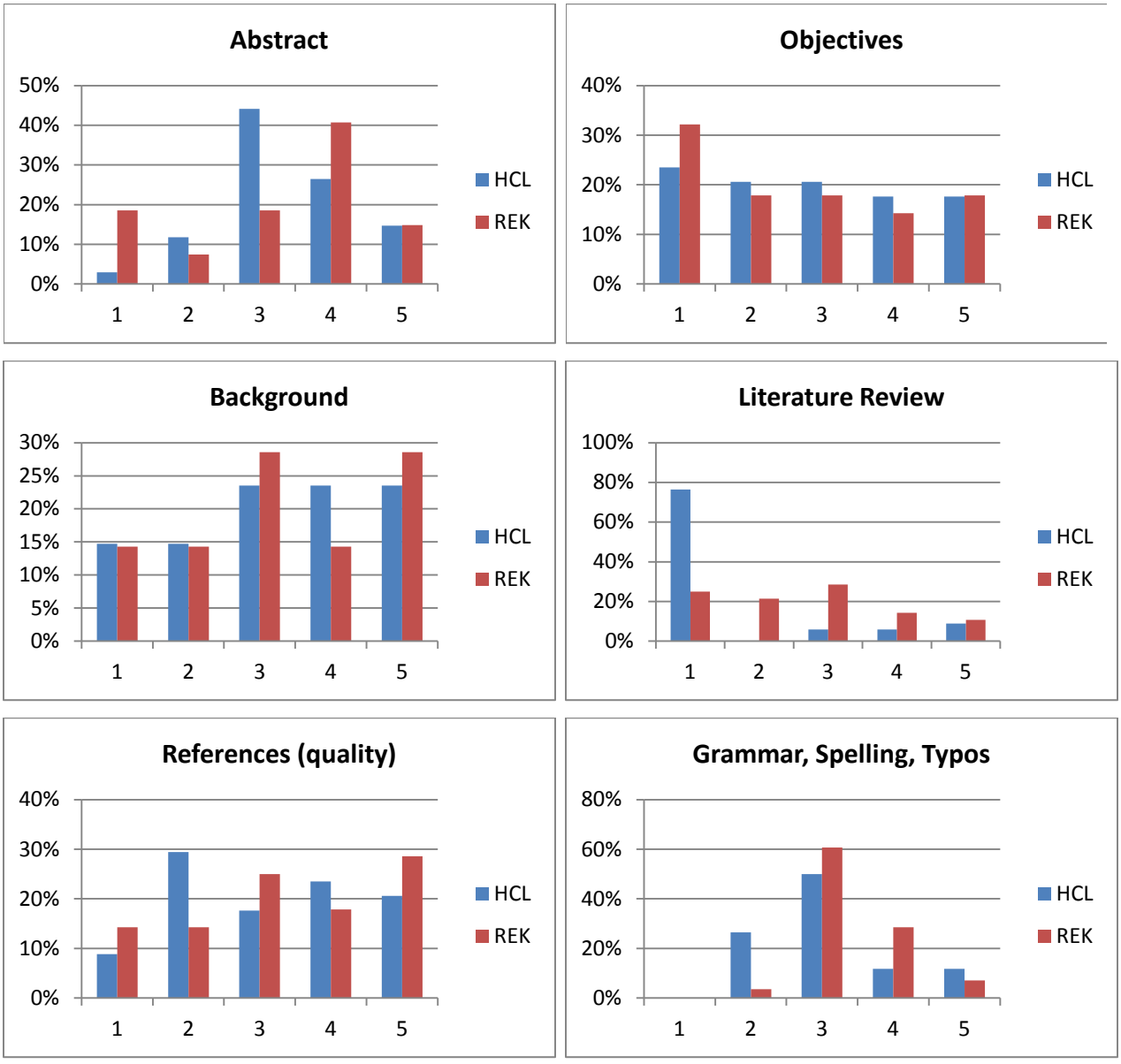
WPI CS

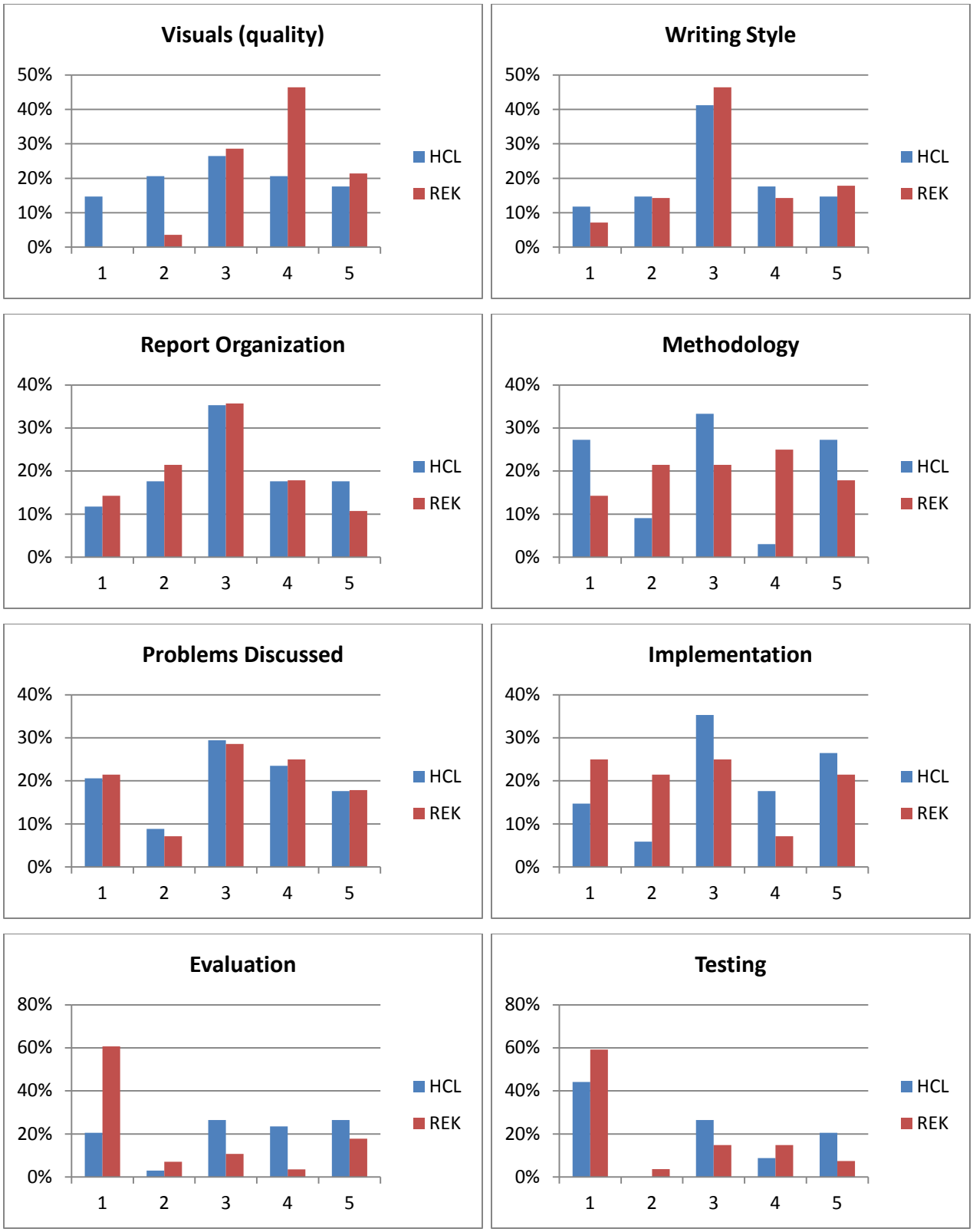
CS Accreditation Committee / Fri May 22 15:49:40 EDT 2015

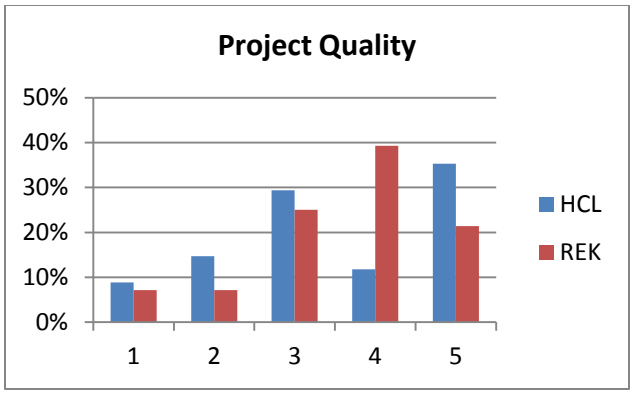
Appendix C DISTRIBUTION OF REVIEW CRITERIA BY TEAM MEMBERS

The following histograms show the distribution of review ratings for each criterion and the two team members. The horizontal scale represents the five-point scale for ratings, and the vertical scale represents the percentage of reviews receiving that rating. Percentages are calculated based on the total number of reports reviewed by each reviewer. The reports reviewed in common therefore are counted twice.

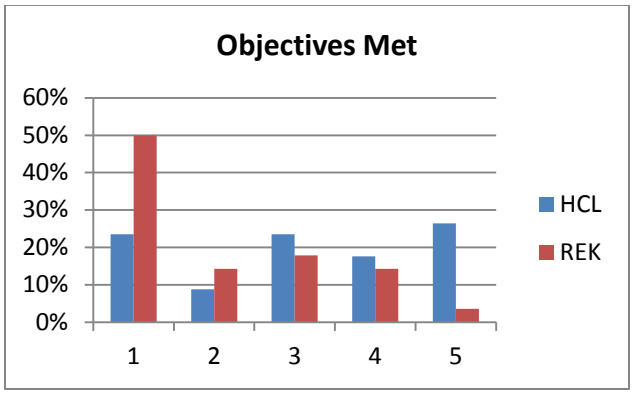
Note, for example, that there was a lot of consistency in the distribution of reviews according to the criteria *Objectives*, *Writing Style*, *Report Organization*, and *Report Quality*. On the other hand, there was considerably less consistency in criteria such as *Literature Review* and *Project Quality*.

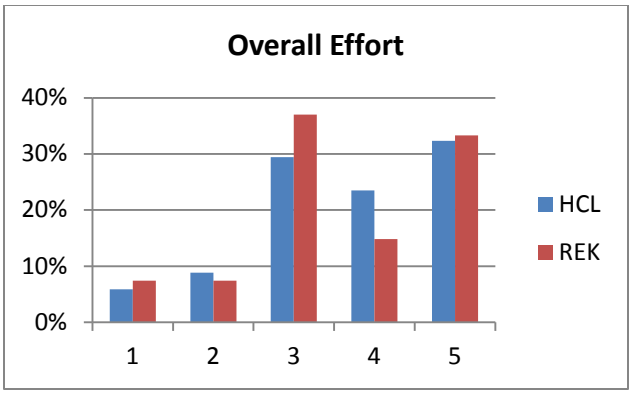


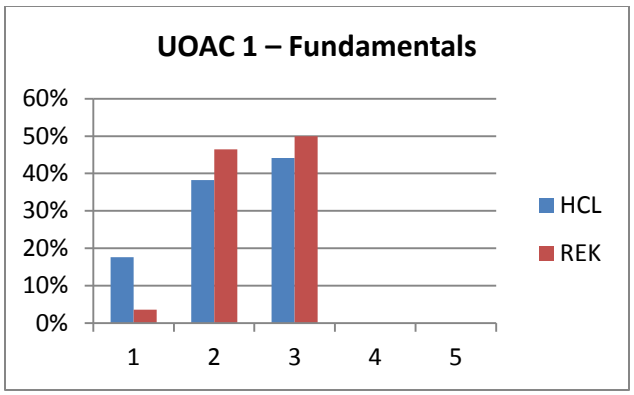

Visuals (quality)
50%
40%
30%
20%
10%
0%
1
2
3
4
5
HCL
REK
Writing Style
50%
40%
30%
20%
10%
0%
1
2
3
4
5
HCL
REK
Report Organization
40%
30%
20%
10%
0%
1
2
3
4
5
HCL
REK
Methodology
40%
30%
20%
10%
0%
1
2
3
4
5
HCL
REK
Problems Discussed
40%
30%
20%
10%
0%
1
2
3
4
5
HCL
REK
Implementation
40%
30%
20%
10%
0%
1
2
3
4
5
HCL
REK
Evaluation
80%
60%
40%
20%
0%
1
2
3
4
5
HCL
REK
Testing
80%
60%
40%
20%
0%
1
2
3
4
5
HCL
REK

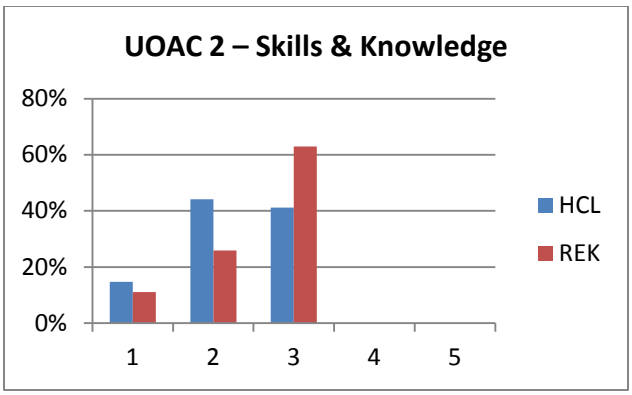

Project Quality
50%
40%
30%
20%
10%
0%
1
2
3
4
5
HCL
REK

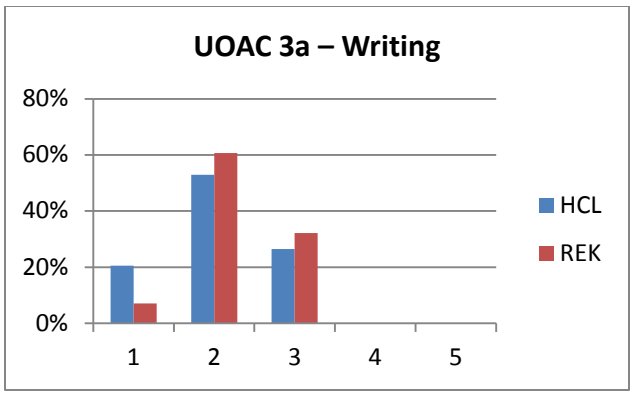

Report Quality
50%
40%
30%
20%
10%
0%
1
2
3
4
5
HCL
REK

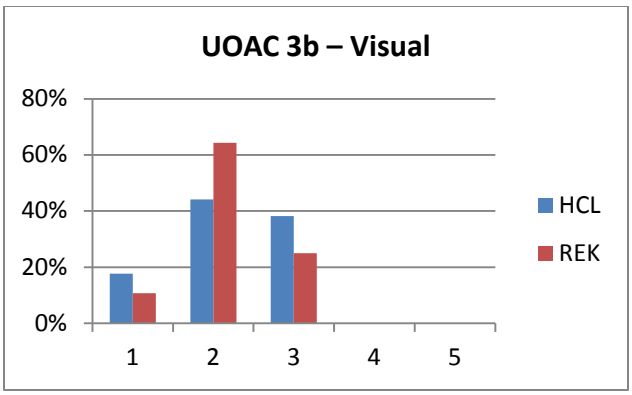

Objectives Met
60%
50%
40%
30%
20%
10%
0%
1
2
3
4
5
HCL
REK

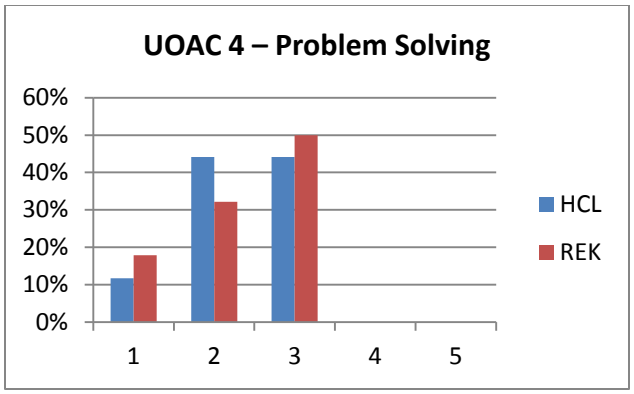

Overall Effort
40%
30%
20%
10%
0%
1
2
3
4
5
HCL
REK

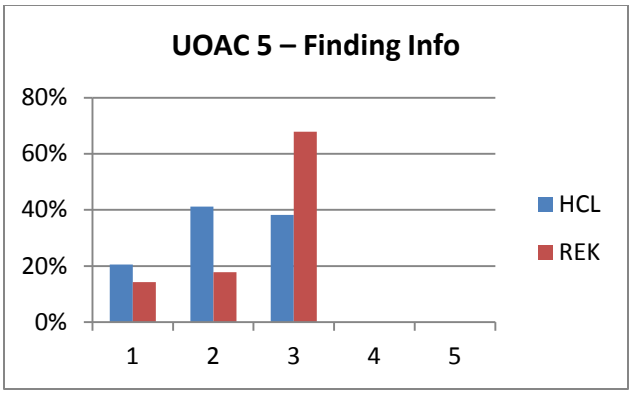

UOAC 1 – Fundamentals
60%
50%
40%
30%
20%
10%
0%
1
2
3
4
5
HCL
REK

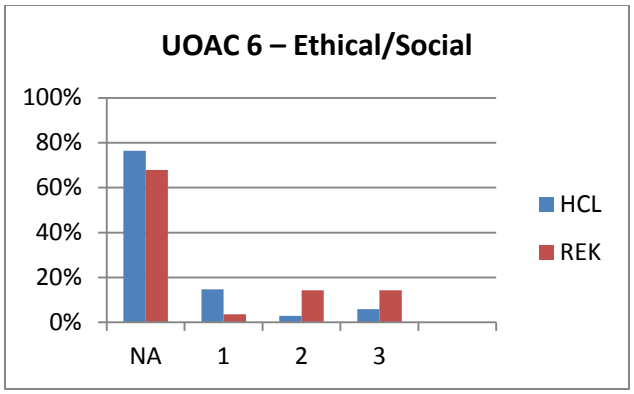

UOAC 2 – Skills & Knowledge
80%
60%
40%
20%
0%
1
2
3
4
5
HCL
REK

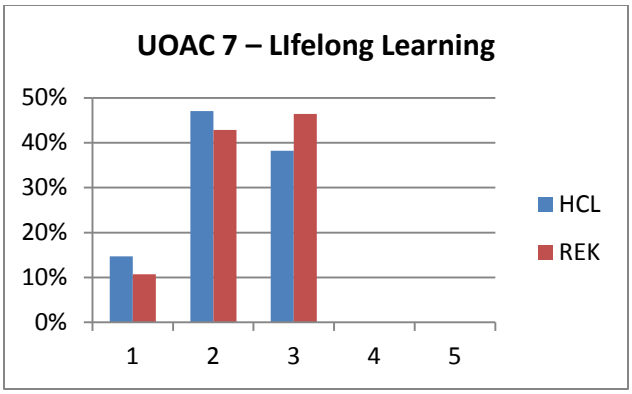

UOAC 3a – Writing
80%
60%
40%
20%
0%
1
2
3
4
5
HCL
REK


UOAC 3b – Visual
80%
60%
40%
20%
0%
1
2
3
4
5
HCL
REK


UOAC 4 – Problem Solving
60%
50%
40%
30%
20%
10%
0%
1
2
3
4
5
HCL
REK


UOAC 5 – Finding Info
80%
60%
40%
20%
0%
1
2
3
4
5
HCL
REK


UOAC 6 – Ethical/Social
100%
80%
60%
40%
20%
0%
NA
1
2
3
HCL
REK


UOAC 7 – LIfelong Learning
50%
40%
30%
20%
10%
0%
1
2
3
4
5
HCL
REK