|  |
| --- |
| **CSM**  **MS Program in Computer Science**  **Department of Computer Science** |
| **Annual Assessment Report for AY 2015-16** |

|  |
| --- |
| 1. **What learning outcome(s) did you assess this year?** |
| In AY 2015-16, we have assessed the following learning outcome:  A.1. Analyze the effects of choice of algorithm and data structure on correctness and performance.  B.1. Analyze, design, implement, debug, verify, and validate software to meet specified requirements. |

|  |
| --- |
| 1. **What instruments did you use to assess them?** |
| We used direct method A.2 (Evaluate software design and implementation ability) in CSci 256 (Wireless Communications and Mobile Computing) and CSci 291T (Bioinformatics). Please note that originally the department proposed to assess CSCi 250 in AY 2015-2016. The course was not offered and hence CSci 256 and CSci 291T are selected for assessment.  The criterion is that “A score of 0-5 is given for each item on the rubric. It is considered acceptable that 70% of the evaluated projects receive an average of 3.5.” |

|  |
| --- |
| 1. **What did you discover from these data?** |
| **a. CSci 256 (Wireless Communications and Mobile Computing)**  For CSci 256, a total of 10 graduate are evaluated on their programming projects. 10 students are divided into 3 groups on research related projects. The assessment concentrated only on programming skills. The evaluation scales from 1-5 based on the attached form. Non-functional specifications and presentation are not included in the rating as non-applicable metrics. The results are as follows:   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **Completion** | **Correctness** | **Documentation** | **Overall** |  |  |  |  | | 4.00 | 3.33 | 3.40 | 3.55 |  |  |  |  | | 4.33 | 4.33 | 3.80 | 4.09 |  |  |  |  | | 4.00 | 3.67 | 3.67 | 3.64 |  |  |  |  |   As shown in the above table, student scores range from 3.55 to 4.09 overall, with all groups above 3.5 (70%). A closer examination shows that all groups received higher than 4.0 rating in the area of “completion”, two groups received above 3.5 in the area of “correctness”, and two groups received above 3.5 in the area of “documentation”.  We further analyze student performance as follows:   * Strength   + All groups were able to complete the assigned projects and meet or even exceed functional specifications, which is considered very good quality for research projects.   + The programming projects involved a whole lifecycle of software development: studying online tutorial, install and configure software, implementation, and testing. All groups demonstrated sufficient competency on “uses software tools and computing resources correctly and efficiently”.   + Two groups (or 7 students) demonstrated a clear understanding the problems and alternative solutions. * Weaknesses:   + Documentation is the weakest area with only one group meeting the 70% requirement. Most groups mainly focused on completing the projects and ensuring the correctness of the results, therefore spending minimum efforts on documentation.   + Some codes lack sufficient testing and therefore the results may have to be further validated.   + Error checking is not sufficient in some programs.   Overall, we are satisfied with the performance of all groups. Being able to make progress on research projects demonstrated the quality of students in programming and problem solving skills. We plan to make the further improvement in Spring 2017 and following semesters:   * Make documentation a required part of the grading, especially, “explain the purpose of each function/module”, “thorough and organized testing”, and “answers/solutions are properly labeled/commented”. * Emphasis the importance of error checking in programs * Assessment of teamwork in programming and identify further areas for improvement.   **b. CSci 291T (Bioinformatics)**  For CSci 291T, during this assessment period, there were a total of 10 graduate students enrolled in CSci 291T in AY15-16. The assessment concentrated on software design and implementation capabilities. The assessment used the SOAP rubric form designed by the Computer Science Department. Students were assessed by ten items. These items are grouped into four categories, i.e., Completion of programming task, Correctness, Documentation and Presentation, and each item is scaled with very poor (score 1) poor (score 2), average (score 3), good (score 4) and excellent (score 5). The assessment form is attached at the end of the report.  Three most serious programming assignments were assessed. The programming assignments are for individual work and each student’s score is computed as the average score from those three selected programs. Students’ scores range from 2.86 to 4.23. The average is 3.2. Five out of 10 students (50%) earned above 3.2; among them two students showed extremely good scores (i.e., 4.23 and 3.93); and the worst score is 2.86. Note the score mentioned above is done by assessing and averaging each student’s ten items from the assessment form.  On the other hand, for each of the ten items assessed on the rubric, the average score (from 10 students) ranges from 2.70 to 3.80. The assessment categories showing the strongest and the weakest average scores are “Completion of Programming Task” (3.65) and “Documentation” (3.04), respectively.  Because “Demonstrates synthesis of solutions and creates alternatives by combining knowledge and information”, “Demonstrates a clear understanding of how various pieces of the problem relate to each other and the whole”, and “Uses software tools and computing resources correctly and effectively” (i.e., 3 items of “Completion of Programming Task”) are to assess A.1. objective (Analyze the effects of choice of algorithm and data structure on correctness and performance) and major parts of B.1 objective (Analyze, design, implement, debug software), from the data shown above, CSci 291T students fulfilled the objective of A.1 and major parts (i.e., analyze, design, implement, debug software) of B.1 with high score. Similarly, students earned average score on “Program meets its functional specifications and expectation of advisor” that is used to assess B.1’s “validation of software to meet specified requirements.”  However, the Department also found two weak parts (i.e., error handling and verification) when assessing objective B.1. The Department plans to educate and encourage students to address clearly the following items from the rubric:  1) “Program contains error checking code and handles all special cases.”  2) “Thorough and organized testing has been completed and output from test cases is included in documentation.”  In summary, the assessment of programming skills for both CSci 256 and 291T identified exception handling and testing are the weaknesses among graduate students. Coincidentally, these weaknesses are also found in undergraduate assessment this year. We will summarize our plan of improvement in Section IV.  **c. Discussion of Student Strength and Weakness**  On July 15, we discussed the strength and weakness of graduate students during the department annual retreat. The results are as follows.  **Strength**   * With very low admission acceptance rate (7%~16% for these two years), the preparation and background of recently admitted students are better. Disqualification and probation rates have been drastically decreased. * After CSci 297 is introduced, students are more serious about their projects/theses and spend more time on looking for solid and novel topics. The quality of CSci 298/299 may gradually improve.   **Weakness**   * Some graduate students’ programming capabilities do not seem to be at the level of master students. They are encouraged to take CSci 250, look for practical projects on campus, or internship opportunities to enhance their background. |

|  |
| --- |
| 1. **What changes did you make as a result of the findings?** |
| The two weaknesses discovered above honestly have become a culture among Computer Science students almost everywhere. Students, both undergraduate and graduate, always prefer and emphasize more on designing and implementing software rather than conducting sufficient testing and error handling. Another reason is that *traditionally* (but not anymore) in software industry, usually quality engineers (i.e., testers) get less pay than software engineers. CSCi 150, 152, 250, 252, 253 are software engineering courses designed to educate good software engineering principles and practices and hopefully to diminish this culture and overcome the misunderstanding. Although none of them is a required course to our undergraduate and graduate students, these courses have been very popular (e.g., in Fall 2016, CSci 150 has 67 students, CSci 250 has 14 students and CSci 253 has 38 students). We hope that students may gradually apply such good principles and practices in their programming assignments of various courses.  In order to improve the weakness, we plan to make the following changes:   * CSci 152 will cover more software testing topics and advocate the importance and advantages of testing and exception handling to all students. We will monitor the progress and introduce any necessary update before next assessment for programming skills. * The Department will discuss whether to include CSCi 250 as an extra required/core course. * As usual, we will continue to disseminate on-campus and off-campus job opportunities to graduate students. We will make more connection with campus’ CTS for potential practical projects that may be converted into CSCi 290 or CSci 298. |

|  |
| --- |
| 1. **What assessment activities will you be conducting in the 2015-16 academic year?** |
| During the next academic year, we will work on three assessment methods: |
| Method A.4 Evaluate writing techniques in CSci 298/299 (for SLO C.1) |
| Method B.4 Discussion of Student Strength and Weakness |

|  |
| --- |
| 1. **What progress have you made on items from your last program review action plan?** |
| The department has made the following progress on MS action plan.  **a) Rebuild the faculty**  **Progress**: we recruited two tenure track faculty: Dr. Jin Park and Dr. Cui Lin in Fall 2011. Dr. Cui Lin resigned in December 2013. Dr. Shigeko Seki has retired starting Fall 2014. The department currently has only five tenured and tenure track faculty (Brent Auernheimer, Ming Li, Shih-Hsi Liu, Jin Park, and Todd Wilson), one 3-year temporary faculty (Prudence Lowe), and three part-time temporary faculty (David Ruby, Santanu Banerjee, and Dhanyu Amarasinghe). The department has been approved to conduct a tenure track search in AY 2016-17 in the area of operating systems, cloud systems, distributed systems, and/or security.  **b) Adding laboratory and classroom facilities**  **Progress**: The department has refreshed McF 201 computer lab with 30 new Windows PCs in July 2015. The department has been approved to convert McF 205 to a hybrid lecture/computer lab by placing 12-15 PCs. Remodeling of McF 205 will start in Fall 2016. Meanwhile, more Computer Science graduate courses (e.g., Csci 250, Csci 253, Csci 256, etc.) have been scheduled in IT 104, which partially alleviate the needs for more lab facility access.  **c) Re-examining culminating experiences**  **Progress**: CSci 297 (Comprehensive Exam, 3 units) and CSci 201 (Colloquium, 2 units) have been approved by university’s curriculum committee in May 2016. The Department starts to offer CSci 297 in Fall 2016 and will offer CSCi 201 in Spring 2017.  **d) Graduate prerequisites and admission requirements**  **Progress**: The graduate committee has discussed improvement on the admission requirements. Starting Fall 2015, we have significantly improved our admission standard as follows:   * Students who needs lower division courses such as CSci 40, 41, and 60 will not be admitted (except those very potential domestic applicants who have working experience in Computer Science industry or have outstanding scores in Computer Science related courses) * Students who need more than 2 prerequisites will be given low priority for admission   As a result, for Fall 2016 admission *only*, we admitted 92 students out of 541 students (17% acceptance rate overall. Domestic student acceptance rate: 22.22% and international student acceptance rate: 16.73%). Among 28 students checked-in, only 9 students require CSci 117 prerequisite in Fall 2016. The arriving rate is 30.43% (28 out of 92), which is about 5% higher than those in Fall 2014 and 2015.  We conclude that the new standard significantly improved the quality of graduate students starting Fall 2015. It will help most students graduate within 2 years.  **e) Assessing course and program**  **Progress**: The Department assessed CSCi 256 and 291T this AY. Due to a wide range of different topics in Computer Science, the Department may select a few more regular courses and introduce a special rubric in order to meet the needs to assess these courses.  **f) Graduate retention plan**  **Progress**: The department continues to support graduate students with TAs and GAs. The department recently established two endowed scholarships and one of them is for graduate students. The department also established a CSCI Student Club with president and many active members were graduate students.  Recently, the department has worked together with local companies to provide job opportunities for students. Many graduate students received part-time job employment during summer or regular semesters. |
| In addition, the reputation of our graduate program plays an important role in improving retention. Our recently graduated students Andy Clifton accepted a tenure track position at Fullerton College and will start his first semester in Fall 2016. |