|  |
| --- |
| **CSM**  **BS Program in Computer Science**  **Department of Computer Science** |
| **Annual Assessment Report for AY 2016-17** |

|  |
| --- |
| 1. **What learning outcome(s) did you assess this year?** |
| In AY 2016-17, we have assessed the following learning outcome:  A.4. Capstone student projects (e.g., CSCI 150/152/198) provide a strong indicator for the student learning outcomes listed in Section 2 of SOAP. Such projects include software engineering activities such as team arrangements, requirements/specifications, design, implementation, testing, tools, schedule, and budgets of a capstone project.  **Criteria**: A score of 0-5 is given for each item in the rubric. It’s considered acceptable that at least 70% of the evaluated projects receive an average score of 3.5. |
|  |

|  |
| --- |
| 1. **What instruments did you use to assess them?** |
| We used direct method A.4 (Capstone project report) in CSci 152 (Software Engineering). Since CSci 152 does not involve real budget, we did not evaluate this item. As for team arrangements and schedule, students used Github “Projects” tab to schedule and arrange tasks among teammates. Instructor reviewed those records and considered them as parts of overall project performance. In summary, the following items are evaluated: Requirements, design, implementation, testing, tools, demo, and overall performance.  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.”  We also conducted BS Program Exit Survey. |

|  |
| --- |
| 1. **What did you discover from these data?** |
| * **CSci 152 (Software Engineering)**   CSci 152 is the second undergraduate course of the sequence of Software Engineering area. In CSci 150 we introduced fundamental Software Engineering concepts and terminologies. Team projects are also an important component of CSci 150. Students use Github for version control. They also use some other features provided by Github to upload their artifacts of requirements, design, test cases, schedule and task responsibilities, among others. In CSci 152, students continue to use Github, but lecture contents are more advanced. We cover design and testing topics in depth.  A total of 31 undergraduate students are evaluated on their CSci 152 team/term projects. 31 students are divided into 8 groups, each with 3 to 4 students. All the projects are Web-based using a variety of platforms, languages, and techniques. For example, there are two Web-based gaming projects mainly using JavaScript (Dungeon Crawler and Platformer Game); two projects (Taste Test and NoteOrious) use Google’s GO language along with some front-end languages (e.g., JavaScript, CSS, HTML5); three projects use PHP and related technologies/platforms (Grab Hub, Fresno State Buy N Sell, Stories project); and one uses Python (Journal project). Some projects also involved database design and management.  The evaluation scales from 1-5, from worst to best. Non-functional specifications are not included in the rating as non-applicable metrics. The results are as follows:    As shown in the above table, overall project scores range from 4.08 to 4.69, with all groups passed 3.5 (70%) minimum requirement. A closer examination shows that all groups received higher rating in the areas of “Demo”, “Requirements” and “Tools” with scores all above 4.40. On the other hand, project scores are lower in “Design” and “Testing”.  With the above results and comments from audience, we further analyze project performance as follows:   * Strength   + Students followed RUP or eXtreme programming (XP) for software development. They incrementally developed and integrated software and a working version is available for demo at the end of semester.   + Students were familiar with Github, Slack, Trello, IDE(s) for their own projects. These tools/IDEs are popular in software industry, which reflects one key benefit of a Capstone course -- bridge between academia and real-world industry.   + Students presented requirements /documents using UML diagrams along with use case descriptions and stored them on Github Wiki page. This convenient setting and easily understood diagrams allowed stakeholders and audience to understand requirements quickly and easily.   + Students obtained knowledge of different kinds of testing and testing tools. They spent time to find suitable testing tools and used them for their own projects. For example, Dungeon Crawler used Mocha testing framework. Fresno State Buy N Sell and the Journal projects used PHP and Python testing frameworks (Pytest), respectively. It is worth mentioning that the Journal project reaches over 90% of code coverage. Also, Fresno State Buy N Sell team also exercised blackbox testing on browsers, which will be mentioned later. * Weaknesses:   + Due to time constraint, some projects may not follow good design principles even though students learned them from lectures.   + Similarly, due to time constraint, all projects mainly exercised unit testing. Only Fresno State Buy N Sell team exercised browser testing using NetRunner to test different versions of Internet Explorer, Google Toggle Device Toolbar to simulate various devices and LoadImpact for stress testing. Also, even though code coverage is covered in lectures, few teams presented the percentage of coverage, or the percentage is not high enough.   + Although some teams claimed they adopt XP, they did not introduce test cases first nor these test cases correspond to requirements consistently. Namely, the test cases they generated cannot fully guaranteed that software was delivered as required (we cannot tell whether features may be missing or overall software is delivered as required if test cases are not linked to requirements correctly and precisely.)   + Continuous testing was not observed during semester. Students still tended to conduct software testing until very last minute.   It is interesting to find that although CSci 152 covers software design and testing in depth, the evaluation results are among the lowest. One possible reason for this is that the advanced topics raise the awareness of good software design and necessity and usefulness of software testing. Students now understand what makes good software design and testing and hence they have higher standard on these two evaluated items and thus graded other teams with lower scores.  Although there are some weaknesses identified, the performance of most teams is satisfactory. It can be easily observed that students spent more time to introduce quality code and introduce meaningful test cases during the semester. We will address how to further improve the weaknesses in the following section.  **b. Exit Interview**  In May 2017, we invited all graduating students to complete a CSCI BS Program Exit Survey. Five responses (three transfer students and two freshmen) were received. A total of 15 questions were asked. Analysis of these responses is summarized as follows:   * Most students graduate within 5-6 semesters. * Most students wanted to focus on Computer Architecture, Software Engineering, and System Software. * Most ended up with focusing on Software Engineering, Web Development, Artificial Intelligence, and Networking. * Artificial Intelligence, Networking, Computer Architecture, Assembly Languages, are among the courses students enjoyed most. * Students are mostly satisfied with the program. * Students listed strength such as core curriculum and learning of theories. * Students listed weakness such as understaffed, not enough courses, and lacking of practical projects. |

|  |
| --- |
| 1. **What changes did you make as a result of the findings?** |
| During last assessment cycle, Csci 152 was also picked for assessment. In the previous report, we identified students were more into programming rather than testing. Because of this, we introduced more software testing topics this academic year. The results of this cycle can easily tell that our emphasis of software testing and good software design have been gradually integrated into students’ “programming mindset”, although as indicated earlier, there are still important weaknesses in software testing. We will continue to emphasizes:   * The importance of the traceability between software requirements and testing. Tools may be introduced to assist students on this; * Remind students the importance of continuous integration and testing. Milestone meetings will be scheduled more frequently to check project progress, especially continuous testing. * Remind students to apply both blackbox and whitebox testing to their projects. Also, remind students that unit testing should also work with other testing approaches. A variety of testing approaches mixing together may further increase the confidence of software quality. |

|  |
| --- |
| 1. **What assessment activities will you be conducting in the 2017-18 academic year?** |
| During the next academic year, we will work on three assessment methods:  Method A.2 Programming Projects  Method A. 3 Course Presentation  Method B.3 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 BS action plan. We have successfully completed our BS five-year program review and received final report in July 2017.  **a) Rebuild the faculty**  **Progress**: we recruited a new tenure track faculty, Dr. Hubert Cecotti, in Fall 2017. The department has been approved to conduct a tenure track search in AY 2017-18.  **b) Develop new assessment plan (SOAP) for B.S., along with supporting instruments**  **Progress**: we’ve updated our BS SOAP in AY 2015-16. The department will discuss further improvement suggestions and make update during AY 2017-18.  **c) Develop CSci 100 for GE Area IB (course title: Introduction to Computational Science)**  **Progress**: the course was successfully offered in Fall 2013. It has been offered every Fall semester.  **d) Establish an Advisory Board**  **Progress**: The department faculty has selected a list of eight potential candidates for an Industry Advisory Board. It is currently pending approval by the college.  **e) Develop Internship and funded-Project procedures**  **Progress**: Dr. Todd Wilson and Dr. Ming Li have been involved in the university Internship Faculty Learning Community (FLC) to understand issues, policies, procedures and required documents for setting up an internship program. Dr. Ming Li has met with Dr. Jaime Arvizu, the Director of the newly established CSM Advising and Resource Center (ARC) and look forward to work with ARC on this effort within the next academic year. In addition, Internship (CSci 196) will be one of the elective courses in the newly proposed CSCI BS curriculum.  **f) Curriculum reviewed with respect to ACM and IEEE Computer Society curriculum recommendations, and ABET accreditation criteria.**  **Progress:** The department faculty understand that it is critical to improve student success. We have established a formal department curriculum committee with members being all full time tenured and tenure track faculty.  The department curriculum committee has studied the ACM Curriculum Recommendation 2013 as a group and proposed a new CSCI curriculum in December 2014. We have received feedback from related programs on the new curriculum and will submit to college curriculum committee for review in AY 2017-18.  **Progress in other items include:**  • Allocation of four faculty research labs.   * Grading support for faculty teaching classes with large enrollment (over 48 students).   • Lab refreshment for computer lab McF 201 in July 2014.   * McF 205 has been converted to a hybrid computer lab with capacity of 28. * Science II 258 has been converted to a department conference room. |