



To: Mark Weiss, Associate Director, SCIS

From: SCIS Undergraduate Committee:

Cristy Charters, Debra Davis, Tim Downey, Radu Jianu, Pat McDermott, Jai Navlakha,
Norman Pestaina, Nagarajan Prabakar (Chair)

Date: 11/3/14

The SCIS Undergraduate Committee has completed its review of the 2011-2013 assessment report for BS in CS. The committee's recommendations are included in the attached document.

Undergraduate Committee's Recommendations on the 2011-2013 Assessment Report

In this document, the relevant sections of the assessment report are reproduced (in italics) to facilitate referencing to the assessment report and the recommendations are highlighted below the sections. The following acronyms are used throughout the report:

AC – Assessments Coordinator
CpE – Computer Engineering
CS – Computer Science
SAC – Subject Area Coordinator
SE – Software Engineering
UGC – Undergraduate Committee

I. RECOMMENDATIONS

A. Recommendations of the Subject Area Coordinators

Subject Area: Computer Organization (SAC: Nagarajan Prabakar)

COP 4610: Enforce the prerequisite Programming III for all students enrolled in the course (including Computer Engineering majors). Repetition of this problem for several years requires ECE Undergraduate Program Director to enforce this prerequisite. Also, the faculty needs to specify clearly about the expected C proficiency at the very first class. Furthermore, students may be given a quiz (about 10-20 short questions) in C during the first week of the term so that each students can gauge his/her ability to cope with the projects.

CpE has changed their program with the Data System Software concentration which includes COP4338. Enforce COP4338 prereq requirement for COP4610 through advisors.

Subject Area: Foundations (SAC: Xudong He)

COT 3420: Two instructors who taught this course noted that the students did not have adequate preparation (it was between “deficient” and “non-existent”) for the class. One commented on the continual deterioration of student quality and lack of motivation. Another commented on the students’ lack of understanding of induction and essential concepts of propositional logic that mandated sacrificing the coverage of first order logic. One possible solution to address these concerns is to offer our own Discrete Math course, which covers some materials such as propositional logic and induction, thus complements COT 3420.

This problem continues to exist.

Use learning outcomes for CS majors – MAD2104 & COT3420.

Subject Area: Programming (SAC: Norman Pestaina)

COP 2210: There are no Course Outcomes Survey data available for this reporting period. The corrective action has been already initiated. SAC recommends that (1) the course outcomes survey must be re-implemented expeditiously, and (2) it might be useful to attempt a correlation between the ratings of the value of COP 2210 course outcomes and the students’ written suggestions on the content of the course.

- To be reviewed with all programming courses

COP 3337: Classroom instruction for this course could be supplemented by providing resources such as closed labs or peer tutoring, or some other mechanism to provide students with additional opportunities for mastering the course outcomes. Furthermore, there may be a need to synchronize the outcomes of COP 2210 with the prerequisites of COP 3337 in order to afford students a smoother transition.

- To be reviewed with all programming courses

COP 4338: The original course outcomes are still listed in the common syllabus for this course. The syllabus must be revised to reflect the revision of the course outcomes.

- Outcomes have been revised and need to be followed up.

COP 4520: The SAC feels that the scarcity of data, and the fact of having only a single offering of this course in the current Assessment Cycle, do not lend to a high degree of confidence in the analysis. Nonetheless, it may be worth considering whether the course prerequisites, COP 3530 and CDA 4101, provide adequate preparation; perhaps students taking this course lack sufficient (academic) maturity.

UGC will revisit after the next course offering.

Subject Area: Software Engineering (SAC: Masoud Sadjadi)

CEN4010: The pre-test based on this course material given by an instructor to the Senior Project class showed that the students are under-prepared with respect to their theoretical and practical knowledge of Software Engineering. A majority of our students have learned how to hide behind their teammates in group projects and pass SE I on the shoulders of their friends. Therefore, I strongly suggest that each individual student becomes responsible to perform his/her share of the project and practice all the different software engineering activities by himself/herself.

The following suggestions are made for the instructors of this course.

- As the process of choosing projects by students becomes time consuming and it may not be easy for the instructor to understand all the details of all the projects, I suggest that one or more projects to be chosen and predefined by the instructor before the semester starts. Note that the total number of the functional requirements for all the chosen projects should be equal or greater than the number of students enrolled in the class.
- Each student should be randomly assigned to one (or more) specific functional requirement(s) of one of the chosen projects.
- Each student must be responsible to practice all the different software engineering activities using his/her assigned functional requirement(s).
- As the load for the software engineering activities are more than enough for a 3-credit course, and as project management is taught in SE II, the students should not be expected to manage their own projects too. The instructor of the course, or his/her TA(s), should assume that role and should make sure that all the students are on time with their tasks. Also, they should have alternative plans, in case some of the students fall behind or ahead of the schedule of the projects.

- *To make sure that each individual student gains the required knowledge and knows how to use it in the assigned project, they all should be given an opportunity to present their assigned work as part of group presentations, after each milestone of the project has reached.*
- *To make sure that each individual student knows how to properly use UML diagrams, they should all be asked to use a UML tool that is approved by the industry and can verify their diagrams. The instructor should ask students to verify their diagrams before submitting their deliverables.*
- *To make sure that all students understand software engineering terms and can use them properly, the definition of such terms should be asked in the mid and/or final tests.*

The course delivery/contents should be discussed among the instructors of this course and provide course change recommendations.

CEN 4021: The following recommendations are made by the SAC.

- *As suggested by one of the instructors also, this course should be only focused on software project planning and management. Therefore, I suggest that we remove the software architecture topic from the syllabus of this course and cover it in details in SE I.*
- *This course needs to be more applied and become more practical. For this, I suggest that we offer this course during the same semester as SE I is offered so that the students from this class can become project managers of the SE I projects. To address the potential issues that may arise, I suggest that we do the following:*
 - *It would be best if the instructor for both SE I and SE II to be the same person. If this is not possible in a semester, then the two instructors from SE I and SE II should meet before the semester starts and plan on how to synchronize their assignments.*
 - *Students in SE II should only gather data from students in SE I and must NOT give direct feedback to the SE I students. Basically, SE II students do NOT actually manage the SE I projects. Instead, SE II students provide the data and outcome of their work to the instructors of SE II for their assignments to be evaluated and receive their grades. Also, they should provide their work to the instructor of SE I for his/her use to actually manage the projects. This should greatly help the SE I instructor with the project management task, giving incentive to allow SE II students to contact SE I students.*
 - *The number of students in SE II may be more than the number of projects in SE I; therefore, more than one student from SE II may be assigned to gather data from the same SE I project. To limit the number of times that SE I students may be interrupted by SE II students to gather project status data, the SE II students assigned to the same SE I project may work together to collect the data, but they must process the data independently to make sure that they learn all the project management activities through practice.*

The course delivery/contents should be discussed among the instructors of this course and provide course change recommendations. Consider the course offering frequency and required/elective aspect of SE-I and SE-II in the recommendations.

CEN 4072: The following recommendations are made.

- *This course needs better software examples for students to practice different testing tools.*
- *Students in this course need the knowledge of software engineering activities.*
- *Based on the above two observations, I make the following suggestions:*
 - *SE I should become a pre-requisite for this course.*
 - *Students can use their own SE I projects as the example to practice different software testing tools.*
 - *When the software from SE I is not sufficient for practicing a software tool, the instructor should provide other software project examples from the past SE I projects done by other students.*

The recommendations need to be discussed with the instructors who teach this course. Also, consider including CEN 4010 as a co-req to CEN 4072.

CIS 4911: The following observations and recommendations are made by the SAC.

- *As this course has gone through major changes over the past two years, some students have rightfully been confused and at times frustrated with the lack of clear direction and clear expectation from this course.*
- *The pre-test taken in Fall 2013 has indicated that in general our senior project students are under prepared for what is expected of them in this course.*
- *A single semester is too short to finish a major project by groups of 2 to 5 students, especially, if the projects are not assigned at the very beginning of the semester.*
- *Students may hide behind their teammates and may pass the course without earning it.*
- *Many of the students taking this course do not have sufficient teamwork experience.*
- *Many of the students taking this course have not taken SE II and have no software project management experience.*
- *Projects do not seem to be diverse enough to cover all aspects of our curriculum and most projects are very software engineering centric.*
- *Based on the above observations, I make the following suggestions:*
 - *SE I should better prepare the students as suggested before.*
 - *The coordinator (another name for senior project's instructor) must reach out to the SCIS faculty members and SCIS industrial partners before the semester starts and ask for project suggestions. To make sure that the suggested projects are appropriate for the senior project course, the only metric should be whether the project is software intensive or not. In other words, whether the students assigned to this project would need to develop a significant software solution to solve a significant problem or not, as opposed to whether the project is a software engineering project or not.*

- *We do have deliverable templates for Software Engineering and System Centric projects. For all the other possible project types, the coordinator should work with the mentor to come up with some appropriate project deliverable templates to be used by the assigned students.*
- *Students should be assigned to the selected projects on the first week of the semester to get them started as early as possible.*
- *The tentative schedule for the whole semester should be given on the first day of class and the expectation should be clearly explained.*
- *The coordinator or his TA(s) should play the role of the project manager for the assigned projects, freeing the students to only worry about their project activities.*
- *According to the timeline and milestones of the projects, every week (or every other week, depending on the size of the class), each individual student should get a chance to present a progress report as part of a group presentation.*
- *The coordinator should give comments/feedback either verbally or in written form to each individual student with respect to his/her performance and the status of the project after students present their work or deliver their deliverables.*
- *Teamwork should be emphasized as this may be the first time students work in a group setting.*

These suggestions are already outlined in the last page of the detailed course syllabus file of CIS 4911 with specific guidelines. They need to be implemented consistently.

B. Recommendations of the Assessments Coordinator

AC-05: The Course Outcomes ratings (Table 1) for the Value and Adequacy of Coverage of COT 3420 is substantially better in this cycle as compared with the previous one (Value: 4.25 v/s 3.86, Coverage: 3.93 v/s 3.53). However, the Subject Area Coordinator's report clearly indicates that the Course Outcomes are not followed consistently by the various instructors of COT 3420. The content and delivery of this course must be clearly specified and followed by all instructors. Furthermore, as suggested by the AC in the previous report, it would not be untimely to consider alternative implementations of COT 3420 to include knowledge units from applied logic areas, for example artificial intelligence, knowledge-based reasoning, robotics, game playing, etc.

The Associate Director needs to form a committee to consider the course syllabus revision. Also, consider how the revised version would conform with ABET requirements.

AC-06: It is challenging to perform meaningful assessment of Student Outcome a) Demonstrate proficiency in the foundation areas of Computer Science including mathematics, discrete structures, logic and the theory of algorithms using the rubric of the Senior Project class because there are essentially no projects attempted by students that address the relevant topics. The point is made for discussion only; no recommendations are made.

Student deliverables should include the aspects of the existing rubric.

AC-08: It is important to have a consistent style of Course Embedded Assessments. When different instructors conduct this assessment, it is natural that their styles differ. SCIS should make Subject Area Coordinators responsible to maintain the consistency of this assessment. Before designing and conducting the assessment for a particular class, instructors should consult the SAC.

Refer this to the Associate Director and Assessments Coordinator.

AC-10: The Subject Area Coordinator for Software Engineering has made substantial suggestions to improve the whole sequence of courses (CEN 4010, CEN 4021, and CIS 4911). The SCIS undergraduate Committee should discuss them seriously getting the SAC input directly in its deliberations, and then should decide if changes are warranted.

Refer to the recommendations made by the UGC to the SAC suggestions (pages 3-5 above).