

**FINAL ASSESSMENT REPORT**  
**Institutional Quality Assurance Program (IQAP) Review**  
**Software Engineering**  
**Undergraduate Program**

**Date of Review:** March 31 – April 1, 2016

*In accordance with the University Institutional Quality Assurance Process (IQAP), this final assessment report provides a synthesis of the external evaluation and the internal response and assessments of the undergraduate programs delivered by the **Department of Computing and Software**. This report identifies the significant strengths of the programs, together with opportunities for program improvement and enhancement, and it sets out and prioritizes the recommendations that have been selected for implementation.*

*The report includes an Implementation Plan that identifies who will be responsible for approving the recommendations set out in the Final Assessment Report; who will be responsible for providing any resources entailed by those recommendations; any changes in organization, policy or governance that will be necessary to meet the recommendations and who will be responsible for acting on those recommendations; and timelines for acting on and monitoring the implementation of those recommendations.*

**Executive Summary of the Cyclical Program Review of the Undergraduate  
Software Engineering Program**

In accordance with the Institutional Quality Assurance Process (IQAP), the Department of Computing and Software submitted a self-study in January 2016 to the Associate Vice-President, Faculty to initiate the cyclical program review of its undergraduate programs. The approved self-study presented program descriptions, learning outcomes, and analyses of data provided by the Office of Institutional Research and Analysis. Appendices to the self-study contained all course outlines associated with the program and the CVs for each full-time member in the department.

One arm's length external reviewer from the Ontario and one internal reviewer were endorsed by the Dean, Faculty of Engineering, and selected by the Associate Vice-President, Faculty. The review team reviewed the self-study documentation and then conducted a site visit to McMaster University on March 31 – April 1, 2016. The visit included interviews with the Provost and Vice-President (Academic); Associate Vice-President, Faculty, Dean and Associate Dean of the Faculty of Engineering, Chair of the department and meetings with groups of current undergraduate students, full-time faculty and support staff.

The Director of the program and the Dean of the Faculty of Engineering submitted responses to the Reviewers' Report (January 2017). Specific recommendations were discussed and clarifications and corrections were presented. Follow-up actions and timelines were included.

The Final Assessment Report was prepared by the Quality Assurance Committee to be submitted to Undergraduate Council and Senate (December 2017).

### **Strengths**

In their report (April 2016), the Review Team highlighted the following strengths of the program:

- The attrition rate for students in the program is relatively low
- “The Department has been successful in recruiting a number of new faculty members with diverse backgrounds”
- The newly introduced “practice and experience” courses and the capstone project course contribute to a strong experiential learning experience
- The program provides a strong background in hardware-oriented programming skills
- The students are exposed to large variety of programming languages and platforms
- The class rooms and laboratories are well equipped and maintained
- The technical and administrative staff are providing excellent support for the
- The Department has created a Continuous Improvement Committee for monitoring the Software Engineering and Mechatronics Engineering programs

### **Areas for Improvement and/or Enhancement**

The Review Team noted the following areas for improvement in the program:

- The learning outcomes at the program level based on CEAB Graduate Attributes and Indicators are too generic and should be specialized for the Software Engineering program
- The differences between the Computer Science and Software Engineering programs are not clear to students
- Software Engineering students entering Level 2 have less knowledge of programming than Computer Science students entering Level 2. Software Engineering students have noticed that they are thus less prepared than Computer Science students in the courses that combine both groups of students
- The program lacks courses, such as web computing and mobile computing. In the application domain
- The required database course should be moved from Level 3 to Level 2
- It is not clear where software maintenance and re-engineering is covered in the program
- The curriculum map does not distinguish between different levels of design content across the curriculum
- Measurement of teamwork is not adequately addressed
- The co-op program is not utilized in assessing learning outcomes
- The increasing enrolment, very high student to faculty ratio, use of a large number of sessional lecturers, and combining courses with Computer Science and Mechatronics Engineering students have negatively impacted the student learning experience
- There is not sufficient office space for sessional lecturers

- The Department does not have a curriculum committee dedicated to the Software Engineering program

## Summary of the Reviewers' Recommendations with the Department's and the Dean's Responses

### Recommendations

Recommendation	Proposed Follow-Up	Responsibility for Leading Follow-Up	Timeline for Addressing Recommendation
<b>1. The review team encourages the Department to refine the current program learning outcomes into more program-specific learning outcomes. More specific program learning outcomes will enable to Department to better focus its curriculum development as well as allow students to better understand the differences between different Software Engineering program options and the difference between the Software Engineering program as a whole and other related programs, such as Computer Science and Computer Engineering.</b>	Learning outcomes are addressed on two levels. <i>Learning objectives</i> are the targeted course-level learning outcomes; they are the components of the pre- and post-conditions for individual courses. <i>Graduate attributes and indicators</i> are the targeted program-level learning outcomes; the graduate attributes are the same for all Canadian engineering programs, while the indicators are the same for all McMaster engineering programs. Rather than introducing a third level of program-specific program-level learning outcomes, we need to develop a tighter mapping between the program-level indicators and the course-level learning objectives.	CAS Associate Chair for Undergraduate Studies	June 30, 2017
<b>2. The Department may consider adding a list of <i>recommended</i> courses in computer technology and</b>	This recommendation is not feasible since students are admitted into the Engineering 1 program from high school and not directly into the Software Engineering program.		

programming to the formal admission requirements.			
<b>3. The Department may consider adding a list of <i>recommended</i> course electives for first year students who are interested in choosing Software Engineering as their degree program, e.g., COMP SCI 1JC3 and COMP SCI 1XA3.</b>	This recommendation is not feasible since Engineering 1 students have no room in their schedules to take technical electives.		
<b>4. The Department may consider making the Database course available (as a mandatory course) even earlier than third year, e.g., the second half of Year 2. If this is not feasible, consider integrating a basic introduction to the use of databases in a second year practice and experience course, e.g., SFWR ENG 2XB3. The Database course should remain a mandatory component of the Software Engineering program.</b>	Although it would be desirable to have the required databases course in Level 2, this is not feasible since none of the courses currently in Level 2 can be easily moved to later levels. The suggestion by the reviewers to incorporate an introduction to databases in the SFWRENG 2XB3 (Software Engineering Practice and Experience: Binding Theory to Practice) is the most promising way to move the subject of databases earlier in curriculum.	CAS Associate Chair for Undergraduate Studies	June 30, 2017
<b>5. The Department may consider adding courses on Web-based and mobile software engineering to the curriculum, while moving some of the advanced hardware-oriented programming courses to electives</b>	The Embedded Systems program is being eliminated as a separate program. Software Engineering students who are interested in embedded systems will be able to take the current embedded systems courses as electives. As a rule, we are making the specialized Computer Science courses, including COMPSCI 4WW3 (Web	CAS Associate Chair for Undergraduate Studies	June 30, 2017

<b>(or mandatory courses in the embedded systems option).</b>	Systems and Web Computing), available to Software Engineering students as technical electives.		
<b>6. The Department may consider strengthening the treatment of fundamental concepts and methods used in Software Maintenance and Reengineering, e.g., by revising the course description for SFWR ENG 3XA3 to explicitly include this topic.</b>	We agree that the treatment of software maintenance and re-engineering should be strengthened and the best vehicle for doing this is SFWRENG 3XA3 (Software Engineering Practice and Experience: Software Project Management).	CAS Associate Chair for Undergraduate Studies	June 30, 2017
<b>7. The Department is encouraged to provide students with opportunities to gain experience with diverse programming languages and platforms in senior program years, wherever possible.</b>	We agree with this recommendation	CAS Associate Chair for Undergraduate Studies	June 30, 2017
<b>8. The Department may consider defining more detailed, program specific learning outcomes that refine the general CEAB graduate attributes. A refined curriculum map may indicate what learning outcomes are <i>introduced</i>, further <i>developed</i>, and <i>specialized</i> in which</b>	See recommendation 1 above		

<b>courses.</b>			
<b>9. The Department may consider adding assessment strategies for teamwork to project-based courses, potentially using the tool of an “Engineering logbook” or a similar mechanism for assessing team collaboration and communication</b>	We agree that the Department should develop better means for assessing teamwork on projects. The suggested logbook idea, that is currently used in courses such as SFWR ENG 3A04, could be implemented by making logbooks an integral part of all Software Engineering courses	CAS Associate Chair for Undergraduate Studies	June 30, 2017
<b>10. The Department may consider the opportunity of assessing the competencies of students participating in the Co-Op program, for example by adding structured questionnaires for work terms supervisors (employers) and students at the exit points of each Co-Op term</b>	This is a good suggestion, but it needs to be investigated and implemented at the Faculty level	Faculty of Engineering Associate Dean, Academic	June 30, 2017
<b>11. The Department may consider a Design learning outcome for its HCI course</b>	We agree that the post-condition of SFWRENG 4HC3 (Human Computer Interfaces) should include a design learning objective	CAS Associate Chair for Undergraduate Studies	June 30, 2017
<b>12. Given the increasing enrolment, upcoming retirements and the need to reduce class sizes, the Department should continue to recruit new faculty members. Specifically, the</b>	The Department intends to hire as many faculty members, including teaching professors, as the Faculty will authorize	CAS Chair	The period of 2017-2019

<b>Department may consider attaining permission to hire one or two teaching professors, as they can be assigned a higher course load</b>			
<b>13. The Department is encouraged to continue recruiting faculty members that increase the diversity of its faculty complement, e.g., gender minorities and faculty with diverse backgrounds</b>	The Department recognizes that it needs to increase the diversity of its faculty, especially with respect to women. Three of the last five faculty hires in CAS were women. The Department is dedicated to continuing hiring in this direction	CAS Chair	On-going
<b>14. The Department may consider exploring options to increase the availability of spaces for meetings between sessional instructors before and after class. Perhaps a keycard reader can be installed in the shared sessional office, so that sessionals do not depend on a single shared physical key to access the shared office</b>	As the number of sessional lecturers has increased, the need for space for them has also increased. CAS, and the Faculty as a whole, is very short of space. Nevertheless, we will work to provide our sessional lecturers adequate space for their needs	CAS Administrator	December 1, 2016
<b>15. The Department may consider creating and communicating a Web site that maintains detailed information on the software and (drop in) laboratories available to students</b>	This facility already exists. See <a href="http://www.cas.mcmaster.ca/support/">http://www.cas.mcmaster.ca/support/</a> .		
<b>16. The Department is encouraged to continue their efforts</b>	We agree. Our hiring plan for 2016-2017 includes the hiring of a senior faculty with the	CAS Chair	June 30, 2017

<p><b>in recruiting faculty members with scholarly interest in practical / application-oriented aspects of software engineering research and teaching</b></p>	<p>proposal characteristics</p>		
<p><b>17. Measures should be taken to reduce class sizes and limit the number of temporary teaching staff (sessionals) in delivering courses</b></p>	<p>We have reduced the number of required Software Engineering courses combined with required Computer Science or Mechatronics Engineering courses from 13 to 6. This has significantly reduced the average size of the required Software Engineering courses at the cost of increasing the number of courses taught by sessional lecturers. Reducing the intake of students into the Software Engineering program is not an option that the Faculty is able to consider at this time, so the only solution to this problem is to hire more faculty members</p>		
<p><b>18. The Department is encouraged to expand its current mentoring program (for Computer Science students) to students in Software Engineering</b></p>	<p>The mentoring program for Computer Science students has not been very successful thus far due to a pronounced lack of participation by the students. A mentoring program for Software Engineering students is desirable, but careful consideration is needed to find ways to better engage the students and to provide effective mentoring given that the Department's faculty workload is already excessive. We need to develop a successful pilot program for Computer Science before developing such a mentor program for Software Engineering</p>	<p>CAS Chair</p>	<p>June 30, 2017</p>
<p><b>19. The Department should develop a</b></p>	<p>We agree: the Department should develop renewed</p>	<p>CAS Chair</p>	<p>June 30, 2017</p>



mission statement / vision document for its Software Engineering program, including <i>specific</i> program learning outcomes	mission and vision statements for each of its undergraduate and graduate programs		
<b>20. The Department should develop detailed descriptions for all program courses including course-specific preconditions and postconditions (learning outcomes). These should be mapped to the program specific learning outcome</b>	This has been done. What remains to be done is to harmonized the pre- and post-conditions across the program and to improve the mapping of the program-based indicators to the course-based learning objectives	CAS Associate Chair for Undergraduate Studies	June 30, 2017
<b>21. Encourage instructors to incrementally incorporate innovative methods for learning and teaching, with an emphasis on multi-media and flip-classroom teaching methods, in order to decrease faculty teaching load and increase classroom attendance</b>	The previous and current CAS chair has been actively encouraging the CAS instructors to work with McMaster's <b>Paul R. MacPherson Institute for Leadership, Innovation and Excellence in Teaching</b> to improve their teaching and to experiment with new teaching formats	CAS Chair	
<b>22. Establish an Industrial Advisory Board with broad representation from different software engineering related industries to advise the Department on trends, curriculum and strategic planning</b>	The Department is interested in establishing an Industrial Advisory Board with broad representation throughout the computing related industries	CAS Chair	June 30, 2017
<b>23. The Department may consider strengthening the</b>	We agree	CAS Associate Chair for Undergraduate Studies	June 30, 2017

<p><b>input from practitioners in its Continuous Improvement process, e.g., by implementing regular surveys of employers (of co-op students as well as graduates) and alumni.</b></p>			
<p><b>24. The Department should create a curriculum committee with dedicated focus on the Software Engineering program and its options.</b></p>	<p>Our department faculty is too small to support a separate curriculum committee for each of our three undergraduate programs</p>		
<p><b>25. The Department may strengthen the role of employer and alumni feedback to Continuous Improvement Committee and the Continuous Improvement process</b></p>	<p>See recommendation 23</p>		
<p><b>26. The Department may find a way to make aggregate, anonymized data from course evaluations available to members of the Continuous Improvement Committee</b></p>	<p>This will require significant care in order to protect instructors' privacy. It does not appear that per course data can be provided</p>	<p>CAS Associate Chair for Undergraduate Studies</p>	<p>June 30, 2017</p>
<p><b>27. The Department may enhance the way it is communicating software / extracurricular lab services and opportunities to students</b></p>	<p>Communicating effectively with students is becoming increasingly more difficult. For example, email is not an effective way to reach most students. In partnership with CAS student associations, clubs, and societies, we are rethinking how to better engage and communicate with students</p>	<p>CAS Chair</p>	<p>June 30, 2017</p>

**Faculty Response:**

As detailed in the Chair's response, the recommendations in the review have led to a series of discussions within the Department and Faculty focused on such items as reduction of class sizes, being aware of the differences in level 2 between the Computer Science (CompSci) students and Software Engineering (SE) students combined courses, the enhancement of such items as teamwork and mapping/tracking of learning outcomes, the establishment of an Industrial Advisory Board, and issues with a large number of sessional instructors. The vast majority of the recommendations are currently being addressed by the Department and include such items as the splitting of combined CompSci and Software Engineering courses, the on-going development of a more comprehensive curriculum map, and the hiring of teaching-track faculty. Unfortunately, some of the recommendations, such as increasing the course entry requirements to the Software Engineering program cannot be implemented due to the common Engineering 1 entry year at McMaster.

Overall, the dean satisfied with the replies of the Department to the concerns raised by the IQAP reviewers.

**Quality Assurance Committee Recommendation**

**McMaster's Quality Assurance Committee (QAC) reviewed the above documentation and the committee recommends that the program should follow the regular course of action with an 18-month progress report and a subsequent full external cyclical review to be conducted no later than 8 years after the start of the last review.**