Graduate Fields
Definition and Proposal Template

Definition

In graduate programs, field refers to an area of specialization or concentration (in multi/interdisciplinary programs a clustered area of specialization) that is related to the demonstrable and collective strengths of the program's faculty. Institutions are not required to declare fields at either the master's or doctoral level. Institutions may wish, through an expedited approval process, to seek the endorsement of the Quality Council.

Graduate Field Proposal Guidelines

1. Indicate the name of the field being proposed and identify the parent program.

Field: Software Engineering
Parent program: MASc in Electrical and Computer Engineering

2. Provide a description of the field (its intellectual focus, etc.) including the appropriateness and consistency of the field name with current usage in the discipline or area of study.

Software enables technological advances that lead to new, high-performance products and systems in every commercial sector, including medical devices, automobiles, aircrafts, power generation systems, mobile phones, and entertainment systems. As a product and a system's functionality grow, so does the need to efficiently and correctly implement the complex software that enables growth. Software engineering is the application of a systematic, disciplined, quantifiable approach to the design, development, testing, operation, and maintenance of software. Software engineering principles and practices are essential for the development of large, complex, or trustworthy systems. Within the software engineering field of the graduate program, students will significantly expand and deepen their understanding of the theory and practice of software engineering in at least one of the following areas: software requirements, architecture and design, testing, verification, and maintenance. Students will gain the necessary skills to evaluate, adapt and develop software engineering processes, metrics, and tools.

3. Comment on the relationship of the admission requirements for the field to those of the parent program. If the same, describe the program admission requirements. If different, describe the field admission requirements, indicate how they are different from those of the parent program, and provide a rationale for the difference in relation to the focus and learning outcomes of the field.

The admission requirements for the field are the same as those of the parent program. Since these requirements are being slightly amended to include mentioning of software engineering for applicant credentials, please refer to Appendix A of the Major Modification Proposal.

4. Comment on the relationship of the curricular requirements for the field to those of the parent program. If the same, describe the program requirements. If different, describe the field requirements, indicate how they are different from those of the parent program, and provide a rationale for the difference in relation to the focus and learning outcomes of the field.

The curricular requirements for the field are the same as those of the parent program. Since these requirements are being changed in terms of number of courses and breadth requirement, please refer to Appendix A of the Major Modification Proposal.

5. Provide a list of courses that will be offered in support of the field. The list of courses must indicate the unit responsible for offering the course (including cross-lists and integrations, as appropriate), the course number, the credit value, the short course description, and whether or not it is an existing or new course. For existing courses, the frequency of offering should be noted. For new courses, full course proposals are required and should be included in the proposal as an appendix. (The list of courses may be organized to reflect the manner in which the courses count towards the program/field requirements, as appropriate; e.g. required versus optional; required from a list of specified courses; specific to certain concentrations, streams or fields within the program, etc.)

All the courses listed below are optional. They are all existing courses. The supervisor plays an important role in the course selection and will normally encourage students to take these courses.

Course: EECS 5421 3.0
Title: Operating System Design
Short course description: A modern operating system has four major components: process management, input/output, memory management, and the file system. This project-oriented course puts operating system principles into action and presents a practical approach to studying implementation aspects of operating systems. A series of projects are included for students to acquire direct experience in the design and construction of operating system components and have each interact correctly with the existing software. The programming environment is C/C++ under UNIX.
Number of offerings (in last five years): 4

Course: EECS 5441 3.0
Title: Real-Time Systems Theory
Short course description: Specification and verification techniques for real-time systems with many interacting components. Formal design of real-time systems using (a) programming languages with unambiguous semantics of time-related behavior and (b) scheduling algorithms.
Number of offerings (in last five years): 1

Course: EECS 5442 3.0
Title: Real-Time Systems Practice
Short course description: The course will focus on the technologies related to the design and implementation of real-time systems. Topics may include: typical real-time applications, process models of real-time systems, scheduling technologies in real-time systems, design and implementation of real-time systems software, real-time systems hardware, real-time operating systems, real-time programming languages, and inspection and verification methods for real-time systems
Number of offerings (in last five years): 2

Course: EECS 5443 3.0
Title: Mobile User Interfaces
Short course description: This course teaches the design and implementation of user interfaces for touchscreen phones and tablet computers. Students develop user interfaces that include touch, multi-touch, vibration, device motion, position, and orientation, environment sensing, and video and audio capture. Lab exercises emphasize these topics in a practical manner.

Number of offerings (in last five years): 4

Course: EECS 6390B 3.0
Title: Scheduling in Hard Real-Time Systems
Short course description: This course discusses concepts and methods for satisfying timing constraints in large, complex hard-real-time systems. Topics include: characteristics of hard-real-time systems, timing constraints, periodic and asynchronous processes, run-time and pre-run-time scheduling, cyclic executives, priority scheduling, preemptive and non-preemptive scheduling, synchronization, schedulability analysis, resource management, and real-time programming language constructs.
Number of offerings (in last five years): 1

Course: EECS 6412 3.0
Title: Data Mining
Short course description: This course introduces fundamental concepts of data mining. It presents various data mining technologies, algorithms and applications. Topics include association rule mining, classification models, sequential pattern mining and clustering.
Number of offerings (in last five years): 5

Course: EECS 6431 3.0
Title: Software Re-Engineering
Short course description: Industrial software systems are usually large and complex, while knowledge of their structure is either lost or inadequately documented. This course presents techniques that aid the comprehension and design recovery of large software systems.
Number of offerings (in last five years): 1

Course: EECS 6432 3.0
Title: Adaptive Software Systems
Short course description: Adaptive software systems are software systems that change their behaviour and structure to cope with changes in environment conditions or in user requirements. Adaptation includes self-optimization, self-protection, self-configuration and self-healing. This course covers basic and advanced concepts in engineering adaptive systems and has a special focus on self-optimization. It introduces the students to the mathematical foundations of adaptive systems including performance models, estimators for performance models, feedback loop architectures and strategies, and optimization.
Number of offerings (in last five years): 2

Course: EECS 6444 3.0
Title: Mining Software Engineering Data to Support the Development, Testing and Maintenance of Large Scale Software Systems
Short course description: Software engineering data (such as source code repositories, execution logs, performance counters, developer mailing lists and bug databases) contains a wealth of information about a project's status and history. Applying data mining techniques on such data, researchers can gain empirically based understanding of software development practices, and practitioners can better manage, maintain and evolve complex software projects.
Number of offerings (in last five years): 3

Course: EECS 6490A
Title: Concurrent Object Oriented Languages

Short course description: In this course, we focus on concurrent programming in the object oriented language Java. The course consists of three main parts. In the first part, we discuss concurrent programming in general. In the second part, we concentrate on writing concurrent programs in Java. In the third and final part, we look at techniques and tools to verify concurrent Java programs.
Number of offerings (in last five years): 2

6. Comment on the expertise of the faculty who will actively support/participate the field and provide a Table of Faculty by field, as follows:

All faculty members mentioned in the table below conduct research in the field and supervise graduate students in the field.

Faculty Member & Rank	Home Unit	Primary Field	Category
Aijun An, Professor	EECS	Computer Science	Full member
Jack Jiang, Assistant professor	EECS	Software Engineering	Full member
Henry Kim, Associate professor	Schulich	Software Engineering	Associate Member
Sotirios Liaskos, Associate professor	ITEC	Software Engineering	Full member
Marin Litoiu, Associate professor	ITEC	Software Engineering	Full member
Jonathan Ostroff, Professor	EECS	Software Engineering	Full member
Vassilios Tzerpos, Associate professor	EECS	Software Engineering	Full member
Franck van Breugel, Professor	EECS	Software Engineering	Full member
Jia Xu, Associate Professor	EECS	Software Engineering	Associate Member

Note: Up-to-date CVs of faculty who will actively participate in delivering the graduate program must be included as an appendix.

7. Comment on the projected in-take into the field, including the anticipated implementation date (i.e. year and term of initial in-take), and indicate if the projected in-take is within or in addition to the existing enrolment targets for the parent program.

The in-take into the field was 7 students this year. Given that the faculty complement in Software Engineering is planned to increase, this number is expected to increase somewhat in the near future This projected in-take is within the existing enrolment targets for the parent program.

The field highlights Software Engineering within the broad spectrum covered by the parent program (from Electrical Engineering to Computer Science) and, therefore, will be helpful for recruitment of new students. Within the highly regulated discipline of engineering, recognition of the type of engineering is very valuable for graduates when seeking employment.

8. Comment on the impact of the field on the parent program, focusing on the extent of diversion of faculty from existing graduate courses and/or supervision, as well as the capacity of the program to absorb any anticipated additional enrolment.

The introduction of this field will *not* impact the parent program. The course requirements are not changed. There is sufficient capacity for supervision.

9. Support statements

- **from the relevant Dean(s)/Principal, with respect to the adequacy of existing resources necessary to support the new field, as well as the commitment to any plans for new/additional resources necessary to implement and/or sustain the new field**
- **from the relevant Faculties/units/programs confirming consultation on/support for the new program, as appropriate**
- **from professional associations, government agencies or policy bodies with respect to the need/demand for the proposed program, as appropriate**