

SOFTWARE

Software is ubiquitous in today's world of information and interdisciplinarity: in our smartphones and computers, in our cars and public transport, in our communication centres, in industries and in all facets of entertainment. This thematic cluster offers students the opportunity to advance their knowledge in all aspects of software design, including the requirements study, choice of architecture, detailed design, construction, commissioning, operation and maintenance. Students enrolled in this thematic cluster will benefit from the recognized expertise of Polytechnique Montréal's team of software engineering professors, some of whom have recently been identified as the most productive researchers in their field (Karanatsiou et al., "A bibliometric assessment of software engineering scholars and institutions (2010–2017)," *Journal of Systems and Software*, 2018). Candidates for this thematic cluster must be in their final year of an engineering degree program and must have completed courses in software design, software processes, software testing and validation, and computer networks.

Students must take 12 to 15 credits among the following :

LOG6406E // Human Centered Inquiry for Software and Computer Engineering (3 cr.)

Human-centered research methods for understanding and supporting computer and software developers, designers, users, and other stakeholders. Interview for understanding users/developers and eliciting requirements. Survey for collecting large-scale user and developer-centered data. Lab-based human-centered study for evaluating computer and software systems. Structured observation of usage and development activities. Automated human-centered data collection and open data. Qualitative and quantitative analysis of human-centered development and usage data. Reporting and presentation of results. Ethical concerns when working with human subjects. Exemplars of these methodologies and methods in real-world computer and software engineering research scenarios.

LOG8371E // Software Quality Engineering (3 cr.)

Basic concepts and definitions relevant to software quality, quality assurance, tests, quality engineering and quality planning. Anomaly prevention and fault classification. Fault tolerance. Software reliability engineering. Quality models. Comparison of different quality assurance techniques. Improvement of the software development process. Measuring the software and its processes. Identification of risks for the quantifiable improvement of quality.

LOG8430E // Software Architecture and Advanced Design (3 cr.)

Advanced software design methods and choosing architecture. Software architectures: multitier, client-server, extendible and dynamic. Advanced concepts of software library installation and dynamic loading of components. Advanced design patterns for distributed systems: service access and configuration, event processing, synchronization and simultaneous access. Emerging approaches in design and architecture: aspect-oriented design, service-oriented architecture and others.

LOG8415E // Advanced Concepts of Cloud Computing (3 cr.)

Key concepts of cloud computing. Key mechanisms and key architectures of cloud computing platforms. Service delivery models of a cloud computing architecture. Virtualization. Big data analytics with MapReduce and NoSQL. Migration of applications to the cloud. Cloud computing patterns. Interoperability issues in the cloud. Quality of Service metrics and Service Level Agreement (SLA). Dependability and security in the cloud. Legal issues related to cloud computing. Mobile cloud computing.

INF8900E // Directed Readings in Computer and Software Engineering (3 cr.)

Selection of a research problem to be explored under the supervision of a professor. Literature review. Critical analysis of selected publications. Preparation and submission of a report.

French Language Course (3 cr.)

Exchange students have access to the Université de Montréal credited French language course offer and will receive by email several weeks before the beginning of classes the detailed application procedure. For more information regarding the French as a second language course offer [click here](#).

[Back to Cover Page](#)

INTERNATIONAL
THEMATIC
CLUSTERS IN
ENGINEERING

**POLYTECHNIQUE
MONTRÉAL**

TECHNOLOGICAL
UNIVERSITY



For more information: point@polymtl.ca // 514 340-4975
POLYMTL.CA/INTERNATIONAL

CODE

LOG8371E

COURSE SYLLABUS

NAME	HOURS/WEEK (in class / practical work / individual work)	CREDITS
Software Quality Engineering	3 / 1.5 / 4.5	3
PREREQUISITES	COREQUISITES	SEMESTER
70 credits for undergraduate students	None	<input type="checkbox"/> W <input type="checkbox"/> S <input checked="" type="checkbox"/> Fall

COURS DESCRIPTION

Basic concepts and definitions relevant to software quality, quality assurance, tests, quality engineering and quality planning. Anomaly prevention and fault classification. Fault tolerance. Software reliability engineering. Quality models. Comparison of various quality assurance techniques. Improvement of the software development process. Measuring software and its processes. Identification of risks for the quantifiable improvement of quality.

COURSE AND LAB OBJECTIVES

By the end of the course, the student will be able to:

- carry out a quality assurance plan;
- determine software validation and verification activities;
- know the principal techniques for software measurements;
- analyze software and process measurements to make a quality diagnosis;
- realize the importance and necessity of quality control;
- adopt a critical position in terms of software quality.

COURSE STRUCTURE (CONTENT AND HOURS) - 1ST SECTION

Subjects	Hours
What do we mean by quality software?	
- Various perspectives, ISO9126 standard	3
- Quality assurance	
- Prevention and reduction of defects	
Quality engineering:	
- Activities and processes	4
- Planning: goals and strategies of quality strategies	
- Quality assessment and improvement	
Software tests:	
- Concepts, issues, and techniques	6
- Activities, management, and automation	
- Adaptation, specialization and integration	
Aspects of quality assurance beyond testing:	
- Software inspection	6
- Formal verification	
- Comparisons of quality assurance techniques	
Quantifiable aspects of quality:	
- Direct and indirect measurements	6
- Analysis, monitoring and reaction	
- Implementation, integration and support tools for a measurement program	
Quality and measurement models:	
- Generic models	6
- Choice of measures and models	
- Assessment frameworks	
Risk identification:	
- Classification of defects, ODC model	6
- Identification techniques: principal components, neural networks	
- Software Reliability Model, IDRM, SRGM, TBRM	
Periodic tests	2
Total :	39 h

STRUCTURE OF PRACTICAL WORK (CONTENT AND HOURS) - 2ND SECTION

Subjects	Hours
One class to become familiar with data resulting from a program to measure the quality of an industrial project.	3
Two classes for the design of a quality plan and to gather measurements about the artifacts.	6
Two classes for the analysis of results from the measurement program and to establish a diagnostic of the quality of the software product	
One class for the analysis of software product security vulnerabilities.	3
Total: 18 h	

ASSESSMENT METHODS

Nature	Individual	In groups	Number	Weight
Periodic testing	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	30 %
Practical work	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2	30 %
Final exam	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	40 %
Individual: 100 %		In groups: 0 %		Total: 100 %