

COURSE OUTLINE

1. INTRODUCTION

The purpose of the present offering of SOEN 6471 is to introduce **state-of-the-art** in **theoretical as well as practical aspects** of **software architecture** to the understanding and development of software-intensive systems. In doing so, the attention will be primarily on **software architecture modeling and software architecture quality**.

The course will combine elements of software architecture **education and training**. To do that, the course will create circumstances that can help traverse the different levels of the **Bloom's Taxonomy**, and in the process of doing so develop certain **desirable habits**, en route to skills. The students will be encouraged and provided opportunities to move up the levels of the **Dreyfus Model of Skill Acquisition**. Figure 1 shows a relationship between competency, habit, skill, and related terms. It is expected that by the end of the course the students can **think, say, and do** like a **software architect**.

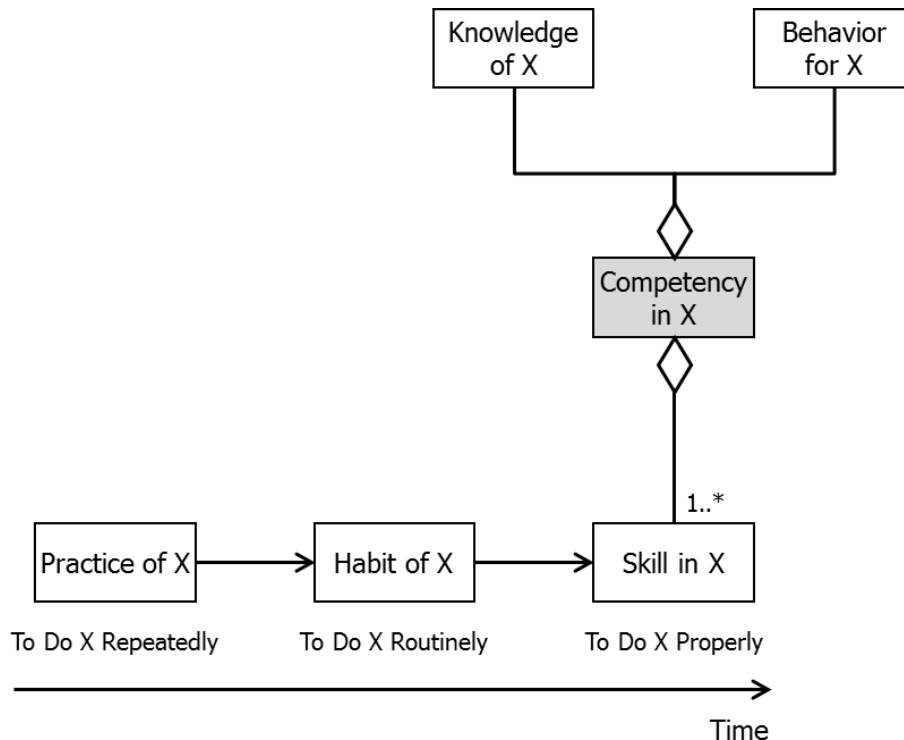


Figure 1.

It is expected that the students will **expand their current universe in multiple directions**, and, by the end of the course, have reached a desirable level of **software engineering maturity \geq CMMI Level 3**.

In particular, it is expected that by the end of the course the students can become proficient in a variety of **software architecture elements of knowledge**, know when to identify and apply them, and know how to derive conclusions from their applications. This, in turn, would require that the students have an **open mind (even towards the things that they do know) and self-assess** regularly. This course could be considered by a student as a path towards **polyliteracy**. **A Guide for Completing a Course Successfully** provides guidelines for such a realization.

2. PREREQUISITES

It is expected that a student has (1) background in basic discrete mathematics, including logic, sets, relations, and graphs; (2) background in software engineering, including exposure to software processes, software requirements (including use cases), and the Unified Modeling Language (UML); (3) experience in technical writing; (4) skill in diagramming; (5) skill in a current object-oriented programming language; (6) experience in making technical presentations; (7) experience in working in a team environment; and (8) the ability of independent decision making.

3. ADMINISTRATION

The course is subject to the **Course Protocol** available on the course Web Site.

4. COMMUNICATION

There are three means of asynchronous/synchronous communication.

4.1. WEB

The URL of the Moodle Page for the course is:

`https://moodle.concordia.ca/moodle/course/view.php?id=175605`.

It is suggested that students check the Home Page regularly for the latest announcements and information.

4.2. ELECTRONIC MAIL

The optimal means for corresponding with the instructor outside the classroom is via e-mail at the address `kamthan@cse.concordia.ca`.

It is expected that a student follows the **Communication Protocol** in an e-mail correspondence.

In a **project-related correspondence**, the rest of the team members should be copied on the message to ensure transparency across all team members.

To ensure this becomes a practice, there will be cost associated with **not** doing so. For each violation, there will be a **deduction of 5 minutes from the total time allowed** for the sender of the e-mail message for the next upcoming test. The student in question will **not** be informed in such cases, but will find out about the reduced time on the day of the respective test.

4.3. IN-PERSON

The instructor can be **available by appointment** if the need for it is properly rationalized by the student making the request.

5. COURSE MATERIAL

5.1. COVERAGE

The object of a course is not to cover the material but to uncover part of it.

— Joseph P. LaSalle

It doesn't matter what we cover; it matters what you discover [and invent].

— Victor Weisskopf (and Yours Truly)

The area of software architecture is established, yet evolving constantly (that is, *statu nascendi*). There are several **critical, equally significant, not necessarily comparable, paths** through any field, and software architecture is no different. This course, as shown in Figure 2, should be viewed as '**a**' rather than '**the**' course on software architecture. In general, the coverage will be **interplay between depth and breadth**.

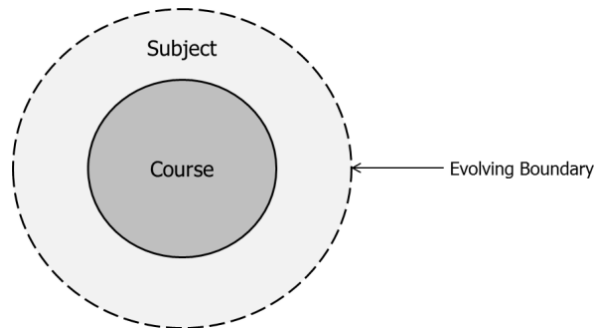


Figure 2.

The following is a broad list of topics to be covered, **directly or indirectly**, in the course:

Context of Software Architecture, Software Process Models and Software Architecture, Stakeholders of Software Architecture, Software Product Quality, Quality of Software Architecture, Design of Software Architecture, Views of Software Architecture, Experiential Knowledge in Software Architecture (Principles, Tactics, Styles, Patterns, Anti-Patterns, Standards, Reference Architectures), Description of Software Architecture, Languages for Describing Software Architecture, Conceptual Modeling, UML and UML Extensions for Software Architecture Description, Software Architecture and Technical Debt, Evaluation of Software Architecture, Software Architecture Reviews, Software Product Line Architectures, Case Studies on Software Architectures, Enterprise Architecture.

This list of topics may evolve according to the needs of the class. The class will be informed in case of changes.

5.2. RESOURCES

The value of an education in a liberal arts college is [...] the training of the mind to think something that cannot be learned from textbooks.

— Albert Einstein

For lectures, no single resource (such as a book) will be adopted and followed as-is. The lecture notes and related material (such as readings or videos) pertaining to the course will be made available on the course Web Site. These are prone to change, and changes will be duly announced. The students are **urged** to continually **think, read, try, and reflect** on each of these regularly. The students are **encouraged to share their thoughts and provide comments** on the lecture notes during the term, possibly leading to a **feedback loop**.

The lecture notes aim to be **original** in some manner, are **prescriptive or descriptive** as necessary, **parallel book chapters** in quality, and are ‘**living documents**’ prone to perpetual evolution. They include support for **mobile microlearning**. The purpose of the readings is to (1) expose the topic, (2) identify the author(s), and (3) underscore the publishing source, and, in doing so, highlight sources that are **authoritative** in some sense. The students may seek clarifications on any readings assigned. However, it is expected that the students are accustomed to **independent study** and possess the ability of **critical thinking** necessary to derive conclusions from any readings assigned. **It is important to not only know what is important, but also what is not.**

If the comments submitted in writing by a student are deemed as “**significant contribution**”, then that student will be waived from attempting a predetermined set of one or more problems on one or more tests, and assigned full marks automatically on those problems. The student will be informed in such cases.

The **Copyright Act** applies to lecture notes. It is **illegal** to post them, to sell them, to share them, or include them in a personal portfolio.

6. TEACHING AND LEARNING STYLE

The doer alone learneth.

— Friedrich Nietzsche

The students can expect a suitable combination of **conservative as well as progressive** teaching style that looks backward in **space and time** more than it looks forward.

The students can expect an **interactive** lecturing style, prompting them to answer questions, engage in discussions, or search for something on the topic under study.

The students should aim for **continuous learning, by knowing and by doing, iteratively as well as incrementally**. This requires going outside one's **comfort zone**.

To do that, the students are encouraged to at least scan the material to be covered in the lecture before coming to the classroom, **use their mobile devices** inside the classroom in a manner that does not disturb others, and **interact regularly** by means of comments and/or questions, both inside as well as outside the classroom. The **Classroom Protocol** applies.

6.1. PROJECT-RELATED INTERLUDES DURING LECTURES

In every lecture, there will be a short period dedicated specifically to inquiries on the state of the project-related work. This will also be an opportunity for the students to bring to attention any issues that are **impeding** the progress of project-related work.

7. ASSESSMENT

The course assessment is based on **two** complementary but interdependent elements, namely **project and tests**.

The course advocates a **separation of educational and personal obligations**. In general, the course will **not accommodate exceptions** based on **individual grounds and/or personal reasons**. In particular, the schedules (dates/times) for submission of project deliverables, and the schedules (dates/times) of the tests, are **non-negotiable**. For the sake of fairness, exceptions, if any, will only be made if they apply to the entire class, not to individuals.

7.1. PROJECT

There will be a team project aimed at **examining the software architecture of an open source software system**. The work related to the project will be subject to the **Communal Work Protocol**.

The project is to be carried out in a **team environment**. To do that, the class will be organized into teams of **equal or almost equal** size. A larger team will be expected to **demonstrably** do more work compared to a team of a smaller size.

The teams and their members will be **predetermined** (fait accompli) at some point in time near the beginning of the term and will normally not change for the rest of the term. (The teams will be formed pseudo-randomly. The members of those teams will also be assigned pseudo-randomly.)

For each team, the **roles and responsibilities** for each team member, for each deliverable, are to be **decided collectively** by all members of that team, by themselves. It is important that there be **minimum overlap** among the responsibilities. The roles and responsibilities for each team member should be made **public**, such as on **social software**, and should remain so for the entire duration of the project. A **non-availability** of a resource with the roles and responsibilities on the due date of a deliverable could lead to a deduction of marks for the entire team for that deliverable.

The project will require **(re)search** by the students.

The project will consist of **three** deliverables. The details on the deliverables, including the work involved and submission schedule, will be made available on the course Web Site.

It is important that the **teams meet regularly** (at least on a weekly basis), as well as **interact with the professor regularly** should they have any questions. It is also in the interest of each team member to work on the project **regularly, individually as well as communally**, throughout the course.

The marking for the deliverables will be **entirely individual**. A deliverable will be assigned a baseline mark, but the marks for individuals may vary. The individual marks for a given deliverable will depend on, apart from the conditions outlined in the description of that deliverable, a **match between the responsibilities and the contribution**, by an individual, to that deliverable. The individual marks will be measured, in part, by the **Course Project: Assessment Form**. A non-submission of the Course Project: Assessment Form by a team member could lead to a deduction of marks for that team member.

In general, missed deadlines related to the project cannot be made up.

7.2. TESTS

There will be **three** tests during the term. The tests will be subject to the **Test Protocol**.

The tests will be based on the **lecture notes, mandatory readings, exercises assigned, and/or public discussions** during the classes. The **prerequisites** for the course and the **project** will be relevant towards the tests. The style of problems on the tests will, in principle, be suggested by **A Manifesto on Testing**.

The tests may include **‘open’ problems** that the students are asked to solve before a test and include their solutions when submitting that test. The open problems will typically be more challenging and require more work than other problems on a test, but less challenging and require less work than problems on an assignment. The open problems will carry **more marks** than other problems on a test. The open problems will be announced **at least a week in advance** of a test they are supposed to appear on.

The syllabuses and schedules of the respective tests will be announced on the course Web Site.

The tests will be conducted **entirely** on **Moodle** as per **A Guide for Moodle-Based Examinations**. Any additional constraints will be posted on the course Web Site.

The students will be responsible for bringing their own equipment, as necessary, for writing tests, and for the **state** of that equipment for the entire duration of the tests.

The students will be expected to come to the classroom and write the tests in-person. An attempt to write a test outside the classroom will be noted as an absence during that test.

In general, missed tests cannot be made up. An **absence** during a test will lead to 0 marks for that test.

8. GRADING

The letter grade for the course will be based on the following distribution of weights:

Project	$D_1(08\%) + D_2(10\%) + D_3(12\%)$	= 30% Marks
Tests	$T_1(16\%) + T_2(18\%) + T_3(36\%)$	= 70% Marks

In order for a student to pass the course, a passing mark must be obtained **cumulatively** on project and tests.

It should be noted that marking is **absolute in some cases and relative in other cases**. This has certain implications, including that the highest overall marks in the class do not necessarily mean the highest possible grade, and the lowest overall marks in the class do not necessarily mean the lowest possible grade.

The marks will be made public so that a student can track his or her progress, absolutely and relatively (with respect to the rest of the students in the class, especially with respect to the **class average**).

There is no fixed, a priori, relationship between the marks expressed as a numerical percentage and the final letter grade for the course, except that a higher percentage corresponds to a better grade. In conformity with the policy of the Department of Computer Science and Software Engineering, the determination of final letter grades from total percentages is not established according to fixed levels, but is scaled to ensure fairness and accuracy.

The grades are assigned based on **performance** only and not on any other criterion.

It is expected that, related to any issues pertaining to marking or grading, each student has **read and understood** the documents entitled **Fairness Protocol** and **Reassessment Protocol** made available on the course Web Site.

9. GENERAL

It is important that each student understands the **constants** pertaining to the course, the **variables** pertaining to the course, and the difference between the constants and the variables. In particular, a constant should not be converted into a variable.

9.1. ACADEMIC CONDUCT

It is assumed that a student registered in this course is aware of and understands the rules outlined in the **Concordia University's Academic Code of Conduct**, and agrees to abide by them. In particular, the **burden of proof of originality** of work rests on the student.

Any incident suspected of collusion or plagiarism will be identified so **explicitly** in corresponding submitted work (project deliverable or test, as the case may be), and will subsequently be **reported automatically** to the University. The incident will then be duly examined by relevant authorities at the University, and the student in question will be informed of the outcome.

9.2. TIMELINES

The timelines for any form of assessment (including the tests and project deliverables) are to be followed as-is. In general, any non-compliance will result in a non-assignment of marks. In particular, any delay in the submission of project deliverables, or a non-attendance at any test, for any reason whatsoever, apart from the reasons listed below, will result in 0 marks.

9.3. ABSENTEEISM/NON-SUBMISSION

There must be a compelling reason for a student not being able to comply towards any mode of assessment (such as, not appearing for a test due to a certain event). The student must submit the reason in **writing** supported by appropriate documents (such as, in case of a medical condition, a doctor's note **only** from the Concordia University Health Services) for any consideration (to be determined by the instructor, whose decision will be binding and final in this matter) within **one week** of the respective event.

10. DISCLAIMER

This course outline should be viewed as an agreement between the instructor and a student with respect to the course, by which both are bound. It is assumed that each student who is registered in the course has **read and understood** the contents of this course outline. **Any** clarifications, if necessary, of the contents of this course outline **must** be sought **no later than the day of the second lecture**.

Note from the University Administration: In the event of extraordinary circumstances beyond the University's control, the content and/or evaluation scheme in this course is subject to change.