University Undergraduate Degree Level Expectations (UUDLES)
Software Engineering

Depth and Breadth of Knowledge, Knowledge of Methodologies, Application of knowledge, Communication Skills

The IEEE/ACM Software Engineering Curriculum Guidelines[1] note that although there are strong similarities between software engineering and traditional engineering, there are also fundamental differences including the following: the foundations of software engineering are primarily in computing science and not in the natural sciences; the focus is on discrete rather than continuous mathematics, and; the concentration is on abstract/logical entities instead of concrete/physical artefacts.

The proposed software engineering program covers the major categories of software knowledge listed in the IEEE/ACM guidelines and also fulfils CEAB engineering requirements as well as requiring methods and skills needed for engineering mission-critical systems. Thus the proposal includes natural sciences, stresses both discrete and continuous mathematics, and treats concrete/physical artefacts (in addition to abstract/logical entities as in computer science).

The primary learning outcomes are:

1. A solid foundation in mathematics, science, and engineering design principles (including the categories of *Mathematical & Engineering Fundamentals* and *Professional Practice* in the IEEE/ACM curriculum).
2. A solid foundation in computer science (including the category of *Computing Essentials* in the IEEE/ACM curriculum).
3. The ability to develop mathematical models and specifications that ensure that a software product satisfies its requirements, especially for mission-critical systems (including the category of *Software Modeling & Analysis* in the IEEE/ACM curriculum).
4. The ability to design, develop, test, document, and maintain high-quality software intensive systems (including the categories of *Software Design* and *Software Validation and Verification* in the IEEE/ACM curriculum).
5. Skills related to Software Evolution, Process, Quality and Management.[2]
6. The ability to communicate effectively and to work in a team.

[1] Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Computing Curricula, Joint Task Force IEEE/ACM, August 23, 2004.

[2] These are the remaining categories in Table 1 (p21) of the IEEE/ACM curriculum.

Specific courses that realize the above categories of learning expectations

1. A solid foundation in science, mathematics and engineering design principles. The first and second year of the proposed program is similar to the first and second year of the Computer Engineering curriculum in this respect.

2. A solid foundation in computer science including:

   a. An intellectual understanding of the central role of algorithms and data structures (CSE 1020/ 1030/ 2011/ 3101).

   b. An understanding of computer hardware and its interplay with software (CSE 2021/ 3201/ 3215/ 3221).

   c. Fundamental programming skills, including a thorough grounding in the methodology of testing, to permit the implementation of algorithms and data structures in software (CSE 1020/ 1030/ 2011/ 2031/ 3311).

   d. The option to either specialize in a particular software engineering application field, such as networks (CSE 3214/ 4214/ 4215), databases (CSE 3421/ 4411/ 4412), and user interfaces (CSE 3461/ 4441), or to obtain a broad knowledge of all three (the elective course structure allows for either approach).

3. The ability to develop mathematical models and specifications that ensure that a software product satisfies its requirements, especially for mission-critical systems. The proposed new courses *System Specification* (3rd year) and *Mission Critical Systems* (4th year) focus on these ideas. CSE 3311 (Software Design) uses design by contract to specify systems. The ability to deal with embedded and control systems is stressed in CSE 3215 and ENG 4550. CSE 3201 (Logic Design) and CSE 3215 (Embedded Systems) provide students with the ability to design complex computing systems that consist of both hardware and software components. CSE 3451 (Signals & Systems) and ENG 4550 (Introduction to Control Systems) provide students with continuous feedback control knowledge to design stable hybrid safety critical systems.

4. The ability to design, develop, test, document, and maintain high-quality software systems. This includes:

   - *Requirements.* The student will identify and analyze criteria and specifications (software engineering requirements) appropriate to specific problems, and plan strategies for their fulfilment (MATH 1090; CSE 3311/ 4312)

   - *Modelling and design* of computer-based systems. The student will demonstrate comprehension of the trade-off involved in design choices (CSE 3311) and apply theoretical knowledge (MATH 1090; CSE 1019/ 3101) and implementation skills (CSE 1020/ 1030/ 2011/ 2031/ 3311/ 4413)

   - *Critical evaluation and testing.* (CSE 2031/ 3311/ 4313)

These principles and methodologies lead to software implementations that are robust, reliable, thoroughly tested, within specification and budget, maintainable and appropriate for their intended "client". Both project courses (CSE 2311 and ENG 4000) reinforce these principles and allow the students to practice them in the context of a large software system.

Active participation in a significant team software design project is an indicator that graduates can successfully integrate the diverse theoretical principles and applied tools they have attained toward the solution of a particular large-scale real-world software design problem, as well as work effectively within a team.

5. Software Evolution, Process, Quality and Management. These topics are treated in depth in the new 4th year Advanced Software Engineering course (CSE 4314 - a short description is provided in Appendix C).

6. Communication Skills: Students in the proposed program will have an opportunity to develop their ability to communicate information, arguments, and analyses accurately and reliably, orally and in writing, to a range of audiences.

   Modern engineering programs emphasize oral and written communication; these are considered vital professional skills and are therefore included in many technical courses. This applies especially to the ENG 1000 (Introduction to Engineering Design), ENG 2001 (Engineering Projects: Management, Economics and Safety), ENG 3000 (Professional Engineering Practice), CSE 2311 (Software Development Project), and ENG 4000 (Engineering Project) courses because of their emphasis on project and group work and the associated documentation and presentation. Integral to many third- and fourth-year courses are individual or team projects that include reports, web pages, and presentations.

   Explicit instruction in writing is provided during the first year one-hour-per-week writing workshop course ENG 1001 (Introduction to Technical Writing) associated with ENG 1000 (Introduction to Engineering Design).

Awareness of Limitations of Knowledge

Graduates of the proposed program will possess a substantial understanding of the limitations of what computer technology can and cannot do, and the ability to distinguish between what computing inherently cannot do from what computing might be able to accomplish in the future with advances of science and technology. Limitations of knowledge are manifested as limitations of computing specifications and processes.

Beyond the theoretical limitations on computing (and formal specification systems such as predicate logic) that the student will learn in CSE 2001, software engineering graduates will be aware of the limitations of static (CSE 3342 and CSE 4315) and dynamic (CSE 4313) software verification approaches, as well as the tradeoffs between them.

Autonomy and Professional Capacity

Students who successfully complete the program will have demonstrated:

- Qualities and transferable skills necessary for further study, employment, and other activities requiring:
    - the exercise of initiative, personal responsibility and accountability in both personal and group contexts;
    - working effectively with others;
    - decision-making in complex contexts;
- The ability to manage their own learning in changing circumstances, and to select an appropriate program of further study;
- The capacity to use knowledge to solve problems, to generate ideas and to test hypotheses; and
- Behaviour consistent with professional and academic integrity and social responsibility.

The proposed program contains several courses that will allow students to demonstrate these qualities, such as ENG 1000, ENG 2001, CSE 2311, ENG 3000, and ENG 4000.