CS-290 **Responsible software**

Hardebolle Cécile

Cursus	Sem.	Type
Communication systems	BA3, BA5	Opt.
Computer science	BA3, BA5	Opt.
HES - IC	H	Opt.

Language of teaching	English
Credits	4
Session	Winter
Semester	Fall
Exam	During the semester
Workload	120h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

Software's growing importance increases engineers' responsibility to integrate ethical concerns in the design and development process. This course teaches students concrete strategies for responsible software engineering, focusing on identifying ethical issues and mitigating risks to minimize harms.

Content

The course combines a) knowledge about a range of ethical challenges related to software, and b) pragmatic tools/strategies that students can use in practice to identify and work on ethical risks in the domain of software.
The course will use cases inspired or based on real-world examples to review a range of ethical challenges related to software, including the following themes and questions:

- Safety: How to anticipate potential harmful impacts resulting from the normal and abnormal use of software at the scale of the individuals, groups or society? What types of mechanisms can be used to protect users from harmful impacts and what are their limits?
- Fairness: What is bias in software? What are its sources and what types of impacts can it have? How to identify and mitigate fairness issues?
- Sustainability: What influences the environmental impact of software in terms of energy consumption, CO2 emission and use of resources? How to estimate and limit this impact?
- Transparency and Autonomy: What issues arise from a lack of transparency in software, and how can methods help users understand software functions and limits? How does software design affect user control and which factors contribute to user empowerment or disempowerment?

Through the course activities, students will get to learn and practice concrete strategies for a responsible approach to software design and development including:

- User and stakeholder analysis strategies
- Strategies for analyzing values
- Impact anticipation strategies
- Risk assessment strategies
- Decision-making strategies

Note: Other concerns, such as security and privacy, are not directly addressed in this course since they are already covered in other courses in the curriculum. However they may be involved in the cases used in the course.

Learning Prerequisites

Required courses

Introduction à la programmation (CS-107)

Important concepts to start the course

Basics of imperative programming

Learning Outcomes

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

- Assess / Evaluate the level of responsibility of existing software using a set of ethical lenses
- Identify ethical questions during software design and development
- Integrate environmental and social concerns into the software design and development process
- Take into consideration the perspectives of different stakeholders in a software project
- Examine stakeholder values and identify value tensions in a software project
- Investigate the potential benefits and potential harms of software for different stakeholders
- Assess / Evaluate the ethical risks associated with software
- Make design or development decisions taking ethical risks into account

Transversal skills

- Take account of the social and human dimensions of the engineering profession.
- Take responsibility for environmental impacts of her/ his actions and decisions.
- Demonstrate the capacity for critical thinking
- Demonstrate a capacity for creativity.
- Use both general and domain specific IT resources and tools
- Access and evaluate appropriate sources of information.

Teaching methods

The course is designed with a flipped classroom format, based on online resources from the "Responsible Software" MOOC, developed by the Center for Digital Education.
Each week begins with a problem representative of the topic before introducing the corresponding theoretical content, in the following typical sequence called "problem-solving before instruction":

1. In-class problem-solving on computer (analysis, design or development), with assistants;
2. Independent study of theoretical content (concepts, principles, strategies...) in the form of videos and/or reading material, guided by short activities such as quizzes or open-ended questions;
3. In-class application to one or more concrete case(s) in class with group activities.

Expected student activities

- Complete the exercises
- Study resources (videos and/or readings) and complete preparation activities (quizzes) prior to classroom sessions
- Apply learning to in-class case studies

Assessment methods

- Graded assignments (40%): 2 assignments (20% each) during the semester
- Final exam (60%): 1 individual written test during the semester

Supervision

Office hours	No
Assistants	Yes
Forum	Yes

Resources

Virtual desktop infrastructure (VDI)

No

Moodle Link

- https://go.epfl.ch/CS-290