International Journal of Engineering Education Vol. 39, No. 3, pp. 542–556, 2023
Printed in Great Britain

0949-149X/91 $3.00+0.00

Sustainability and Ethicality are Peripheral to Students' Software Design*

SIARA ISAAC[1]**, ADITI KOTHIYAL[2], PIERLUCA BORSÒ[3] and BRYAN FORD[3]
[1] Center for Learning Sciences, École polytechnique fédérale de Lausanne (EPFL), Lausanne, Switzerland.
[2] CHILI Lab, École polytechnique fédérale de Lausanne (EPFL) and Indian Institute of Technology, Gandhinagar, India.
[3] DEDIS Lab, École polytechnique fédérale de Lausanne (EPFL), Lausanne, Switzerland.
E-mail: siara.isaac@epfl.ch, aditi.kothiyal@iitgn.ac.in, pierluca.borso@epfl.ch, bryan.ford@epfl.ch

The conceptual design phase is a fascinating moment to observe how a design task is interpreted, as the (often implicit) relative importance students accord to the various requirements and constraints offers a window into the thinking underpinning their designs. Our qualitative study used the think-aloud protocol with 11 third year computer science students working on a software design task to investigate the criteria that students used to guide and evaluate their developing conceptual designs. While the trio of feasibility, economic viability, and consumer desirability are often used in design decisions, our analysis also looked for how aspects of ethics (i.e. ethicality) and sustainability informed students' thinking. We found that considerations of feasibility and consumer desirability dominated students' thinking, while economic constraints were rarely addressed and even less often the economic impact pertaining directly to the software design. Students' consideration of ethicality in terms of data privacy and accommodations for disability (an explicit criterion in the design task) indicate that many students did not see ethical aspects as sufficiently important to influence their design choices. Sustainability was introduced tangentially in the design task but was absent from students' thinking and design decisions. Our findings suggest that ethicality and sustainability should be explicitly included in the design thinking model taught to students for software design to ensure that they bring these considerations to their professional work and therefore to the next generation of software.

Keywords: software design; sustainability; ethics; ethicality; think aloud; human-centred design

1. Introduction

Eli Blevis' definition of design as "*an act of choosing among or informing choices of future ways of being*" [1] highlights the power of design decisions to shape the world. To make these choices, designers typically adopt some criteria to guide and inform their decisions. Human-centered design [2, 3] is an approach to design that has gained importance as a way to ensure that the user's needs and contexts are an integral part of the design. Within HCD, feasibility, viability and desirability reign as the core criteria guiding designers' choices [2–4]. Desirability has been defined as that which is grounded in the hopes, fears and needs of humans [2], although it is often narrowly interpreted to pertain exclusively to the consumer of the design product (for example, see [3] or [4]). This is underlined by how Martin Maguire [4, p. 589] states that "multidisciplinary design teams" are one of the key principles of HCD yet none of the 8 "stakeholders" given as examples bring a perspective beyond direct interactions with the product. Ezio Manzini [5] raises the critical question as to whether the needs of individual humans or those of society as a whole should take primacy in design decisions. In response, approaches to ethical and value-centered design have grown considerably over the past two decades. These approaches advocate for a systematic and comprehensive incorporation of human values and ethics throughout the design process [6-7]. While these approaches support the consideration of these aspects during the design process, the scope for designers to allow ethics and sustainability to inform their choices depends on the environment in which they are working and the strength of their ethical positions [8–9]. Ivan Szekely [10] found that IT professionals generally respond to direct instructions to meet ethical standards, but because they normally comply with requests from their employers rather than meeting ethical standards due to their own motivations. This is not sufficient.

The world that our graduates will enter needs them to consider how their designs contribute to ways of being that are environmentally sustainable and contribute to robust civil society (i.e. ethicality or desirability for society as a whole, rather than exclusively desirable for individual users). This requires students to engage their thinking about societal and environmental aspects in the context of their disciplinary thinking. Having students use the perspectives and tools of ethical and value-centered design [8] is one way for them to make relevant, situative connections between these fundamental, non-technical skills and their disciplinary design thinking. Joanna Lönngren [11] argues that despite faculty and student interest in ethics, engineering

** Corresponding author.

* Accepted 26 January 2023.

students' introduction to ethics is largely ineffective because engineering culture consistently communicates that ethics are unimportant. Moreover, investigations into changes in ethical reasoning and social responsibility with education have repeatedly found that engineering undergraduates' scores on measures of ethical reasoning and social responsibility either remain static or decline over the course of their education [12-14]. In light of the major challenges to the environment and civil society currently facing humanity, engineering programs must explicitly expand the HCD trifecta to include design aspects beyond the technical and market value of engineered products. Perhaps students are already doing this; to investigate, we designed a qualitative study to make direct, contextualized observations of engineering students' actions during a software design task and our analysis framework explicitly included sustainability and ethicality as well as the traditional criteria of feasibility, desirability and viability, as shown in Fig. 1.

At our institution, third year computer science students typically undertake a large project spanning a full semester. In this paper, we seek to understand how students whose projects focused on software design integrate users' perspectives, ethics and sustainability into their software design thinking. Accordingly, we used a qualitative research methodology to directly observe students as they engaged in the conceptual phase of completing a software design task. Using a think aloud protocol, we were able to directly investigate the design criteria that students employed and those for which students may require additional teaching and learning support to develop proficiency. Specifically, we sought to understand how computer science students incorporate considerations of sustainability and ethicality, in addition to feasibility, viability and desirability in the conceptual phase of software design. This approach does not investigate students' knowledge, opinions or beliefs about these issues but rather the extent to which they *actually* integrate them in their disciplinary thinking. Using a software design task and a think-aloud protocol we investigated the following questions:

1. How are considerations of feasibility, viability, desirability, ethicality and sustainability incorporated into software engineering students' design thinking? What qualitative differences are apparent in how aspects explicitly, or implicitly, specified in the design task are addressed?
2. Does this short protocol allow students to exhibit qualitatively different design approaches relevant to their thinking in human-centred design, ethics or sustainability?

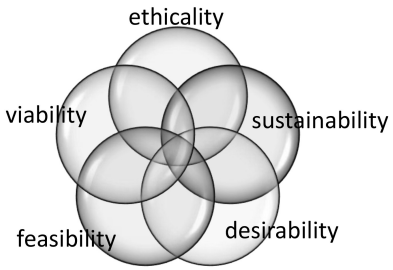

Fig. 1. Our 5 part lens for considering students' software design.

2. Theoretical Framework

The human-centred design criteria trifecta of desirability, feasibility and viability came from IDEO in the early 2000s [2], becoming so popular as to be standardised by ISO 9241–210: 2019. The dominance of this model can be seen in, for example, Nguyen Ngoc et al.'s 2022 review of intelligent systems design which found human-centred design to be the most frequently used model [15]. The three core tenets of the model are looking for ideas that meet the needs, hopes and fears of customers (*desirability*), are technically *feasible* to construct and operate, and are economically *viable*. The criterion of *desirability* is often understood narrowly to focus on meeting the needs or desires of individuals, however there is growing awareness to include ethical considerations, human values and the broader societal implications of the designed product [5, 6, 16]. Therefore, we argue that students' software design education should employ a broader set of explicit criteria, including environmental sustainability and ethicality in terms of the ethical or potential social impact of the software. This extended set of 5 criteria is better aligned with the assessment criteria of accreditation bodies such as the CTi [17] and ABET [18] which explicitly include training in sustainability and ethics, and with strategic goals within our institution. As the human-centred design criteria have been extensively discussed [4], we devote the following paragraphs to exploring the relevance of the (often marginalised) aspects of sustainability and ethicality in software design.

We considered two ways software can contribute to environmental sustainability: greening *of* software and greening *by* software. While the environmental impact of software is not immediately apparent, it directs and influences how computer hardware operates and therefore the resulting energy consumption and carbon emission. As software becomes ubiquitous in our lives, its impact on the environment has similarly grown [19]. For example, one study estimates that maintaining the Bitcoin network requires roughly twice the amount of energy of Switzerland as a whole [20]. Studies

finding that, for example, the final 0.08% increase in AI flower identification consumed almost 400% more energy than the first stage [19], have built awareness of the need to consider the environmental costs of investing huge computational power to achieve incremental performance improvement. The Green Software Foundation [21] has recommendations and guidelines for how software designers can reduce the environmental impact of their products. The potential for greening *by* IT is also significant. Nathan Stegall [22] argues that the role of the designer in the current world is not simply to create "sustainable products" but rather to design products that encourage sustainable behaviour.

We focused on two aspects of ethicality in students' thinking: privacy and accessibility. Alan Thomson and Daniel Schmoldt [23, p.86] caution that "[t]he ubiquity of software systems in all aspects of public and private institutions means that the environment that they create needs to be critically examined as they are developed and deployed". Accordingly, proposed ethical principles for the design of ICT tools include providing user autonomy, being non-discriminatory and minimizing and protecting data [24]. Data privacy is fundamental to issues of ethics in computing and the subject of a "plethora of research and review articles as well as books" [25, p. 122]. Kathrin Bednar et al. [25] identify systems engineers (including software designers) as those with the competencies and skills to create and implement products with appropriate privacy protection. Yet their review of prior work highlights the lack of attention to privacy, dismissed as being too abstract, not necessary at a prototyping phase or not their moral responsibility, as well as excluded from standard computer science education textbooks.

Accessibility is the second aspect of ethicality that we explored. Through their work, software designers create technological tools that lead to new ways of living and working, and create novel social practices. However, they often design for the typical user – the user who represents a majority of the population – thereby leading to a re-enforcing biases and inequities already pervading society and continuing to marginalize certain social groups [26]. As with sustainability, there are two ways in which software can contribute to inclusion – inclusivity *of* software and inclusivity *by* software. As software becomes a means to improve the lives of people, it is imperative that the software is usable and accessible by all groups of users, and that the opportunities created by software are accessible by all groups of users [27]. Extensive research on software design methods that include the needs of a diverse set of users has shown the importance of considering accessibility throughout the software design life cycle and not just as a property of the final product [28]. However, as smartphones become ubiquitous, opportunities for inclusion by software must also be considered [29]. For instance, studies have shown how informal language learning on mobile phones can help with the social inclusion of immigrants [30]. The consideration of inclusivity by software is currently limited within software design, and we argue that it should be central to the practice of software design.

While there is increasing recognition that ethics and sustainability need to be better integrated into software design, both in terms of the user experience and the potential to influence human behaviour, it is not clear to what extent these aspects are incorporated transversally into the thinking and software design of current students. To explore this, we applied our five lenses to computer science students' software designs intersected with a human and value centred software design approach. The five lenses are shown in Fig. 1 and described in Fig. 2.

3. Research Design and Method

Given the nature of our research questions, we choose a qualitative methodology to allow us to directly observe students' software design process in a relatively *in situ* format. Qualitative research methods are underexploited in engineering education research [31], yet offer great potential to explore different research questions and to collect rich, contextualized observations. In our study, we are interested in what students *actually do* during the conceptual part of software design. While a quantitative method such as a questionnaire would allow us to collect many more observations, this approach would access students' tacit knowledge and opinions about what they *should do*, but not provide data about how these aspects are *actually incorporated* into their design thinking. Such rich data would be available using a qualitative approach as described below. Further, our second research question, about the potential to incorporate the five design aspects into the software design task, was also well served by a qualitative approach.

3.1 Think-aloud Protocol to Observe Design Considerations

Think-aloud protocols are a qualitative research method that originated in the field of cognitive processing in psychology [32] and typically involve setting a task for interviewees to complete while narrating their actions and thoughts. This concurrent approach, as opposed to having students

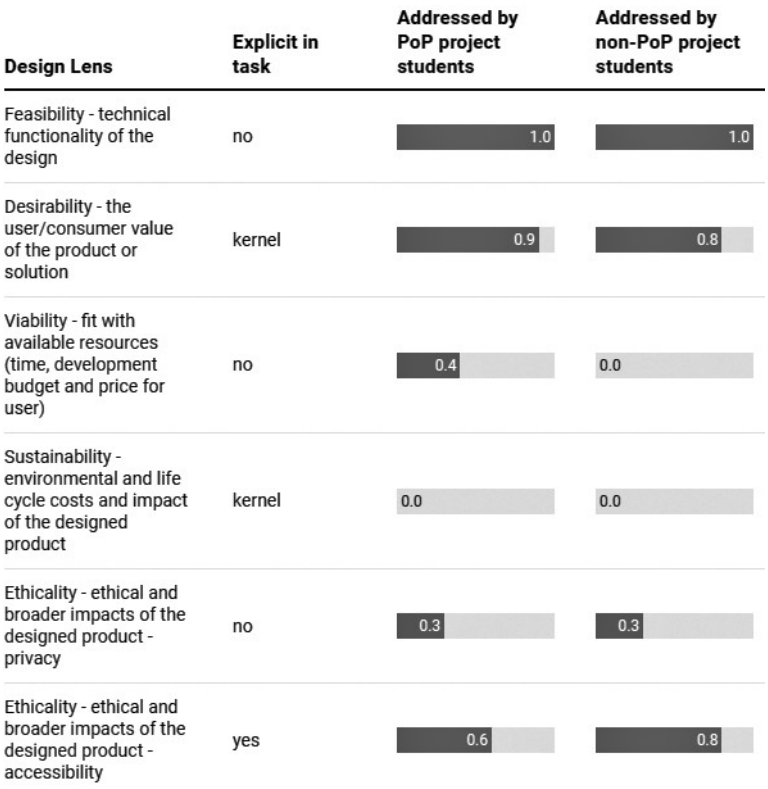

Fig. 2. Summary of students' consideration of the five lenses of design. Numbers within the bars represent the proportion of students in a group who addressed a particular design lens.

review a video recording to describe what they were doing and thinking, generally provides more information about students' approach to problem solving and decision-making [33]. Gary Olson et al. [34] state that think-aloud protocols are highly effective for studying complex thinking processes and are particularly well suited to observing the different ways for going about a task. They have previously been used to study engineering students' problem solving [35–40], and the tasks and the instructions should be constructed to ensure coherence with the research questions. While respecting Linda Baker and Lorraine Cerro's [41] recommendations to avoid cueing participants about how to respond, in this work we did however include kernels that students could develop to engage with each of the five design aspects.

3.2 Task

We created two software design tasks with the scope for students to engage with the five lenses for software design, while not exceeding a difficulty level appropriate for third year computer science undergraduates. We established the validity of the task by asking a set of instructors and teaching assistants of software design courses to examine the tasks and respond to a survey assessing whether the tasks were clear, appropriate, solvable in a limited time and amenable to multiple solutions. We then revised the tasks based on the feedback and chose one that was recommended by more instructors. The task was:

> "The town of Geopolis would like to propose a parking spot finder to facilitate people visiting the city centre. The city would like that the app would both (i) suggest times during the week when the city centre is less busy to reduce pressure during rush hour and (ii) propose specific parking places that are immediately available. The app should work for both residents and tourists. The businesses and environmental advocates think that it is reasonable for people to walk up to 500m from their parking spot to their desired destination, but feedback from citizens suggests that parents of young children, people with mobility issues, and the elderly consider this too far. There may be other issues to consider as well. Your task is to outline a software design for such a parking spot finder that satisfies the requirements of the city and the people."

3.3 Participants

In Spring 2021, we sent all third year computer science students in our institution (N = 98 however 18 students were on exchange at other institutions) an email inviting them to participate in our study if their semester project involved software design. Additional direct invitations were extended to students of the Proof-of-Personhood (PoP) project

[42], a semester project hosted by a research lab that focuses on the design of privacy-preserving software systems. The project is atypical in our institution as 10 or more students get involved as a group in a large-scale software engineering project (50 kLOCs, 40+ contributors). Through the development of a Sybil-resistant pseudonymous online identity demonstrator, PoP exposes students to concerns such as usability, privacy preservation and identity management. Of the 11 male students who accepted to participate, seven came from the PoP project and four students came from four different software design projects. All participants completed the full protocol and were compensated with a 15 CHF payment to their campus card.

3.4 Procedure

We collected direct, qualitative observations of students' actions as they developed their software design by recording their self-narration of their actions (think aloud protocol), followed by a brief interview to enable us to collect more detail about the thinking underpinning their actions. Participants were sent the information sheet and consent form in advance, and instructed to join an online meeting room with a pen and scrap paper on hand. After two initial sessions to ensure alignment between the two interviewers (Author 1, Author 2), participants met individually with one of the researchers online with audio recording. After an introduction that stressed our interest in their process (rather than the final output), participants were asked to choose a pseudonym as an identifier for their data. Students in PoP were constrained to names starting with « p » and other students were obliged to avoid this letter. Participants were then sent the software design task as a PDF and encouraged to start thinking aloud by reading the text aloud. This proved to be an effective prompt and it was rarely necessary to remind students to keep narrating their thinking. Students were left to work without interruption until they reported having completed the task or for 20 minutes, when they were asked if they were ready to stop or would like to continue for another 3 minutes. The think aloud software design task was followed by a brief semi-structured interview. The interview questions, presented in Table 1, did not explicitly inquire about ethics or sustainability to avoid influencing the perceptions in this small cohort of students about was expected of them [43]. Given that the whole procedure took 30–45 minutes, in practice the interviews adhered closely to the prepared questions which sought to explore how students had gone about creating this design and did not probe their opinions or general ideas about software design. The protocol and data management plan was approved by our institutional human research ethics committee.

3.5 Analysis

A transcription of each audio recording was made and imported as a text file into NVivo, a qualitative data analysis software, chosen for its features for grouping and visualising themes across several data sources. As the goal of the analysis was to identify the presence of the five lenses in student design protocols, we analysed the data using deductive thematic analysis [44], applying the framework to code students' protocols for each of the five lenses. Since our first research question focuses on identifying whether a student incorporated a particular lens in their thinking and given that it was a short task, we applied a low threshold for determining the presence of a lens in students' reflections. If a student's utterance was related to the criteria as defined in Section 2, it was coded as present if the student said that it should be addressed or considered (even if no practical or tangible measure was proposed). It was coded as absent if a student explicitly said that it should be disregarded or did not mention it. For instance, the statement "In the end you still have to track the car and have a device in it if you want geolocalization. And, of course, image recognition from the satellite or anything is completely done and would be much too much work to analyse every image at specific times." was coded as *absence of the ethicality issue of privacy* since the concept of geolocalization is mentioned without any reference to the potential misuse of the associated information. On the other hand, the statement "So yeah, I think that would be the easiest way because that way you don't need to actually identify people. It's the best way because

Table 1. The interview questions that immediately followed the think aloud design task

1. Walk me through what you did to solve this question.
2. How did you define the final goal or outcome of your design? What were the key constraints or features that you considered?
3. Do you think your design will work the way that it is supposed to?
4. Where did this idea come from? Why did you decide to do it this way? Did you consider other ways to design the software?
5. How confident do you feel with your final design?
6. How do you evaluate that this is the right approach?
7. How could you verify if the design works? Is there another way you could check your design?

otherwise the problem is much too big'' was coded as *presence of the ethicality issue of privacy* since it mentions concerns about identifying people.

Two researchers (the first two authors of the paper) applied the framework to code four out of eleven transcripts. We then discussed whether we agreed on the others' coding of the presence or absence of each lens, and disagreements were resolved through iterative discussion. This discussion allowed us to define a set of criteria specifying how to code the presence or absence of a lens. Next one author coded the rest of the data for the lenses of feasibility and desirability, while the other coded for the lenses of viability, ethicality and sustainability. Finally, each researcher reviewed the other's work to ensure that we agreed on the labeling of the presence or absence of the five lenses in all the student protocols. Additionally, the word search function in Nvivo was used to verify that we had not overlooked instances of students' referring to aspects of viability, ethicality or sustainability using the following broad set of related terms as queries:

- Viability: cost* money* invest* budget spend*
- Ethicality string1: priva* informat* securit* confident* encrypt*
- Ethicality string2: parent* elder* old* mobilit* special child* bab* disab*
- Sustainability string: sustain* enviro* climat* efficien* power energ*

These words were chosen by considering the terms that appear in the definition of the lens and their synonyms (for instance, cost, money, budget etc. for the financial viability of a design) and terms related to the lens in the context of the given task. For instance, when thinking about the ethicality issue of accessibility in our task, the relevant terms are elderly, parents, babies, mobility, etc. and so these and related terms were included in the search. We also ensured that all the terms that we identified related to the lens in the deductive coding were included in the search.

4. Findings

Our data consists of our direct, qualitative observations of students' actions as they developed their software design and their responses to our interview questions about *what* they did and *why* they proceeded in this way for the given task. Tables 2–5 provide an overall view of the different design concerns explored by students designated by ✎ if it occurred during the think aloud portion and 💬 if during the interview portion of the protocol. When a student explicitly mentioned a concern, we report at least one representative example. Cases where students did not address an aspect are indicated by 💬 and, when possible, a sentence from the transcript where the omission is evident.

4.1 Feasibility and Desirability

These are the core disciplinary aspects of the task, arising from students' software skills and knowledge. While it was difficult to quantify precisely, these issues clearly consumed the vast majority of students' time and thought. In this work, we consider technical feasibility and desirability as defined in the study of engineering design and prototyping [45]. Feasibility refers to the ''ability of the end design to complete some previously defined basic function'' and desirability is defined as ''how the user will engage with the product, whether the user will find the product compelling, and how desirable the product is''. In line with these definitions, we focused on two facets to analyse students' considerations of these lenses: for feasibility, we looked at how they verified or proposed to test the functionality of their design to meet the given requirements and constraints and for desirability, we looked at students' use of user stories which illustrate their thinking about how a user would interact with the product and whether they would find it useful and easy to use. Observations related to these two lenses are discussed in the following paragraphs, and a selection of relevant extracts are presented in Table 2. The legend, also used in subsequent tables, indicates when a comment was made during the think aloud or interview portions.

We observed that only five of eleven students spontaneously checked their solutions while thinking aloud, to ensure that all the requirements were met. However, in response to our interview question about whether the design would work as intended, all students presented coherent ideas for testing their designs such as, running a simulation, user testing, comparing with other similar applications, comparing the output with the expected output, asking others to evaluate the design, checking whether all requirements are met and checking if the identified issues are mitigated. Together these strategies indicate that students considered testing their designs from several different perspectives, such as conceptual, technical and usability as seen in Sasha's response (Table 2) when he suggests three steps for checking the app. Asking others to evaluate the design was cited by students most often and user testing was second. Students indicated that the goal of asking others to evaluate the design was to identify technical flaws, while the goal of user testing was to ensure that the app was usable and working satisfactorily in real-world conditions. This suggests that students are aware of the impor-

Table 2. Student quotes about feasibility and testing their designs

Paolo Okay, then I'll read again just to think to see if some other thing comes to my mind... How would I verify it? Well, probably the best thing to do is some testing, from a user point of view, not from a code point of view, like with absolutely no bias, just going to some users and tell them, well use this application.
Polska So yeah, I'm just going to reread the assignment to see if I answered everything correctly. I would like come to a meeting and ask my engineering peers if they think this would be right. So communicating with the other members of the team, I guess, that are designing the app as well. So maybe, like trying to, to do a general fault of what could go wrong, where everyone would like, give us feedback, and imagine what could go wrong with the design of the thing.
Patricia So, I mean, I think this is a software idea that mostly satisfies this. I'm not sure if I'm missing something. I think there are ways to do trials in small areas of the city, but you can't do just trials with a small amount of people because then the app doesn't work. So, what you could do is then you start with small areas of the city. If you start with quite a busy area, which is and you sort of force anybody who wants to park there to use the app at risk of being fined. And then, this way you could, sort of, you know, go and check if it's just a few streets, for example, you can check if parking spots are free, or not. And if what's said in the app is properly reflected and same with rush hour.
Parker With a limited set of users and a limited area and try to just do some testing of the whole thing with users with prescripted movement from users to know if the core issues work. But after that, if it works, then you have the usual test with feedback from users and procedure to raise issues that will be corrected.
Paul I would have to decide some metrics, which I can measure [...] how many users does it have, in the end? How helpful will it be? And of course, some other metrics, but in the beginning the usage will probably be like it would need to be growing and then at a certain point, we could start to measure if the rush hour was reduced or something like that.
Pineapple I went through and finished this design, and tested it, I would just compare if that's the result I was expecting or not [...] Yeah, I would have a few scenarios where we would use this design or the software and set the initial outcome of the design? And if it matches with what the software output [...] maybe some parts of the design were were not correct.
PoPuser I look yes if everything is well linked [...] I read the statement and see that all the components of the application are coherent and correct. I think the best thing is to confront yourself with other people, to see if they have other things or how. before embarking on the project.
Sasha Yeah okay for this there's multiple ways. Usually what I try to do is just code something quickly see how it looks. And maybe try to refine it. The second step would be to ask someone I know for example and show them the app. They will tell me, give me feedback. And the third step would be like to let the user have a preview of it. And maybe release like a pre release to 100 users which gives you feedback and then you can you can improve on it. But really the first task is just to code it see how it works and try to improve on it. Then ask one person then ask many people.
Luma I think I would have two choices here. The first one would be to create like kind of a simulation, based on a bit of randomness and of having some cars coming in cars leaving and cars staying for a random amount of time and see if it works or not. And the other one would be to keep the app for people to test and see if. So for example, have a smaller parking lot and kind of scale this big parking lot to like a smaller one and maybe 20 people, evaluate how good the app is.
Richard I think I covered the points that are required [...] I'm just reading through it to make sure that I didn't miss any of the requirements. I think that this would meet all the requirements, yes. After I make a design I will recheck every requirement. And check precisely if the requirement is accomplished or not. And we go over each requirement to check that in the app the requirement is checked.
Jeffery So let's check if the requirements are met [...] First, the app will store the state of all parking spots at different times of the day. And if the percentage of unavailable parking spots is above the threshold, it will mark that period as rush hour.

tance of the criteria of feasibility of their proposed designs, and even though they don't always explicitly evaluate feasibility while they are working on a design, they understand how and why such evaluations are undertaken.

The goal of building user stories in human-centered design is to understand the users and their contexts, to develop empathy for them and incorporate their needs integrally into the design. During this design task, almost all students (ten) considered how users would interact with the system, sometimes imagining themselves as the user as seen in Paul, Popuser and Richard's reflections in Table 3. None of the students made an explicit list of stakeholders for the app. However, we do see in Table 3 that students' user stories were quite detailed, incorporating user interactions with the system, users' potential perceptions of the system as a result of their interactions and reflections on what they could do to mitigate potentially negative perceptions. This process of imagining user actions also helped students to identify flaws in their design, such as requirements that they had ignored, possible implementation challenges and constraints not explicitly mentioned in the problem statement. However, most students except Sasha and Luma, did not build user stories from the perspective of the low-mobility user, which was an important constraint in the problem. In fact, as we will see in our findings related to ethicality, having this constraint explicitly specified in the problem did not necessarily mean that students developed empathy for this type of user and incorporated their needs in the design. This points to a serious lacuna in students' understanding and application of user stories for design, and the importance of including different kinds of users in design task assignments.

Table 3. Student quotes on desirability and building user stories

Paolo The application should however give the possibility to the user to extend their parking spot so their car is already there the they already booked the thing so maybe they want to stay one hour more there should be a plus sign like extend your parking and you will press like okay extend for another two hours.
Polska For example, when a user says that he wants to go to the restaurant, the app would search for every parking spot 500 metres away from this restaurant. And it would find the closest parking spot that is not full [. . .] So then the user would reserve his place on the app. And this would reduce the number of places for this parking lot.
Patricia The mental process is sort of thinking of things a bit like user stories. So, um, when you read sort of the constraint that, you know, what you want from the app, you even try to convert it into. . . I mean I think about it from a technical point of view how that would be possible.
Parker So yeah, everyone could receive every parking spot. event, but then it would the app would be crowded with unnecessary information? I mean, it would be yeah it's kind of thing yes on the user experience.
Paul I based it on my own expectation from such a system, I thought, what would I like, I would like to enter my location and then see the closest available parking space. So I've basically done that. And of course a few lines that I read here. But it didn't clearly say you would need to show the closest one. This was based on my own thinking.
Pineapple
PoPuser It's personal experience, for example looking like when you search for locations on Google.
Sasha There should be a preference the user says I want only parking places below a certain meter in their preference, from the destination they're looking for. So this way you can be like if someone is able to walk long distances and is willing to look a bit further, if it means they would find a parking spot. They'll take these options.
Luma If all parking spots are occupied then maybe they do not want to wait, even though they specify that they want to be closer. So I'm, I think people might be unhappy to, if they need to wait for a couple of hours before parking spot is present.
Richard So I imagined myself trying to visit the city, and using the app to find the parking spots and thought. I was thinking about my process about what I would do. To need to find a parking spot and what I would require. So I placed myself into someone that actually uses the app and try to think about what would be most important for me.
Jeffery

Table legend: student comment during think aloud; Student comment during interview; Omitted theme; Ambiguous student comment during think aloud.

Table 4. Student quotes on viability and financial aspects

Paolo Then it would give back the money for the 30 minutes probably minus the price to send the money and little commission of the application.
Parker Some detecting device in the parking spots that knows, realises if a car is on top, so yeah, it would be very expensive and the server. . . that would have to receive loads and loads of information.
Paul But if it would have sensors, then of course, we could get the exact number that might be out of reach. That depends on, of course, the budget and of the whole project, and the accuracy. So there's always a trade off there.

4.2 Viability

Viability refers to "creating prototypes that test the design's likelihood of fitting into time and budget constraints" [45]. While the task was purely at the conceptual design phase, the task assignment introduced many real-world factors. Cost was not explicitly mentioned at all in the task, as is frequently the case with school assignments, despite money being a very present real-world constraint.

We observed three (all PoP students) out of eleven students considering a financial aspect of their proposed design during the think aloud, as reported in Table 4. The remaining eight omitted to say anything about the cost of their design, including Patricia who mentioned that having people or cameras to check if parking spots were occupied would require "coordination", making his omission to consider the cost of these measures quite striking. Among the three students who did mention cost, Paolo referenced the cost of electronic money transfers and Paul the cost of sensors. Parker also considered the cost of sensors and was the only student to identify a cost directly related to software design when he cited "servers. . . and loads of information". Overall, these observations suggest that cost is insufficiently present in students' design thinking processes and, importantly, the costs of software development, maintenance and data management may be overlooked by computer science students. This suggests a critical lack in their curriculum that should be addressed.

4.3 Ethicality

As outlined in the introduction, our use of this term refers to (potential) effects beyond the intended use of the design product for the user, other individuals, communities or society. Given the nature of the software to design, there was scope for students to consider ethicality in terms of data privacy issues related to physical location, payment, and user

Table 5. Student quotes on ethicality, privacy and accessibility

Ethicality: privacy	Ethicality: accessibility
Paolo So as a first approach, one should probably use a database to store people's information. . . And Google Firebase could be one possibility. So surely, one will have to open a project on Google Firebase. . . just take information from the user when it edits. So a profile picture, name, age, whatever could be relevant, maybe not age, the age is not that relevant. But this could be useful also for the settings.	When people who are low mobility look, it will propose [reserved spots] to them, which means that there might be more parking spots reserved for them throughout the city so only propose parking spots in busy areas to low mobility.
Polska So yeah, so what you would enter in the application would be the exact location of where he wants to go. The application would already know if he's an elderly person or not.	We don't want them to have a parking spot at 500 metres, that would be too far for them. So maybe for those people, we would reduce the searching the searching condition, and we could go like, for 200 metres for those persons.
Patricia This would be very tricky to implement without requiring people to give over quite a lot of information [. . .] I think requiring everybody to hand over their location whenever they park in the city to the government is maybe a bit of a violation of privacy [. . .] So it would be maybe interesting to make sure that all of this information is encrypted and not accessible by any city officials [. . .] it's a bit complicated to make work, because umm number plates are associated to people. And so if the city wanted to for example send a fine to somebody, then they would obviously need that number plate and address.	
Parker The cybersecurity problem that you have with this is not actually specific to our problem. Because I think that connected cars will have a lot of cybersecurity problems on their own. So, I mean, that would not be up to us to solve, I guess. [. . .] I'm doing a project where anonymity and accountability is the centre of it. . . So to find a way to ensure privacy of the users and our accountability. And that's why I didn't immediately to have like something engineering detecting cars, rather than users signaling it.	And if it's more than 500 metres at this point I don't know what to do at that point. If there is nothing, maybe suggest the closest anyway. I don't know. And for young children, people with mobility issues.
Paul For the user experience, you have to design something like if the user travels at a certain speed, maybe just a pop up to ask him if he wants to unbook the place. So it might be unfeasible depending on how many cameras and maybe also some privacy restrictions there. So I'll just go for the integration with the parking meter.	I think this 500 metres, this exact number is not that important for the project. I think everybody wants to get the closest one. And if you have the information before, and if. . . you're a mother with young children or have mobility issues, you can then still decide to go take a taxi, which can drive you all the way [and] not park. . . we will try to display all information available about the parking spot. And of course, the distance from the desired location. And then people can make up their own mind to see whether they will park there or not.
Pineapple	We should also qualify the person as an elderly or with mobility issues or children or if it's a simple worker. So, if we have a parking spot that can be occupied by a non-constrained individual or one of the constrained ones we should give priority to the constrained one. . . We should also have a hierarchy I think. . .
PoPuser One important information is the location of the user. We can track all the users of the application by their speed to see if they are in the car. And where they want to park. We can know which parking they have occupied or not.	After reading the instructions, I focused on the first 2 points 'suggest times during the week when the city centre is less busy' and 'spots that are immediately available'. I said to myself that these are the 2 most important things.
Sasha And it could be an issue because we wouldn't want to know like what destination people go to regularly. So they should not be logged into the database. If it's a public app. I don't know if it's private, maybe the company would like to have this kind of information.	So this way if someone is unable to walk long distances is willing to look a bit further, if it means they would find a parking spot. They'll take these options. This could be an issue for elderly people that are not that used to applications. So maybe that could be a way to mitigate that, and I guess, like elderly people usually go to take a [paper] ticket for a parking place.
Luma	So I believe that the elderly, sort of people who need to be closer to the destination, their parking spot should be kind of distributed across the border of this parking centre. . . I think I need to have 2 queues for this. One that is exclusively for people who do not need to be close to their destination.
Richard The first interface is a page where the user enters his destination and condition if he is handicapped or has a family with younger children [. . .] So we just open Google Maps with the given destination, so that would be easier than implementing a personal application with the route.	So what I would think to do is have a bit bigger parking lots where there are more places every one kilometre in the city. And then have additional places like just one or two places in the street maybe just for people with handicaps and families.
Jeffery	So the design should take into account different categories of people for whom this 500 metres is not optimal.

profiles. Further, one criteria of the task related to ethical design considerations through the request to provide additional support or preference to people with certain profiles (mobility issues, young families). Observations related to these two aspects of ethicality are discussed, respectively, in the following two paragraphs and relevant extracts from the think aloud are presented in Table 5.

For the first aspect related to privacy, of the seven students who did not address privacy issues, the omissions are most striking in cases where students proposed using geolocalisation, user profiles with potentially sensitive data, or integration with large data-harvesting platforms like Google. Illustrative quotes are presented in Table 5. Unsurprisingly given the centrality of security issues to the PoP project, we see that the two students (Parker and Patricia) who engaged significantly with the challenge of managing privacy and functionality are PoP students. These students identified potential concerns at several different points in the design process including issues of geolocalisation, user profile data, access by city officials, and issuing parking fines. Given the focus of the PoP project and the project deliverables created by the students, we are confident that all students have the knowledge, awareness and skills to engage with issues of privacy. Our observations in this study pertain to students' ethical sensitivity and ethical motivation [46] about bringing privacy issues into their design thinking. Of the four non-PoP students in our study, Sasha was the only one to consider privacy concerns, however, his comment suggests that such ethical limitations would be of concern to governments and not to private companies. Paul's verbalisations on the concerns he perceives for using geolocalisation data are somewhat ambiguous, and may originate in legal compliance rather than an ethical position on privacy. These observations align with previous work which has found professional computer scientists focus on fulfilling their project briefs [25], without consideration of broader societal implications of their design.

For the second aspect of ethicality, designing for the needs of people for whom walking 500m is problematic, the issue was made explicit in the task. Seven students, as illustrated by the quotes in Table 5, appear to have automatically accepted that such users would require accommodations to address their mobility constraints with respect to parking. However, three students (PoPUser, Paul and Sasha) appear to have not relised that for people with low mobility these limitations are very real barriers to participating in society, with PoPUser excluding them from his list of important factors and both Paul and Sasha implying that providing all users with the same information to make their own choice was sufficient accommodation. For example, Paul's thinking aloud implies that taking a taxi to avoid having to walk long distances is simply a matter of choice, rather than a potential onerous daily cost for people with mobility issues. Paul does however note that elderly people often avoid apps in preference to physical machines, but he does not make any design decisions to address this potential exclusion. This suggests that he is attending to "user desirability" issues and not those of ethicality. Patricia omitted any mention of users with mobility issues from his design. A relevant comparison can be made with the constraint that "the app should work for both tourists and residents" which ten students mentioned (data not shown) compared to only seven students who engaged with the constraints of low mobility people. These four instances of omission or dismissal of the limitations of people with mobility issues from the early design phase indicates a lack of consideration of ethicality in the thinking of approximately half of the computer science students who participated, despite such accommodations being listed explicitly in the task description. This observation aligns with previous work which has found professional computer scientists on the feasibility and implementation aspects in their problem solving [25], apparently interpreting ethicality as beyond their remit.

4.4 Sustainability

An opportunity for students to consider the environmental implications of their design was introduced in the task statement with the phrase "environmental advocates think that. . ." We anticipated that this kernel would serve as a prompt to students to consider how their design could support environmentally sustainable transportation habits. We used Nvivo to directly query the transcripts of the think-aloud and interview portions and confirmed zero instances of sustainability in students' reflections. Although he did not cite sustainability, Richard did consider allowing users to input multiple destinations to get parking between them, therefore reducing driving when he said "you could also enter maybe the multiple destinations, and then a link would be drawn between all the locations to find the nearest parking spot between all the different locations". Students do not appear to spontaneously incorporate sustainability as a transversal, omnipresent contextual responsibility in their design decisions despite the looming climate disaster.

5. Discussion and Implications

Our first research questions focused on how software engineering students incorporate the five

lenses in their design reflections and the qualitative differences in how students engaged with each. While the design task has scope to engage all five lenses, only ethicality in terms of accommodating low mobility people is explicitly listed as a design constraint. As summarised in Fig. 2, feasibility, viability and ethicality in terms of privacy were not explicitly cited in the task, while desirability and sustainability were introduced obliquely. We have termed these tangential references as ''kernels'' because they could serve as a basis for students to develop consideration of these ideas in their design. In answer to the first research question, we found that at least 3 students engaged with each of the lenses except sustainability. We summarize the differences in how individual students attended to different lenses in qualitatively different ways, seen in Tables 2–5, in the following paragraphs. In answer to our second research question, we found the rich observations produced by our protocol could serve to evaluate the impact of training targeting the application of 5 lenses in software design. We did not find appreciable differences between students who participated in PoP vs another project, potential reasons for which we will discuss below.

Feasibility was the only lens present in all 11 students' thinking, despite it being one of the three lenses which were not explicitly flagged in the task. This finding is not surprising; given the nature of the task and that students were explicitly recruited to work on a software design task, they appear to have seen the conceptual, technical and usability aspects as fundamental to the task. The ethicality of privacy was the second lens not introduced in the task; it was addressed by approximately one-quarter of students. There was no qualitative difference between students who participated in PoP and other students, despite privacy being a central consideration in the PoP project. With the caveat of the small number of participants, we could surmise that students do not see privacy as an omnipresent concern in software design. Another explanation could be that students do not accord themselves the authority to introduce additional design aspects. An interesting parallel to hierarchical 4-levels of moral reasoning [44] can be made here, where training targeting the two lower levels (ethical awareness and ethical sensitivity) has been shown to be insufficient for engineering students to develop ethical motivation or ethical agency [47]. Perhaps the larger ''culture of disengagement'' [48] within engineering education may have contributed to why, despite the emphasis on privacy issues within PoP, students did not consider it as integral to all their software design. This is coherent with the findings of Bednar and colleagues [25] who report that two-thirds of senior IT professionals they interviewed had negative attitudes about implementation of privacy, due to it being inconvenient, boring or difficult, despite half of the respondents stating that privacy is sensible or important. Bednar and colleagues [25] also found that IT professionals do not experience pressure from the public about how they manage privacy, resulting in its remaining subject to their organisational context. Finally, Bednar and colleagues [25] observed almost half of comments by IT professionals about who is responsible for privacy fell into the category of either ''not me'' or ''someone else''. This connects to other findings within engineering education which show that students consider ethics to be extraneous to the responsibilities of engineering [11] and their perception of their own social responsibility often decreases during the course of their engineering training [13]. However, studies also suggest that the integration of ethics and social responsibility across the curriculum, with a systematic pedagogical approach can change students' views and make them more socially responsible designers [11, 13, 49].

Viability is the third aspect not introduced in the task; it was addressed by three PoP students (approx half of PoP students representing one-quarter of all students). As shown in Fig. 2, this is the main difference between PoP students and other students. While viability is not relevant for projects that remain at a conceptual stage, it is a very real constraint susceptible to being overlooked by students' due to their poor financial literacy [50, 51]. The PoP teaching team independently identified students' lack of awareness of cost during their teaching activities, and this analysis suggests that they have been more successful than other projects in prompting students to consider cost.

Comparing the two lenses that were referred to obliquely in the task, sustainability was ignored by all students, both in terms of the (likely very small) direct environmental impact of creating a new app and also the (potentially larger) opportunity to influence driving and transportation emissions. This omission suggests that this is not a criteria within the students' consideration and that our kernel mentioning ''environmental advocates'' was not sufficient to stimulate students to include sustainability in their design thinking. This is perhaps to be expected if sustainability was not introduced in their prior design courses. Indeed, ''sustainability'' in software design often refers to the business model or product lifespan, rather than environmental sustainability. One example is how ''sustainability'' is used in the Principles for Digital Development guidelines [52] and the absence of environmental concerns from the Ethical AI prin-

ciples [53]. It appears that environmental sustainability is currently peripheral to software design considerations.

Desirability, present as a kernel in the task through the inclusion of user requests, was the second most commonly addressed lens (nine students) through their construction of detailed user stories to explore the requirements and evaluate their designs. All students would have been introduced to strategies for building user stories for design during their core disciplinary courses. However, it seems that this training remained at a practical level and did not prompt them to develop empathy for the user and consider all aspects of their context, as evident from their consideration of ethicality.

Ethicality in terms of accommodating low mobility people was an explicit constraint within the task. While seven students addressed this constraint in their design, it was surprising that the remaining students explicitly ignored this requirement in their conceptual designs. Carla Zoltowski and colleagues [54] found that students' conception of human-centered design develops from technology-centered to empathic design, and their understanding of and integration of users into the design evolves as their conception of design becomes more comprehensive. The most comprehensive category of design, empathic design, requires designers to have a broad and deep connection to users and understanding of the context. Our findings agree with this as we see that while students created user stories, these were often limited in their scope to either students' own expectations from such a system or expectations from a ''normative'' user, without considering the needs of a wider range of users. This places our students in category 4 of Zoltowski et al.'s framework, ''keeping users' needs in mind'' within the student ways of experiencing human-centered design. This suggests that computer science students would benefit from more ''critical or immersive experiences'' involving real clients and users to develop a more comprehensive understanding of human-centered design.

Our qualitative approach sought to develop a rich description of students' thinking during a software design task and not to produce generalisable results. The major potential threat to the validity of our findings is the short observation period with a single task where each lens is explicit, implicit or absent in the design brief. This sets up a specific context where some lenses could appear more or less relevant to students, irrespective of their general approach. Another limitation could be bias arising from non-representativity of the 11 male students who volunteered to participate; a number determined by those who choose to volunteer rather than by conceptual saturation. While the lack of gender diversity is evident, a larger group of students may have introduced additional perspectives that would influence the overall picture created by this small sample.

Questionnaires probing students' conceptions of ethics or sustainability are effective for surveying large numbers of students. One such example is the DIT which employs case studies to assess students' moral reasoning [46]. While the engineering version of the DIT cases are about engineers and engineering decisions, they present as an exercise on ethics rather than being immersed within technical, disciplinary considerations. Per our second research question, we are interested in how to generate authentic, contextual observations about how students think about design, ethics and sustainability. The findings discussed above suggest that our design task and think aloud protocol were effective in allowing students to exhibit a range of different approaches to the five lenses within a 30 minute session. While longer sessions would likely have provided more insight into students' thinking, the scale of engineering programmes means that such methods are too resource-intensive to be broadly or frequently used. In this context, we find our think aloud design task provides a resource-light, non-invasive, contextualised way to observe how students mediate ethicality and sustainability in their disciplinary design thinking.

Approaches to integrating ethicality or sustainability in engineering design courses have been reported [49, 55–58], including some particularly interesting examples from civil engineering that teach both inclusion and sustainability [59, 60]. Laura Fernandez-Robles et al.'s call for clearer guidelines in professional engineering work to provide guidance on how and when to incorporate ethicality, social responsibility and sustainability is also pertinent [61]. Our findings underline the importance of integrating these aspects into how software design and offer a way to evaluate the impact of such interventions. We argue that software engineering students should be introduced to a broader set of design criteria that explicitly includes sustainability and ethicality in addition to the traditional criteria of feasibility, desirability and viability, as shown in Fig. 1. Incorporating these issues earlier and more transversally across the curriculum is one potential way to address the worrying observations that engineering students' ethical thinking advances more slowly than students from many other fields [14] and that professional computer scientists generally only include ethical concerns if they are explicitly told to do so [25]. Secondly, our five lenses and design thinking protocol offers a concrete way to assess progress towards a software engineering culture grounded in a system of human values and ethics [62].

6. Conclusion

In this paper, we present an approach to study the human and valued centered design processes of engineering students using a five-lens design framework using a think aloud protocol with students working on a software design task. The findings from our thematic analysis of 11 computer science students' protocols suggest that viability, ethicality and sustainability were generally absent from students' in-the-moment thinking (included in less than one-quarter of students' designs when not an explicit design constraint). We do not suggest that this means that students do not care, or know, about these issues but rather that they do not transversally incorporate these important themes into their disciplinary thinking. While the fact that these students still have 2 years of training to finish their education as software engineers should be kept in mind, our findings indicate that such considerations are not integrated across their design experiences and (at best) come later in their education, if at all. Based on these findings, we argue that contextualized teaching, and assessing, of engineering students' integration of ethicality and sustainability should be foundational to their software design training.

The degree to which the different lenses were foregrounded in the assigned task is important, as students could perceive these as prompts to address some aspects but not others. The omission of sustainability considerations, despite the kernel in the problem statement, suggests this theme is not at the forefront of students' thinking. Indeed, how students interpret and pick up on the different lenses is also a key insight of this study – what aspects do computer science students see as germane to a software design task? What aspects/lenses do they omit or intentionally marginalise? We found that individual students incorporated different lenses to different degrees. Given the short time allocated to students for their software design in this study, it is not reasonable to expect each student's thinking to address all five aspects in a rigorous way. Rather we are interested in the overall picture of what students do and do not include, and whether they will have a sufficiently robust perspective to guide their work as professional engineers to contribute constructively to civil society and the environment. Our observation that these issues are omitted is coherent with previous work that has found that engineering students ethical development is insufficient and supports our argument that computer science students should be actively provided with discipline-appropriate tools to integrate ethicality and sustainability into their designs. In future work, we will investigate if using tangibles to develop their design prompts students to incorporate all five lenses in their software design, as well as how to develop students' ethical motivation and agency to take an active role in setting the priorities for future software design.

Acknowledgements – The authors would like to thank Dr. Vero Estrada-Galiñanes for her useful feedback on this manuscript and Prof. Pierre Dillenbourg for his material support of this project.

References

1. E. Blevis, Sustainable interaction design, *Proceedings of the SIGCHI conference on Human factors in computing systems – CHI '07*, New York, April 2007, pp. 503–512, 2007.
2. Human-Centered Design Toolkit, https://www.ideo.com/work/human-centered-design-toolkit/, Accessed July 15, 2022.
3. M. Steen, Tensions in human-centred design, *CoDesign*, **7**(1), pp. 45–60, 2011.
4. M. Maguire, Methods to support human-centred design, *International journal of human-computer studies*, **55**(4), pp. 587–634, 2001.
5. E. Manzini, Design, ethics and sustainability, *Guidelines for a Transition Phase. University of Art and Design*, Helsinki, June, pp. 9–15, 2006.
6. B. Friedman, P. Kahn and A. Borning, Value sensitive design: Theory and methods, *University of Washington technical report*, 2-12-2001.
7. D. Mougouei, H. Perera, W. Hussain, R. Shams and J. Whittle, Operationalizing Human Values in Software: A Research Roadmap, *Proceedings of the 26th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '18)*, Lake Buena Vista, FL, USA, November 4–9, 2018.
8. J. Donia and J. Shaw, Ethics and Values in Design: A Structured Review and Theoretical Critique, *Science and Engineering Ethics*, **27**(5), 2021.
9. B. Berenbach and M. Broy, Professional and ethical dilemmas in software engineering, *Computer*, **42**(1), pp. 74–80, 2009.
10. I. Szekely. What do IT professionals think about surveillance? in C. Fuchs, K. Boersma, A. Albrechtslund, and M. Sandoval (eds), *Internet and surveillance: The challenges of Web 2.0 and social media*, vol. 16, Routledge, New York, NY, pp. 198–219, 2013.
11. J. Lönngren, Exploring the discursive construction of ethics in an introductory engineering course, *Journal of Engineering Education*, **110**(1), pp. 44–69, 2021.
12. R. Tormey, I. LeDuc, S. Isaac, C. Hardebolle and I.V. Cardia, The formal and hidden curricula of ethics in engineering education. *Proceedings of the 43rd SEFI Annual Conference 2015 – Diversity in Engineering Education: An Opportunity to Face the New Trends of Engineering – SEFI 2015*, Orléans, France, June 29–July 2, 2015.
13. A. R. Bielefeldt and N. E. Canney, Changes in the Social Responsibility Attitudes of Engineering Students Over Time, *Science and Engineering Ethics*, **22**(5), pp. 1535–1551, 2016.

14. J. O'Flaherty and J. Gleeson, Longitudinal study of levels of moral reasoning of undergraduate students in an Irish university : the influence of contextual factors, *Irish Educational Studies*, **33**(1), pp. 57–74, 2014.
15. H. Nguyen Ngoc, G. Lasa and I. Iriarte, Human-centred design in industry 4.0: case study review and opportunities for future research, *Journal of Intelligent Manufacturing*, **33**, pp. 35–76, 2022.
16. B. Knowles, O. Bates and M. Håkansson. This changes sustainable HCI. *Conference on Human Factors in Computing Systems – Proceedings*, Montréal, QC, Canada, April, 1–12, pp. 1–12, 2018.
17. https://www.cti-commission.fr/fonds-documentaire/document/25/chapitre/1217 Accessed July 13, 2022.
18. https://www.abet.org/equipping-graduates-to-build-a-sustainable-world/ Accessed July 13, 2022.
19. How Green Is Your Software?, https://hbr.org/2020/09/how-green-is-your-software, Accessed 22 March 2022.
20. Cambridge Bitcoin Electricity Consumption Index, https://ccaf.io/cbeci/index, Accessed 25 March 2022.
21. 10 Recommendations for Green Software Development, https://greensoftware.foundation/articles/10-recommendations-for-green-software-development, Accessed 24 March 2022.
22. N. Stegall, Designing for Sustainability: A Philosophy for Ecologically Intentional Design, *Design Issues*, **22**(2), pp. 56–63, 2006.
23. A. J. Thomson and D. L. Schmoldt, Ethics in computer software design and development, *Computers and Electronics in Agriculture*, **30**(1–3), pp. 85–102, 2001.
24. J. Stoyanovich, S. Abiteboul and G. Miklau, Data, Responsibly: Fairness, Neutrality and Transparency in Data Analysis, *International Conference on Extending Database Technology*, Bordeaux, France, March 2016.
25. K. Bednar, S. Spiekermann, and M. Langheinrich. Engineering Privacy by Design: Are engineers ready to live up to the challenge? *The Information Society* **35**(3), pp. 122–142, 2019.
26. L. Dombrowski, E. Harmon and S. Fox. Social justice-oriented interaction design: Outlining key design strategies and commitments. *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, Brisbane, QLD, Australia, June 4–8, pp. 656–671, 2016.
27. J. Barroso, L. M. Lopez, H. Paredes, F. Puehretmair and T. Rocha, Special issue on accessibility and software design for all, *Universal Access in the Information Society*, **19**(3), pp. 483–484, 2020.
28. D. Paiva, M. Barroso, A. P. Freire and R. P. de Mattos Fortes, Accessibility and Software Engineering Processes: A Systematic Literature Review, *The Journal of Systems & Software*, **171**, p. 110819, 2021.
29. A. Kukulska-Hulme, M. Gaved, A. Brasher, A. Jones, E. Scanlon and L. Paletta, Designing for inclusion through incidental language, *ICT in Language Learning (5th Edition)*, Florence, Italy, 15–16 Nov, 2012.
30. A. Jones, A. Kukulska-Hulme, L. Norris, M. Gaved, E. Scanlon, J. Jones and A. Brasher, A field trial Supporting immigrant language learning on smartphones, *Studies in the Education of Adults*, **49**(2), pp. 228–252. 2018.
31. J. M. Case and G. Light, Emerging research methodologies in engineering education research, *Journal of Engineering Education*, **100**(1), pp. 186–210, 2011.
32. A. K. Ericsson and H. A. Simon, How to study thinking in everyday life: Contrasting think-aloud protocols with descriptions and explanations of thinking, *Mind, Culture, and Activity*, **5**(3), pp. 178–186, 1998.
33. H. Kuusela and P. Paul, A comparison of concurrent and retrospective verbal protocol analysis, *American Journal of Psychology*, **113**(3), pp. 387–404, 2000.
34. G. M. Olson, S. A. Duffy and R. L. Mack, Thinking-Out-Loud as a Method for Studying Real-Time Comprehension Processes, in D. E. Kieras, M. A. Just (eds), *New methods in reading comprehension research*, pp. 253–286. Hillsdale, New Jersey. Lawrence Erlbaum Associates, Publisher. 2018.
35. H. H. C. M. Christiaans and K. H. Dorst, Cognitive models in industrial design engineering: a protocol study, *Design Theory and Methodology*, **42**(1), pp. 131–140, 1992.
36. C. J. Atman and K. M. Bursic, Verbal protocol analysis as a method to document engineering student design processes, *Journal of Engineering Education*, **87**(2), pp. 121–132. 1998.
37. P. R. Pintrich, C. A. Wolters and G. P. Baxter, 2. Assessing metacognition and self-regulated learning, in G. Schraw & J. C. Impara (eds), *Issues in the Measurement of Metacognition*, Buros Institute of Mental Measurements, Lincoln, NE, p. 43, 2000.
38. C. J. Atman, M. E. Cardella, J. Turns and R. Adams, Comparing freshman and senior engineering design processes: an in-depth follow-up study, *Design Studies*, **26**(4), pp. 325–357, 2005.
39. R. Taraban, A. DeFinis, A. G. Brown, E. E. Anderson and M. P. Sharma, A paradigm for assessing conceptual and procedural knowledge in engineering students, *Journal of Engineering Education*, **96**(4), pp. 335–345, 2007.
40. T. A. Litzinger, P. Van Meter, C. M. Firetto, L. J. Passmore, C. B. Masters, S. R. Turns, G. Gray, F. Costanzo and S. E. Zappe, A cognitive study of problem solving in statics, *Journal of Engineering Education*, **99**(4), pp. 337–353, 2010.
41. L. Baker and L. C. Cerro. 3. Assessing Metacognition in Children and Adults, *Issues in the Measurement of Metacognition*, **4**, 2000.
42. B. Ford, Identity and Personhood in Digital Democracy: Evaluating Inclusion, Equality, Security, and Privacy in Pseudonym Parties and Other Proofs of Personhood, *arXiv preprint arXiv:2011.02412*, 2020.
43. R. M. Lee and D. Riesman and the sociology of the interview, *Sociological Quarterly*, **49**, pp. 285–307, 2008.
44. V. Braun and V. Clarke, Thematic analysis, in H. Cooper (Ed), APA Handbook of Research Methods in Psychology: Vol. 2. Research Designs, American Psychological Association, 2012.
45. J. Menold, K. Jablokow and T. Simpson, Prototype for X (PFX): A holistic framework for structuring prototyping methods to support engineering design, *Design Studies*, **50**, May 1, pp. 70–112, 2017.
46. J. Rest, *DIT manual*. Minneapolis, MN: Center for the Study of Ethical Development, University of Minnesota, 1986.
47. D. Bairaktarova and A. Woodcock, Engineering student's ethical awareness and behavior: A new motivational model. *Science and Engineering Ethics*, **23**(4), pp. 1129–1157, 2017.
48. E. A Cech, Culture of Disengagement in Engineering Education?, *Science Technology and Human Values*, **39**(1), pp. 42–72, 2014.
49. E. Gimenez-Carbo, M. Gomez-Martin and I. Andres-Domenech, Assessment of Ethical, Environmental and Professional Responsibility Training of Civil Engineers, *International Journal of Engineering Education*, **38**(3), pp. 621–30, 2022.
50. T. Binoy and P. Subhashree, Factors that influence financial literacy among engineering students, *Procedia Computer Science*, **172**, pp. 480–487, 2020.
51. K. Ergün, Financial literacy among university students: A study in eight European countries, *International Journal of Consumer Studies*, **42**(1), pp. 2–15, 2018.

52. Principles for Digital Development, https://digitalprinciples.org/principle/build-for-sustainability/, Accessed 22 March 2022.
53. The Responsible Machine Learning Principles, https://ethical.institute/principles.html, Accessed 22 March 2022.
54. C. B. Zoltowski, W. C. Oakes and M. E. Cardella, Students' Ways of Experiencing Human-Centered Design, *Journal of Engineering Education*, **101**(1), pp. 28–59, 2012.
55. M. P. Rosa, Experimental Education of Collaborative Design – the Case of an Inclusive Bus Stop for a Tourist Transportation Hub, *International Journal of Engineering Education*, **38**(3), pp. 589–99, 2022.
56. A. K. Winkens and C. Leicht-Scholten, Teaching Essential Competencies for Social and Sustainable Engineering Design-Case Study of a Research-Oriented Master's Seminar, *International Journal of Engineering Education*, **38**(3), pp. 600–10, 2022.
57. R. Prabhu, E. Starkey and M. A. Alzayed, Exploring the Relationship between Students' Trait Empathy, their Attitudes Towards Sustainability, and their Reflections on a Workshop on Sustainable Design, *International Journal of Engineering Education*, **38**(3), pp. 695–708, 2022.
58. Y. Jalali, C. Matheis and V. K. Lohani, Imagination and Moral Deliberation: A Case Study of an Ethics Discussion Session, *International Journal of Engineering Education*, **38**(3), pp. 709–18, 2022.
59. T. Siller, R. A. Atadero, A. M. Casper and C. H. Paguyo, Leveraging Sustainability to Teach About Social Justice in Civil Engineering Curricula, *International Journal of Engineering Education*, **38**(3), pp. 742–55, 2022.
60. K. Jahan, S. Farrell, H. Hartman and T. Forin, Integrating Inclusivity and Sustainability in Civil Engineering Courses, *International Journal of Engineering Education*, **38**(3), pp. 727–41, 2022.
61. L. Fernandez-Robles, M. Castejon-Limas, A. Gutierrez-Fernandez, F.J. Rodriguez and C. Fernandez-Llamas, Analytical Framework to Investigate Ethics, Social Responsibility and Sustainability in Engineering Project Management. *International Journal of Engineering Education*, **38**(3), pp. 673–83, 2022.

Siara Isaac is the project manager and a researcher in the 3T PLAY project at the École polytechnique fédérale de Lausanne (EPFL). Her work focuses on how students develop transversal skills in science and engineering contexts, including her 2022 book with Roland Tormey entitled *Facilitating Experiential Learning in Higher Education – Teaching and Supervising in Labs, Fieldwork, Studios, and Projects* (Routledge). She has a particular interest in the development of students' epistemic proficiency around managing ambiguity and exercising judgement. Siara has previously worked as teaching advisor and has taught in Canada, China, France and Switzerland. She holds a MSc in Chemistry from McGill University and a PhD in Educational Research from Lancaster University.

Aditi Kothiyal is an Assistant Teaching Professor at the Indian Institute of Technology Gandhinagar affiliated with the Center for Creative Learning. Previously, she was a researcher at the CHILI Lab at the Swiss Federal Institute of Technology in Lausanne (EPFL) where her work focussed on design, orchestration and learning analytics of educational robotics learning activities. Her research interests lie at the boundaries of engineering education, educational technologies and embodied learning, specifically how educational technologies can be used to create real-world embodied problem-solving and learning experiences for engineering undergraduates. She received a PhD in Educational Technology from the Indian Institute of Technology Bombay and a MSc in Electrical Engineering from The Ohio State University, USA.

Pierluca Borsò is head of engineering in the Decentralized/Distributed Systems (DEDIS) lab at the Swiss Federal Institute of Technology in Lausanne (EPFL). He has considerable industry experience in software engineering, IT infrastructure management and data science, as well as project and team management.

Bryan Ford leads the Decentralized/Distributed Systems (DEDIS) lab at the Swiss Federal Institute of Technology in Lausanne (EPFL). He focuses broadly on building secure decentralized systems, including privacy and anonymous communication, systems security, and blockchain technology. Since earning his PhD at MIT, Bryan has held faculty positions at Yale University and EPFL.