

2021-09-02

The Development of Teaching Case Studies to Explore Ethical Issues Associated with Computer Programming

Michael Collins

Technological University Dublin, michael.collins@tudublin.ie

Damian Gordon

Technological University Dublin

Dympna O'Sullivan

Technological University Dublin

Follow this and additional works at: <https://arrow.tudublin.ie/scschcomcon>



Part of the [Other Computer Sciences Commons](#), and the [Programming Languages and Compilers Commons](#)

Recommended Citation

Gordon, D., Collins, M., & O'Sullivan, D. (2021). The Development of Teaching Case Studies to Explore Ethical Issues Associated with Computer Programming. Technological University Dublin. DOI: 10.21427/NWGH-KT02

This Conference Paper is brought to you for free and open access by the School of Computer Sciences at ARROW@TU Dublin. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@TU Dublin. For more information, please contact arrow.admin@tudublin.ie, aisling.coyne@tudublin.ie, gerard.connolly@tudublin.ie.



This work is licensed under a [Creative Commons Attribution-NonCommercial-Share Alike 4.0 License](#)
Funder: European Union

The Development of Teaching Case Studies to Explore Ethical Issues Associated with Computer Programming

Four Case Studies on Programming Ethics.

Damian Gordon
ASCNet Research Group,
Technological University Dublin,
Dublin, Ireland
Damian.X.Gordon@TUDublin.ie

Michael Collins
ASCNet Research Group,
Technological University Dublin,
Dublin, Ireland
Michael.Collins@TUDublin.ie

Dympna O’Sullivan
ASCNet Research Group,
Technological University Dublin,
Dublin, Ireland
Dympna.OSullivan@TUDublin.ie

ABSTRACT

In the past decade software products have become pervasive in many aspects of people’s lives around the world. Unfortunately, the quality of the experience an individual has interacting with that software is dependent on the quality of the software itself, and it is becoming more and more evident that many large software products contain a range of issues and errors, and these issues are not known to the developers of these systems, and they are unaware of the deleterious impacts of those issues on the individuals who use these systems. The authors of this paper are developing a new digital ethics curriculum for the instruction of computer science students. In this paper we present case studies that were explored to demonstrate programming issues to First Year Computer Science students. Each case study outlines key issues associated with a particular scenario and is accompanied by specific questions to be used by the instructor to allow students to begin to reflect on, and evaluate, the implications of these issues. The objective of this teaching content is to ensure that the students are presented with, and engage with, ethical considerations early in their studies and well before they encounter them in an employment setting.

CCS CONCEPTS

• **Social and professional topics;** • **Professional topics;**

KEYWORDS

Programming Issues, Programming Ethics, Ethical Case studies

ACM Reference Format:

Damian Gordon, Michael Collins, and Dympna O’Sullivan. 2021. The Development of Teaching Case Studies to Explore Ethical Issues Associated with Computer Programming: Four Case Studies on Programming Ethics.. In *United Kingdom and Ireland Computing Education Research conference. (UKICER ’21)*, September 02, 03, 2021, Glasgow, United Kingdom. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3481282.3481293>



This work is licensed under a Creative Commons Attribution International 4.0 License.

UKICER ’21, September 02, 03, 2021, Glasgow, United Kingdom
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8568-8/21/09.
<https://doi.org/10.1145/3481282.3481293>

1 INTRODUCTION

This research is part of an Erasmus+ project, Ethics4EU [12], which is exploring issues around the teaching of ethics in computer science curricula. To achieve this goal a number of steps have already been undertaken, including a Pan-European survey of the attitudes of computer science departments in 61 universities across 23 different European countries towards the teaching of computer ethics. This found that 36% of respondents (or 22 universities) do not teach any computer ethics, citing either a lack of available time or a lack of expertise as being the key reasons as to why they don’t teach this topic [13]. Research has consistently shown that ethics is an important missing element in computer science education unlike all other science disciplines [16, 34], thus one of the key objectives of this project is to develop a range of teaching content and instructor guides to encourage the instruction of digital ethics to university students. Given the ethics issues particularly related to programming issues, a number of resources are being developed in that area. In this paper we present sample educational content that was developed as part of the project, specifically lessons that focus on how issues with software can adversely impact a wide range of people. The content is designed to serve as a way to improve computer science students’ ability at consequence scanning – a way to consider the potential consequences of new software on people, communities and the planet [10].

The ethical lenses that are being used in the Ethics4EU project are the three classic moral theories which explore the ethics of actions and their consequences [19]:

- **Deontology:** This model of ethics argues that the morality of a scenario should be judged on the actions taken in that scenario and whether they adhere to a set of principles. Therefore, actions such as the Volkswagen emission scandal, where the Volkswagen Group altered software to make it appear their cars were scoring better in their emissions tests than they should have been [21], or the British post office scandal [9] whereby between 1991 and 2015, over 900 post office workers were prosecuted for alleged theft, false accounting and/or fraud due to known software errors in the post office accounting systems which reported that there were financial discrepancies, resulting in wrongful imprisonment, loss of reputation and livelihood, and bankruptcy [23].
- **Utilitarianism:** This model of ethics argues that the morality of a scenario should be judged on the consequences of the scenario, and the actions taken to achieve these consequences are not important. For example, if a software system

has errors in it, whether those errors are accidental or deliberate is irrelevant, all that matters are the consequences of those errors. There are numerous well-known examples of this, including the destruction of the 1962 Mariner 1 space probe, which cost the US government \$160 million in 2021 adjusted dollars [22], the failure in the propulsion system of the United States Navy cruiser USS Yorktown [33], the simple programming error that caused a number of bank terminals for the Bank of Queensland, Australia to be inoperable for up to a week [35], the Boeing 787 Dreamliner computer systems that needed to be rebooted every 248 days [2], or the English couple who switched electrical supplier and their monthly bill went from £87 to £53,480,062 due to a minor programming error [5].

- *Virtue Ethics*: This model of ethics argues that the morality of a scenario rests solely on the character and intentions of the individual in the scenario, therefore neither their actions or the consequences of those actions are as important as the virtues and vices of the individual involved. For example, professional codes of conduct emphasize the responsibility of the individual to conduct themselves in a virtuous way [17], and this applies when governments make public health choices also [3].

It is inevitable that errors will occur in computer programming [36], but whether these occur accidentally or deliberately, what is important, in an ethical sense, is whether or not the individuals and organisation responsible take accountability for these errors.

The history of computer ethics is almost as old as computers themselves, dating back to the works of Weiner and Weizenbaum almost 70 years ago [7]. In 1985 James Moor defined computer ethics as “*the analysis of the nature and social impact of computer technology and the corresponding formulation and justification of policies for the ethical use of such technology*”, and in the same paper Moor argued that even if computers make it possible for a someone to do a vast number of things, the question must be asked whether one ought to do them [26]. In the same year Deborah Johnson published her seminal book “Computer Ethics” where she noted that computers introduce new and unfamiliar ethical dilemmas and moral problems that did not exist previously [20]. In the 1990s, the concept of “*value-sensitive computer design*” emerged, focusing on designing new technology in ways that prevent harm to human values [15]. At the same time, others including Donald Gotterbarn [17] pointed out that computing ethics should be part of a professional code of conduct, which resulted in the development of a number of codes of ethics and codes of conduct for computing professionals, for example, the ACM Code of Ethics [1]. Recent work on programming ethics by Fiesler *et al.* proposes a new way forward for incorporating ethics into programming modules, by focusing on incorporating the ethical content into the assignments and assessments and leaving the teaching content unchanged, which has delivered promising results [14]. Although the Ethics4EU project is taking a different approach to the development of content, even so, the work of Fiesler *et al.* will be incorporated into the project.

To help first year computer students understand some of the potential pitfalls of working in the software development industry and the importance of ensuring that technology is fit for purpose,

case studies were chosen to highlight some of these troubling issues. A case study method was chosen as these are designed to explore real-world phenomena and they focus on interpreting events and exploring the impact of the case study on the broader society, including ethical issues [11, 24] which are in contrast to much of the teaching that is typically more quantitative in nature. This may mean a change of teaching style for computer science lecturers who are typically instructing students on content, which is well understood and not subject to significant individual interpretations whereas the goal of teaching digital ethics is to explore a range of ethical questions and dilemmas and to allow the students to make up their own minds what they would or would not do in terms of their professional practice. This represents a change in teaching paradigm from a more Behaviourist approach towards a more Constructivist approach [31], and to address this change of teaching approach, a number of instructor guides are being developed to accompany this content.

2 METHODS

This experiment is a qualitative research study using a Participatory Action Research (PAR) methodology [18], that focuses on the following research questions:

1. Do computer ethics case studies highlight the importance of ethics for computer professionals?
2. Does delivering the computer ethics case studies in a Constructivist manner help students see the case from multiple perspectives?

The lessons were delivered using paired teaching [18], and to evaluate the outcomes of the case studies, the lecturers recorded their reflections about the classes on a weekly basis in diaries. After the lessons were completed, the students were invited to participate in a survey to collect their feedback on the process. This process was developed with reference to the BERA Guidelines to ensure that it adhered to ethical educational research practices [6].

The case studies are delivered in a constructivist environment, where each class had three parts to it: (1) the main lesson delivered by a pair of lecturers introducing the topic for consideration, (2) the discussion session where students were put in break-out rooms and took their time to discuss the key points of the lesson, as well as consider some questions supplied to them, and (3) the sharing session, where students shared their thoughts with the entire class, either using the on-line classroom, or in an anonymous note-making environment (Padlet). This structure was inspired by the TPACK framework [25], and it allowed the students to develop an ownership of the content, and to construct their own meaning of the lessons being taught [4]. To help students anticipate the potential outcomes of software systems (i.e. consequence scanning), they need to reflect on three questions:

1. What are some of the potential consequences (intended and unintended) of this software?
2. Which of those are positive consequences, and how can we enhance them?
3. Which of those are negative consequences, and what should we do to mitigate them?

As this content was delivered as part of a first-year computer science programming module, the case studies were designed to

be accessible to computer novices, and focused specifically on topics that are directly relevant to other content being delivered in this module [28]. There are a number of pre-existing computer ethics case studies, for example, those provided by the ACM (<https://ethics.acm.org/integrity-project/case-studies/>), but it was felt that the lessons would be more effective if case studies were identified that would be both relevant to the students and of interest to them. To help brainstorm some of the relevant content, a PESTLE (Political, Economic, Sociological, Technological, Legal and Environmental) analysis was undertaken [37], yielding four key areas that were deemed to be relevant and appealing to the students: Social (the 2020 Irish state examinations), Political (the 2016 political bias of Google), Technical (autonomous cars), and Legal (judicial sentencing software), each of which will be detailed in section 3 of this paper.

3 CONTENT DEVELOPED

Four case studies were carefully selected and developed, which focused on a number of programming-related ethical scenarios. One case study per week was delivered (over 4 weeks) as part of a first-year programming module to allow the students time to assimilate the ethical learning outcomes related to that particular case study and also understand and reflect on the nature of the programming scenario involved.

3.1 Irish State Examinations 2020

The Covid-19 pandemic had a profound effect on the Irish state's Leaving Certificate examination in 2020. This examination is offered to all students who have completed the senior cycle of their secondary education (high-school), the results of which serve as the Irish state's university matriculation as well as offering apprenticeships, job-training, etc., to those wishing to take that career path.

Due to a national stay-at-home rule by the Irish Government for health and safety reasons, the Irish Government cancelled the Leaving Certificate in 2020 and replaced it with an estimated-grading system. A student's grade in each subject was estimated based on their expected performance in the Leaving Certificate combined with their School's statistical profile of achievement. This profile was created based on the student's prior examination results and other related assessments to date. Following this process, all estimated results were submitted to the Irish Department of Education and Skills. When the Department received the estimated results from all Schools, a national standardisation process was applied to ensure a consistent standard in the estimated-grade process nationally.

The national standardisation process was put out to tender resulting in an external company being hired and contracted to develop the software to implement the algorithm used in this process. As part of on-going quality assurance and verification process, two separate professional bodies oversaw the development of this software. They detected four errors. These errors resulted in approximately 10,000 students being assigned a lower estimated grade in one or more subjects to what they should have received following the national standardisation process. As soon as the errors were detected, the affected students were identified and corrections made.

However, the delays in making these corrections meant that some students had not received correct offers for university places and had to wait to commence their third-level study in the following academic year.

The authors posed three questions to the students to consider and discuss the ethical implications of this case study and the impact the resulting errors had caused. These were:

1. Did it make sense to have a Canadian company develop the software to implement the national standardisation process algorithm instead of an Irish company, which would be expected to have a better understanding of the Irish Leaving Certificate process?
2. If an Irish company had made a bid to develop the software at a much higher cost than other competitors, would your group have agreed to award the contract?
3. Can you make four recommendations to avoid this issue occurring again?

The students were placed in groups to discuss and answer the three questions.

3.2 Search Engine Bias

The Google search engine is one of many services offered by Google to search for information available on the Internet. A feature offered by this search engine is to auto-complete search parameters as the user enters their search string. The auto-completion is based on a number of factors including the user's prior search history and also commonly used phrases that people have searched previously. In many cases, this auto-completion completes the search string with a high accuracy rate. However, there are instances when the auto-completion makes poor suggestions, which might seem amusing but can cause offense.

The Google search auto-completion is part of the overall search engine, which uses a page-rank algorithm to perform the search process and return a set of results. The results returned are ranked in a specific order. Using these ordered results, the search engine attempts to auto-complete the search string whilst the user is typing. Google's rationale for offering this auto-completion of the search string is to provide a more personalised search experience and increase the relevance of search results for any given user. There are factors that can influence the ranking of the search results returned and subsequent auto-completion. These include:

1. *Fee payment to Google:* By paying a fee to Google, an individual or organisation can improve the ranking of their web presence.
2. *Search engine optimization:* Specialised companies can be hired to improve the ranking of their client's web presence through a number of mechanisms such as (i) cross-linking pages on the same site, (ii) cross-linking pages on different sites, (iii) regular updates to the content of the site, (iv) adding additional metatag information to a web page that is hidden from the user but is used by the search engine when searching and ranking the page.
3. *Political and Legal factors:* It may be the case that in certain jurisdictions, political and legal factors will influence the search results and subsequent ranking.

The following questions were posed to the students on reflection of this case study:

1. Do you use any search engine other than Google?
2. Do you logout from your Google account before using the Google search engine?
3. Think of four guidelines you would give parents to teach their children to be more aware about online bias.

The students were divided into groups and tasked with discussing and answering these questions. This included their thoughts about the ethical implications of the search engine ranking algorithm and associated auto-completion. Specifically, the students needed to consider how programmers involved in the design and development of the search algorithm can have a direct influence in the ranking of returned results and the factors they must consider.

3.3 Judicial Sentencing Software

In the United States of America, courts of law in some states are employing commercial software systems to assist the judiciary in sentencing criminal defendants. There are publications freely available that have analysed these sentencing systems and in some cases, found that certain bias is being employed by the system when making a sentencing recommendation. In some cases, racial profiling and gender-bias are factors being included in the algorithm used when making a sentence recommendation to the presiding judge in the case before the court.

Essentially, the software is developed to assist the judiciary in making a sentencing recommendation especially when there are several factors to be considered in favour and against the defendant. The algorithm developed in this system should be unbiased in all scenarios but there is evidence that biased factors have been included by the system's programmers.

Similar to the previous case studies, the students were divided into groups and tasked with discussing the ethical concerns to be considered when developing the software for such software systems. They were also asked to answer the following three questions:

1. If the overall rate of crime is significantly reduced by using this kind of sentencing recommendation software, is it ethical from a deontological perspective?
2. If the overall rate of crime is significantly reduced by using this kind of sentencing recommendation software, is it ethical from a utilitarian perspective?
3. If the overall rate of crime is significantly reduced by using this kind of sentencing recommendation software, is it ethical from a virtue ethics perspective?

3.4 Automotive Accident Algorithms

In recent years, the automobile industry has seen some car manufacturers incorporating self-driving as an available feature. This feature enables the car to autonomously navigate between two geographical points without any, or minimal, intervention by the driver. The car uses an array of sensors to capture data in its environment, which is input to software controlling the car's mobility and navigation.

The software developed to enable autonomous self-driving must be capable of responding to the threat of a potential or imminent accident. Various accident algorithms need to be implemented in the

software to facilitate the car's response to such threats. When implementing these algorithms, the programmers need to be cognisant of parameters that might include legal, moral, cultural, ethical and geographical factors. Additionally, artificial intelligence (A.I.) features may also be incorporated in the software, which creates further complexity.

The students were given the following questions to discuss and report:

1. In the event of an accident occurring, who is responsible? Is it the driver of the car who engaged the self-driving feature or the programmers who developed the self-driving code?
2. Should A.I. be incorporated in the self-driving software whereby a car in self-driving mode can decide itself how to respond to the threat of an imminent accident?

4 DISCUSSION AND RESULTS

The four case studies were delivered to a group of 175 first year computer science students at Technological University of Dublin, Ireland between the 15th of April 2021 and the 30th of April 2021, as part of the computer programming module, which is assessed using a combination of continuous assessment and examinations. The content was delivered using a virtual classroom for the main lessons, with breakout rooms for the students to discuss the ethical issues from each session in smaller groups, and Padlet (<https://padlet.com/>) was used as an idea sharing space where participants could highlight their key take-aways from each lesson.

The students were surveyed with a questionnaire with seven questions using a combination of open-ended (O) and closed-ended (C) questions to evaluate the effectiveness of these sessions. The questions are presented below and were based on examples from A.N. Oppenheim's 2000 book on Questionnaire design [27]:

(C) How interesting would you rate the ethics classes? [Not at all interesting] [A little bit interesting] [Fairly interesting] [Very Interesting] [Really, really interesting]

(O) In what way(s) were the ethics classes interesting (if they were)?

(C) Did the ethics classes encourage you to look at computer issues from multiple perspectives? [Not at all] [A tiny bit] [A little bit] [Somewhat] [Totally]

(O) In what way(s) did the ethics classes encourage you to look at issues from multiple perspectives (if they did)?

(O) What, for you, were the three key takeaways from the ethics lessons?

(O) In terms of the content of this lesson, how did you find it?

(O) In terms of the format of this lesson, how did you find it?

The survey was deployed following the completion of the classes, between the 30th of April 2021 and 3rd May 2021 using Microsoft Forms. A total 25 students of the 175 that participated in the classes responded to the survey giving us a response rate of 14.29%. This response rate is relatively low, however as noted by Porter and Umbach [29], student response rates can vary dramatically between different educational institutions, varying from as low as 14% to as high as 70%. Sheehan [32] suggests that there are a number of factors that impact the response rate, including the perceived relevance of the survey to the participants (in terms of importance and timeliness of the survey), whether the participants are notified that

a survey will be occurring, whether they are reminded to do the survey after they have received it, and finally, the number of questions involved in the survey. Additionally, it is worth noting that as a result of the COVID-19 pandemic, the participants of this survey have never met in person, and therefore, may not have developed the sense of *esprit de corps* that would typically be found in First Year classgroups, which may have also impacted on the response rate. The students were given the following key instructions: (i) the survey is voluntary, (ii) all submissions do not record the students' names, and (iii) the results will be published as part of the broader discussion on these issues.

Question 1 used a closed-ended question to ask the respondents to indicate whether or not they found the ethics content interesting and an overwhelming majority of them (23/25) indicated that they found the ethics content very interesting.

Question 2, an open-ended question, asked the students in what ways they found the content interesting. The lessons were positively received, and they allowed the students to understand the potential implications of the work once they are working in the computer industry. Indicative quotes include: *"It's interesting to know how programs that serve very important purposes that can potentially change someone's life for worse or better can be ruined by a rushed design and bad programming practices"*, and *"all topics that were discussed were really interesting and the fact it gave us an insight on how the things were related to real life situations was great to see and learn about"*. They also found it enjoyable to interact with their classmates on these topics, for example, *"really got a good discussion and got to share an opinion"*, and *"Interacting with classmates"*. Many found the content enjoyable, for example, *"They were a bit of fun, and a change of pace from the rest of the module"*.

Question 3 used a closed-ended question to ask the participants if they felt that the content presented encouraged them in future to look at issues from multiple perspectives and a large majority of them (21/25) did feel that it encouraged looking at things from multiple perspectives.

Question 4 asked in which ways did the ethics classes encourage the participants to look at issues from multiple perspectives. The respondents all agreed and it also made them consider issues from specific groups such as *"minorities and people with disabilities"* as well as *"the perspectives of the most vulnerable"*. The interaction was mentioned by the majority of participants, for example *"The group discussion helped to show different perspectives from my classmates."* and *"discussion about them and sharing of ideas helped me see things newly"*. Some noted that it helped them *"think of things that I didn't think had to be written into code"* and others noted that it *"made me think about problems which may occur in our lives"*. Most crucially, the students came to realize the importance of ethics in programming, for example *"I started seeing code as something that can change people's lives and impact many real world things"*.

Question 5 asked the participants for three key takeaways from the ethics lessons and the participants focused on some crucial issues, including that small errors can have big consequences, for example *"Even the smallest mistakes can cause serious disruption, impacting people's lives. always take a step back, think about it"* and *"Basic mistakes can be dangerous"*. They also expressed a great deal of concern over Artificial Intelligence, for example *"The ways that AI affects people's lives that we do not know about"*. Most importantly

the participants noted the importance of thinking about things before programming them (consequence scanning), for example *"It's important to always consider the outcomes of your actions, whether they're large or small"*.

Question 6 asked the participants what they thought about the content of the lessons; all of the participants referred to the content as interesting and relevant to them, *"Really interesting as it is to do with recent events and very relevant"*, and *"thought provoking"*. Many noted that the self-driving cars lesson was particularly engaging, for example *"I found it very engaging. I personally enjoyed the study on self-driving cars and the ethics behind programming cars to avoid accidents"*, and *"Very interesting, such as learning about autonomous cars"*. Some indicated that they *"wished for more ethics classes in the future"* and some mentioned that the content was *"super interesting and very well delivered by two awesome lecturers"*. Most importantly the participants found the content to be easy to understand, for example *"Really easy to follow and to understand"*, and *"I found the content easy to understand, mostly because of the relatable and modern-day examples of ethical issues"*.

Question 7 asked the participants what they thought about the format of the lessons. The students responded saying they particularly enjoyed the format of the lessons with the three parts (lesson, discussion, and share), for example *"The content delivery and format was excellent. I found the transition to the breakout rooms then the return to the min room super effective to allow us to express our opinions on the topics"*, *"It was great. I liked it as it was very balanced when the lecturer explains and presents the content and then lets the students reflect on the questions asked together, making it very engaging and enjoyable"*, and *"I enjoyed the anonymous notepad and the slides!"*. Since the classes were delivered on-line (necessitated by COVID-19 restrictions) some students found the discussion process challenging, for example *"I thought the breakout rooms were a good idea. It's easier to discuss your opinions with smaller groups than all together. Although still not everyone will be comfortable voicing their opinions, I think it's a little easier to do in person than online which will hopefully be the case for future years"* and *"Having one person from each group have to speak was sometimes a bit awkward though, especially with the online meeting format"*. Overall, the students expressed their satisfaction with the mix of teaching approaches, for example *"Slides were very clear and being able to discuss the topics in small groups was useful"*, *"Good format, got us to work together"*, and *"The class was enjoyable. Information is clear and easy to understand"*.

In summary, the students who participated in the survey felt these case studies and subsequent analysis and discussion helped them consider the consequences of different programming tasks on the end-users and, for example, it got them to question whether A.I. should be used in situations where people's lives can be affected by the outcome. They also felt these sessions on ethics gave them a new perspective on programming and they enjoyed discussing the different topics with their classmates and seeing their different perspectives. They found the topics themselves interesting, authentic, and relevant to their lives and their future careers.

In parallel with the lessons the two lecturers involved in delivering this content kept reflective diaries, and considered a few issues after each lesson, including whether or not there was a good balance between the technical and ethical issues, looking at the positives and negatives of each of the case studies, and noting any

missing content or errors in the slides. The key reflections that may be of use to other CS educators includes that when teaching in a Constructivist paradigm it is important to listen to the students' voices; to be encouraging, and to be supportive and positive. Also, to highlight the importance of consequence scanning; students have to be asked to imagine, and reflect on, the potential of both intended and unintended consequences of a software system. Most powerfully, in a constructivist teaching session, an educator is free to say the phrase "I don't know, but I'll check", and to acknowledge that the students, as *digital natives* [30], may have reflected more deeply on some of the issues, for example, the trade-offs between digital privacy and on-line services, and to praise their perspectives on those issues. Other key take-aways from this study include the observation that the cases that most appealed to the students were those that were tangible and nuanced, as opposed to hypothetical and simplistic. Also, by including videos and links to newspaper articles and blogs, the students felt the content was relevant and timely. Both the lecturers and the students noted that many of the ethical programming issues could have been prevented with better software testing – an issue that was discussed and reflected on by them in detail. Finally, it is evident that all of the students who participated in the survey enjoyed the cases, and wished for more time for discussion sessions in general in all of their classes.

5 LIMITATIONS

The goal of this research is to explore the impact of topic-focused case studies on first year computer science students' understanding of some of the ethical issues associated with software development. There are several limitations to this study, firstly, the duration of the study was from 15th of April 2021 to 3rd May 2021, a duration of 18 days (necessitated by the academic calendar of the institution in which this study took place), which is a limited timespan for this type of study. A follow-up study with the same students in the next academic year would be useful to identify what they can recall from the case studies and what lessons they have learned from them. Secondly, the relatively low response rate of 14.29% makes it difficult to draw any general conclusions from this study, however, this low response rate was related to the limitations of the students' schedule, and follow-up studies can be scheduled more effectively. Finally, the use of a questionnaire as the sole means of data collection may have limited the type of student who was willing to participate in the study, and may have limited the types of answers that the students gave. A complimentary focus group session may have provided a different perspective on the study, and will be used in future iterations of this research.

6 CONCLUSIONS

This research outlined the design and delivery of four case studies focused on programming ethics as part of a first-year computer programming module, and this was followed by a qualitative evaluation process. The goal of this process is to raise awareness of the potential ethical conflicts that students may encounter once they graduate college and begin their professional lives. Crucially, there is no intention in the lessons to tell the students what the "right answer" is in terms of any ethical dilemmas but rather by having students discuss they topics themselves, in combination

with a description of different ethical perspectives (deontology, utilitarianism, and virtue ethics), they were scaffolded in developing their own views on the content. This mode of teaching represented a shift of teaching paradigm from Behaviourist approach to a more Constructivist approach, which was supported by the three-part lesson structure (lesson, discussion, and share). Based on the feedback from the students who participated in survey we can tentatively conclude that the students found the lessons extremely enjoyable and engaging and they felt the break-out rooms were essential in allowing them to see the content through other people's eyes, which was further enhanced by the Padlet activities. Some students found conversing on-line to be somewhat cumbersome, nonetheless, all of the students who participated in the survey remarked that these sessions were a catalyst for them to consider more thoroughly the potential ethical consequences of the software they develop, and to scan for the potential (intended and unintended) consequences of their decisions on real people.

ACKNOWLEDGMENTS

The authors of this paper and the participants of the Ethics4EU project gratefully acknowledge the support of the Erasmus+ programme of the European Union. The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

REFERENCES

- [1] ACM (1992). ACM code of ethics and professional conduct. Code of Ethics. <https://ethics.acm.org/code-of-ethics/previous-versions/1992-acm-code/>
- [2] Alvarez, E. (2015) "To keep a Boeing Dreamliner flying, reboot once every 248 days", <https://au.finance.yahoo.com/news/2015-05-01-boeing-787-dreamliner-software-bug.html>, last accessed 2021/06/06.
- [3] Arshad, A., Anderson, B., & Sharif, A. (2019). Comparison of organ donation and transplantation rates between opt-out and opt-in systems. *Kidney International*, 95(6), 1453-1460.
- [4] Ben-Ari, M. (2001). Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching*, 20(1), pp. 45-73.
- [5] British Broadcasting Corporation (BBC) News (2014) "Roughlee couple sent £53m electricity bill" <https://www.bbc.com/news/uk-england-lancashire-28444589>, last accessed 2021/06/06.
- [6] British Educational Research Association (2004) Revised Ethical Guidelines for Educational Research, BERA: Macclesfield.
- [7] Bynum, T. W. (2001). Computer ethics: Its birth and its future. *Ethics and Information Technology*, 3(2), 109-112.
- [8] Chevalier, J.M. and Buckles, D.J. (2013) *Participatory Action Research: Theory and Methods for Engaged Inquiry*, Routledge UK.
- [9] Davies, R. (2021) "Post Office's aggressive pursuit of staff casts shadow over ex-boss's tenure", <https://www.theguardian.com/business/2021/apr/23/post-office-aggressive-pursuit-of-staff-ex-boss-paula-vennells>, last accessed 2021/06/06.
- [10] Doteveryone, Consequence Scanning, An Agile Practice for Responsible Innovators, <https://www.doteveryone.org.uk/project/consequence-scanning/>, last accessed 2021/06/06.
- [11] Dyer, B., Hughson, M.A., Duns, J., Ricketson, S. "Teaching Note: Creating a Corporations Law Case Study", *Legal Education Review*, 8 (1997)
- [12] Ethics4EU, Erasmus+, www.Ethics4EU.eu, last accessed 2021/06/06.
- [13] Ethics4EU Intellectual Output 2 Report, Existing Competencies in the Teaching of Ethics in Computer Science Faculties, <http://ethics4eu.eu/outcomes/existing-competencies-in-the-teaching-of-ethics-in-computer-science-faculties-research-report/>
- [14] Fiesler, C., Friske, M., Garrett, N., Muzny, F., Smith, J. J., & Zietz, J. (2021, March). Integrating Ethics into Introductory Programming Classes. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (pp. 1027-1033).
- [15] Flanagan, M., Howe, D. C., & Nissenbaum, H. (2008). Embodying values in technology: Theory and practice. *Information Technology and Moral Philosophy*, 322.

- [16] Gotterbarn, D., Wolf, M. J., Flick, C., & Miller, K. (2018). THINKING PROFESSIONALLY The continual evolution of interest in computing ethics. *ACM Inroads*, 9(2), 10-12.
- [17] Gotterbarn, D. (1991). Computer ethics: Responsibility regained. *National Forum*, 71(3), 26.
- [18] Holland, T., Sherman, S. B., & Harris, S. (2018). Paired teaching: A professional development model for adopting evidence-based practices. *College teaching*, 66(3), 148-157.
- [19] Hursthouse, R., & Crisp, R. (2013). Normative virtue ethics. *Ethica*, 645.
- [20] Johnson, D. (1985). *Computer ethics*. Englewood Cliffs (NJ).
- [21] Jung, J. C., & Sharon, E. (2019). The Volkswagen emissions scandal and its aftermath. *Global Business and Organizational Excellence*, 38(4), 6-15.
- [22] Lieberman, H., Fry, C. (2001) "Will software ever work?", *Communications of the ACM*, 44(3), pp. 122-124.
- [23] Marshall, P. "The Intersection of Law, Ethics and Politics" <https://www.postofficetrial.com/2021/06/marshall-spells-it-out-speech-to.html>, last accessed 2021/06/06.
- [24] Martin DA., Conlon E., Bowe B. A Constructivist Approach to the use of Case Studies in teaching Engineering Ethics. In: Auer M., Guralnick D., Simionics I. (eds) *Teaching and Learning in a Digital World*. ICL 2018. *Advances in Intelligent Systems and Computing*, vol 715. Springer.
- [25] Mishra, P., Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers college record*, 108(6), pp. 1017-1054.
- [26] Moor, J. H. (1985). What is computer ethics? *Metaphilosophy*, 16(4), 266-275.
- [27] Oppenheim, A. N. (2000). *Questionnaire design, interviewing and attitude measurement*. Bloomsbury Publishing.
- [28] Pearl, J., "Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison.
- [29] Porter, S. R., & Umbach, P. D. (2006). Student survey response rates across institutions: Why do they vary?. *Research in Higher education*, 47(2), 229-247.
- [30] Prensky, M., & Berry, B. D. (2001). Do they really think differently. *On the horizon*, 9(6), 1-9.
- [31] Schunk, D. H. "Learning Theories." Printice Hall Inc. (1996).
- [32] Sheehan, K. B. (2001). E-mail survey response rates: A review. *Journal of computer-mediated communication*, 6(2), JCMC621.
- [33] Slabodkin, G. (1998). "Software glitches leave Navy Smart Ship dead in the water". *Government Computer News*. <https://gcn.com/Articles/1998/07/13/Software-glitches-leave-Navy-Smart-Ship-dead-in-the-water.aspx/>, last accessed 2021/06/06.
- [34] Spradling, C., Soh, L.K., Ansoorge, C. Ethics Training and Decision-Making: Do Computer Science Programs Need Help?. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, pp. 153-157, 2008.
- [35] Stafford, P. "Businesses hit by Bank of Queensland EFTPOS bug" <https://web.archive.org/web/20140407101220/http://www.smartcompany.com.au/growth/economy/12681-20100104-businesses-hit-by-bank-of-queensland-eftpos-bug.html>, last accessed 2021/06/06.
- [36] Thimbleby, H. (2012). "Heedless programming: ignoring detectable error is a widespread hazard". *Software: Practice and Experience*, 42(11), pp. 1393-1407.
- [37] Widya Yudha, S., Tjahjono, B., Kolios, A. (2018). A PESTLE policy mapping and stakeholder analysis of Indonesia's fossil fuel energy industry. *Energies*, 11(5), p. 1272.