18

ETHICAL ISSUES IN SOFTWARE ENGINEERING

Stephanie J. Lunn, Isis Hazewindus, Prajish Prasad, and Vivek Ramachandran

Introduction

From the first home computers in the 1970s to the use of algorithms in application engineering, technology has expansively progressed over the last several decades (O'Regan & O'Regan, 2008). The resulting hardware, software, and data generated have become ubiquitous in our lives, permeating numerous industries such as healthcare, finance, agriculture, manufacturing, and education. Creating and utilizing these technologies can wield tremendous influence, offering solutions across domains and society at large. Meanwhile, the conversation regarding ethical issues and implications surrounding these technologies expands.

Despite early twentieth-century critics of modern technology (Heidegger, 1977; Ellul, 1964; Mumford, 1934, 1967) warning about humans becoming slaves to technology, the prevailing view on technology has remained primarily utopian. Initially, technology and its artifacts were considered neutral, holding no intrinsic morality, with the user assessing its ethical aspects. Such approaches shift moral judgment from creators to users, subject to their end goals, thereby overlooking any underlying ethical concerns in the technologies themselves. Consequently, computers and software are perceived as objective entities, neglecting the influence of the software engineers behind the technologies, who bring their own viewpoints, beliefs, and biases into the products they and their teams create.

Nevertheless, the extensive integration of computing into society, the rise of 'Big Data,' and the impacts of created tools have necessitated questioning technology's purported neutral status. Although technology can enhance the quality of living, it can be harmful as well. Various scandals concerning algorithmic applications, data misuse, and security breaches make it increasingly evident that technology is dependent on the (conscious or subconscious) biases and decisions of the people and organizations building and maintaining them (Brown et al., 2021; Hysa et al., 2023). Computing solutions are applied to a range of topics and can enable access and dissemination of useful information, but can also exacerbate inequity, amplify stereotypes, and/or lead to issues around privacy and security – for example, through hacking into personal accounts (Xu & Tang, 2020). Thus, the influence of humans and the development and use of technology are inextricably linked, and decisions can have consequences not linked to the designers' intent.

DOI: 10.4324/9781003464259-22

To help software engineers become aware of their responsibilities in the creation, dissemination, and maintenance of software, as well as their consideration of data use, educators must provide environments that promote not only learning how to code, but also how to recognize ethical issues and accurately respond to them. This is where ethics education can play a crucial role; scholars have emphasized the need to focus on *how* rather than *what* to think (Horton et al., 2022). Integrating ethics and encouraging social justice early on can promote students' agency and encourage them to consider the implications of decisions (Moore, 2020; Ferreira & Vardi, 2021). It is critical for ethics to be conceptualized as part of practice, and to transition disciplinary mindsets away from 'avoiding traps' toward anticipating 'sociotechnical risks' (Andrus et al., 2020, p. 77).

Incorporating ethics into education is not novel but constitutes an ongoing effort. Numerous organizations, governmental funding agencies, and institutions of higher learning have previously acknowledged the need for concentrated and comprehensive ethics education. A joint task force comprised of representatives from the Association for Computing Machinery (ACM) and Institute of Electrical and Electronics Engineers (IEEE) – two major international computing societies – worked together to define a guiding 'Software Engineering Code of Ethics' to encourage standards for teaching and practice (Gotterbarn et al., 1997, 1999). Although this code can provide a beneficial foundation for framing ethical issues, decision-making in the real world can include any number of variables and requires all individuals to take ownership and responsibility for choices. Towards this goal, 'IEEE 7000' is the first standard that describes principles surrounding unintended risks and encourages 'responsible innovation' with consideration of human and social values (Spiekermann, 2021). Recognizing the importance of personal responsibility in software design and use, IEEE 7000 centers around combining material value ethics and moral philosophy to drive value-based engineering (Spiekermann & Winkler, 2022).

Despite such endeavors, attempts to foster ethical mindsets are not always widespread (Connolly, 2020). Ethics are often taught either as standalone courses or are incorporated within existing courses by engineering instructors (Fiesler et al., 2020). Yet, integrating ethical awareness and decision-making cannot be undertaken in isolation; ethics instruction must intertwine with technical understanding (Martin et al., 1996). Presenting ethical concepts sporadically may be insufficient to alter students' thinking. Instead, a widespread approach is required, with lessons embedded throughout curricula and recurring periodically throughout a student's matriculation.

As societies evolve and along with them the definitions of technology, the field of computing likewise advances, and dynamic subfields – such as artificial intelligence (AI), computer vision (CV), human-–computer interaction (HCI), and machine learning (ML) – emerge and expand. Continually considering the ways in which technology can impact accountability and transparency is essential. Moreover, we must reflect on how ethics can be applied to advocate for social justice and encourage diverse voices (Cheong et al., 2021; Ferreira & Vardi, 2021). With this chapter, we hope to contribute to a long-term and sustained approach to molding ethically minded software engineers by providing didactic tools to establish educational and professional environments where inclusion and morality are at the forefront of the discussion. We encourage conversations on and integration of ethics throughout computing curricula, touching upon themes like responsibility and social impact.

In the section that follows, we will introduce ourselves as authors of this chapter and describe our positionality to frame the chapter. Then, we will then elaborate on ethics and its place in software engineering (SE), before diving into practical approaches to integrate ethics in SE pedagogy. Finally, we outline case studies that instructors can use to convey important ethical issues in SE. Our tools can be applied within different courses with distinct objectives and assignments, with reinforcement of ethical thinking as a final goal, while providing the opportunity for students to practice ethics and empower them to "effectively actualize their (own) values in future professional decision making" (Cohen et al., 2021, p. 861).

Positionality

Given the role that personal values, experiences, and backgrounds can have in how topics are approached, interpreted, and defined (Martin et al., 2022), we want to be transparent as authors. We aim to disclose how we have influenced what is presented. In this section, we share information about who we are and the role that plays in the perspectives and approaches we use.

The first author, Stephanie, identifies as a white woman with a computer science (CS), computing education, and engineering education background. She is an assistant professor at a large, research-focused university in the southeastern United States. Her experiences with sexism and imposter phenomena in computing have driven her efforts to aid in students' technical and professional development and to cultivate more inclusive mindsets.

The second author, Isis, identifies as a white cis woman with a background in philosophy, data ethics, and the development of educational programs. Her role as a data ethics consultant and educator shaped her view on careful decision-making and ethics as a practice of continuous reflection and conscientiousness. Being the first in her family to receive training at university, she is conscious of the importance of introducing matters like ethics in ways explicitly tailored to the target group – to encourage enthusiasm and curiosity rather than alienation or indifference.

The third author, Prajish, identifies as an Indian man with a CS and educational technology background. He has experience working for an EdTech company as a software developer. He works as an assistant professor at a liberal arts university in India and teaches CS subjects like SE and introductory programming. His experiences in the software industry, as well as his research experience in educational technology, have made him realize that a more holistic instructional approach is required to help students engage with engineering problems equitably and effectively.

The fourth author, Vivek, is a non-binary Indian with a robotics, mechanical engineering, and engineering education background. His experiences in learning and teaching in Asia, Europe, and North America – where he was at times a member of the ethnic and/or racial minority – have shaped his worldview. His desire to see ethics and societal responsibility being prioritized in research and teaching in robotics and engineering domains drives his pedagogical practice.

Ethics and morality in software engineering

In philosophy, "if we are interested … in what our guiding ideals should be, in what sort of life is worth living, in how we should treat one another" (Shafer-Landau, 2018, p. 1), ethics is the domain that we turn to. This domain is a broad and can be categorized into three subdomains under which different types of looking at ethics can be grouped (Shafer-Landau, 2018):

1. *Normative ethics*: focuses on standards and what we ought to do. Irrespective of the circumstances, it considers *What are our moral duties?* and *Which actions are morally right, and which actions are morally wrong?* Normative ethics encompasses approaches like utilitarianism, consequentialism, deontology, feminist ethics, and virtue ethics.
2. *Meta-ethics*: focuses on the status of our moral claims and values. *Can we say moral theories or statements are 'true'?* We might even wonder about the existence of something like a universal moral viewpoint.

3. *Value theory*: focuses on questions concerning the nature of defining goodness and the good life: *What is a 'good life' and how do we pursue it?* It balances moral and natural goods.

More elaborate discussion of these ethical frameworks is provided in Chapter 2 of this handbook. Here, in this chapter, when we speak of ethics, we refer to normative ethics. We want to provide educators and students with the tools to help them grasp ethics and reflect on what is good in terms of actions and decision-making while keeping in mind that defining what is 'good' is complicated. Issues can be highly complex, contextual, and subjective, involving multiple actors with their own values, expectations, and worldviews. As such, we need a way to consider those issues cautiously and form as broad of a perspective as possible. With ethics, there may not be a simple solution offering clear-cut answers; rather, an ongoing process can guide and navigate conversations about these complex issues.

Cultivating ethical mindsets throughout the software development and data life cycles

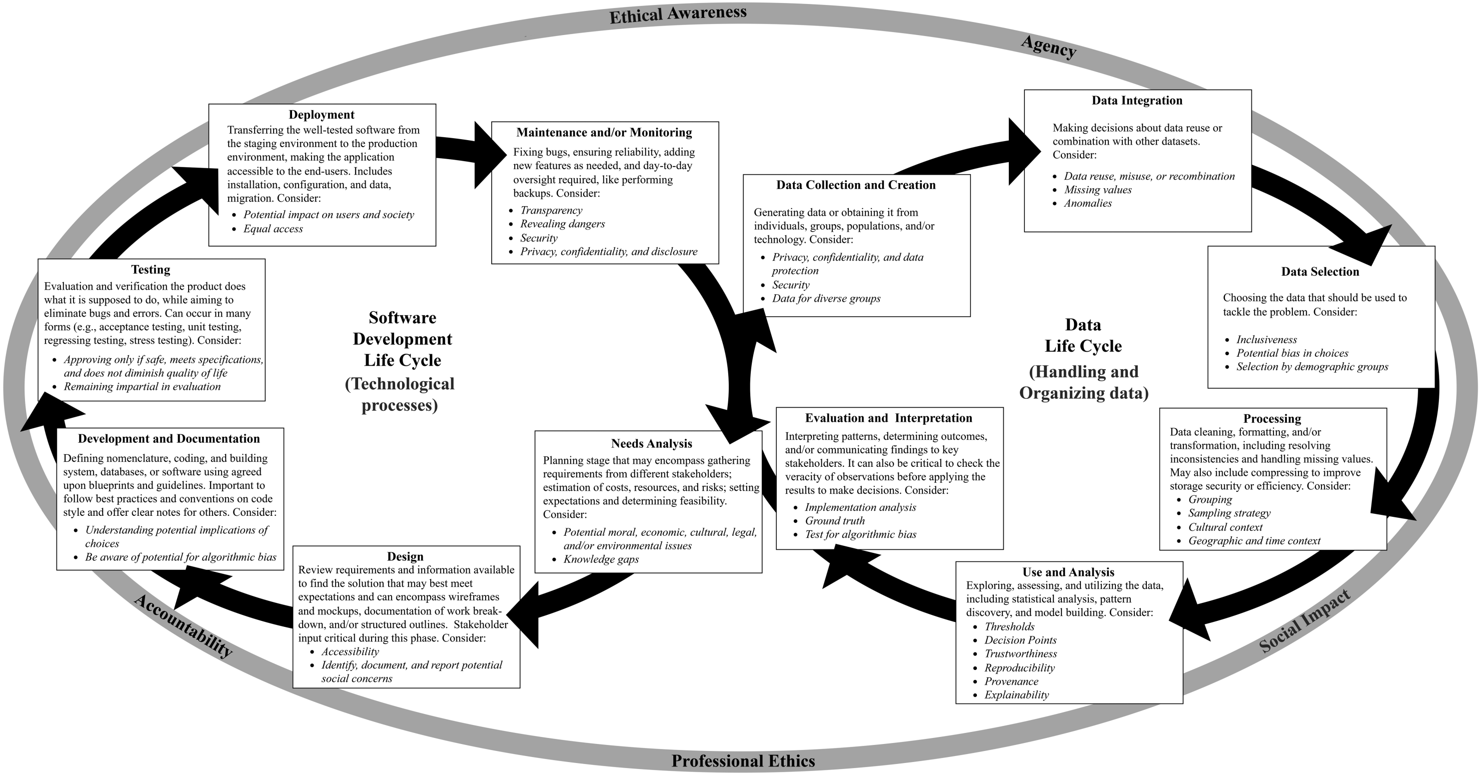
We strive to instil moral values in computing students to empower them to make ethical decisions throughout the software development life cycle (SDLC) and the data life cycle (DLC). The term 'software development life cycle' describes the formal or informal process, or methodology, employed for the design, creation, and maintenance of software (Sommerville, 2016). It has been defined using multiple models (e.g., Waterfall, Spiral) (Ruparelia, 2010; Mall, 2018), and while the names for each of the phases and specifics vary, the core concepts remain more or less fixed. Linked to software design, the 'data life cycle' comprises distinct phases (Wing, 2019), and the term describes the process of generating, using, and managing data (e.g., datasets, databases, code). Once the data life cycle is complete, knowledge gained can result in feedback that can inform/ improve additional data collection and creation. Throughout the SDLC and DLC, an 'Ethics by Design' approach can be employed to emphasize fairness, transparency, accountability, social and environmental well-being, privacy, data governance, and human agency (Dainow & Brey, 2021).

Teaching students to embrace 'Ethics by Design' involves tools, discussions, and assignments that promote ethical reflection and can aid in problem-solving. The approach can enhance choice consideration and inform planning, thinking, and actions in the life cycles. In Figure 18.1, we provide a conceptual model that unites the SDLC, DLC, and ethics (Ashok et al., 2022; Janeja, 2019; Karim et al., 2017; Wing, 2019).

Although each life cycle is presented separately, the arrows leading into each other are meant to emphasize that the two cycles are frequently interrelated. Data generation can follow software development, and data results can spark further software development. Ethical principles must be stressed across the process. We propose this new conceptualization, which we label the ETHOS model, as a way for educators to reconcile the components it encompasses:

Ethical (ethics)
Technological processes (software engineering life cycle),
Handling, and
Organizing data (data life cycle) for
Social impact (ethics)

These components appear in Figure 18.1 to make their connectivity explicit and highlight that consideration of ethics should encapsulate all decisions, choices, and actions taken – and all with


Ethical Awareness
Agency
Social Impact
Professional Ethics
Accountability
Software Development Life Cycle (Technological processes)
Data Life Cycle (Handling and Organizing data)

Needs Analysis
Planning stage that may encompass gathering requirements from different stakeholders; estimation of costs, resources, and risks; setting expectations and determining feasibility. Consider:
- *Potential moral, economic, cultural, legal, and/or environmental issues*
- *Knowledge gaps*

Design
Review requirements and information available to find the solution that may best meet expectations and can encompass wireframes and mockups, documentation of work break-down, and/or structured outlines. Stakeholder input critical during this phase. Consider:
- *Accessibility*
- *Identify, document, and report potential social concerns*

Development and Documentation
Defining nomenclature, coding, and building system, databases, or software using agreed upon blueprints and guidelines. Important to follow best practices and conventions on code style and offer clear notes for others. Consider:
- *Understanding potential implications of choices*
- *Be aware of potential for algorithmic bias*

Testing
Evaluation and verification the product does what it is supposed to do, while aiming to eliminate bugs and errors. Can occur in many forms (e.g., acceptance testing, unit testing, regressing testing, stress testing). Consider:
- *Approving only if safe, meets specifications, and does not diminish quality of life*
- *Remaining impartial in evaluation*

Deployment
Transferring the well-tested software from the staging environment to the production environment, making the application accessible to the end-users. Includes installation, configuration, and data, migration. Consider:
- *Potential impact on users and society*
- *Equal access*

Maintenance and/or Monitoring
Fixing bugs, ensuring reliability, adding new features as needed, and day-to-day oversight required, like performing backups. Consider:
- *Transparency*
- *Revealing dangers*
- *Security*
- *Privacy, confidentiality, and disclosure*

Data Collection and Creation
Generating data or obtaining it from individuals, groups, populations, and/or technology. Consider:
- *Privacy, confidentiality, and data protection*
- *Security*
- *Data for diverse groups*

Data Integration
Making decisions about data reuse or combination with other datasets. Consider:
- *Data reuse, misuse, or recombination*
- *Missing values*
- *Anomalies*

Data Selection
Choosing the data that should be used to tackle the problem. Consider:
- *Inclusiveness*
- *Potential bias in choices*
- *Selection by demographic groups*

Processing
Data cleaning, formatting, and/or transformation, including resolving inconsistencies and handling missing values. May also include compressing to improve storage security or efficiency. Consider:
- *Grouping*
- *Sampling strategy*
- *Cultural context*
- *Geographic and time context*

Use and Analysis
Exploring, assessing, and utilizing the data, including statistical analysis, pattern discovery, and model building. Consider:
- *Thresholds*
- *Decision Points*
- *Trustworthiness*
- *Reproducibility*
- *Provenance*
- *Explainability*

Evaluation and Interpretation
Interpreting patterns, determining outcomes, and/or communicating findings to key stakeholders. It can also be critical to check the veracity of observations before applying the results to make decisions. Consider:
- *Implementation analysis*
- *Ground truth*
- *Test for algorithmic bias*

Figure 18.1 ETHOS conceptual model applying ethics throughout the SDLC and DLC (drawing from Ashok et al., 2022; Janeja, 2019; Karim et al., 2017; Wing, 2019).

regard for social impact. An ‘Ethics by Design’ perspective, as described by the ETHOS model, highlights the value of being mindful of and addressing ethical issues as early as possible. It also lends itself to following up on these issues and evaluating them throughout the process. As identified in the outer ring of this model, this necessitates consideration of additional concepts, including:

1. *Accountability and transparency*: To address the challenge posed by the ‘many-hands problem’ of attributing outcomes to a single individual in software projects that involve multiple people (van de Poel, 2015), the ETHOS model fosters a culture of individual ownership for actions, choices, and mistakes. In addition to complying with regulations, this involves collaboration, understanding expectations, ensuring accuracy, considering consequences, valuing quality, and being honest about decisions. Transparency involves communication, traceability (documenting or tracking goals, history, and assumptions), and intelligibility (comprehensibility and monitoring programs and systems to achieve outcomes).
2. *Agency*: Computing professionals make ethical decisions shaping the creation, maintenance, and application of products, programs, and data. They actively influence the ethical landscape through decision-making rather than passively following established norms. Thus, empowering individuals to address possible issues and propose solutions as they arise is valuable. Those involved ought to have autonomy over ethical deliberations, strategies, and actions that support the freedom and diversity of users. They should also reconcile differing interests and values while balancing project and product specifications.
3. *Ethical awareness*: Recognizing, analyzing, and navigating moral situations is crucial for individuals and organizations as it adds a human element of control. This goes beyond understanding the impact on various actors, including the environment. Hence, it is essential to prioritize the sustainability of products, programs, or data and anticipate potential repercussions at the conclusion of the item’s life cycle. Proactively establishing safeguards in the SDLC and DLC to detect and mitigate problems is part of this, but ongoing appraisal and reflection are equally important for mindful decision-making.
4. *Professional ethics*: Conduct standards should be an integral part of the SDLC and the DLC. Individuals and organizations must uphold an ethical code – and values – centered on respect, transparency, honesty, fairness, privacy, and human-centric well-being. Individuals should understand diverse definitions and manifestations of these concepts, enabling them to make ethical decisions in various situations. Such ethical conduct may appear through fidelity, beneficence (i.e., promoting doing good or the well-being of others), non-maleficence (i.e., avoiding harming or injuring others), and justice (i.e., distributing benefits and burdens fairly).
5. *Social impact*: Ethical lapses in the SDLC and DLC can have serious consequences in real-world situations. Accordingly, it is important to extend beyond the technical aspects of software engineering to consider social, economic, cultural, and environmental impacts on humanity. Such thinking and behaviors demand global awareness, cultural competence, attention to accessibility, and the inclusion of diverse perspectives – encouraging proactive contributions to technology in alignment with varying ethical values. It necessitates addressing societal challenges, promoting social good, and identifying, seeking to prevent, and resolving potential negative impacts of programs, applications, and/or data.

We advise using the ETHOS model to emphasize human values and students’ personal responsibility and encourage students to reflect on their own roles, biases, and decisions. Although we present bullet points in Figure 18.1 to provide instructors with ideas for further discussion points, a simplified version may make the content more digestible in the classroom. Introducing individuals to existing codes like the ACM's Code of Ethics or having students create their own codes fosters awareness of their professional responsibilities. Everyone taking moral responsibility and striving for continuous improvement and just behaviors can contribute to a positive shift in our interdependent society.

The following sections describe several ways educators can approach these topics. We encourage the exploration of different (ethical) viewpoints. Instructors can initiate meaningful conversations in classrooms using several 'dialogical approaches.' Students can also engage with various 'reflection tools' to help them ruminate on ethical issues and concerns in projects. These dialogue approaches and reflection tools can be considered to further students' engagement with ethical concepts, issues, and thought processes. The sections below focus on software engineering, and, again, for a more general discussion on using reflective and dialogical approaches in engineering ethics education, we refer readers to Chapter 25.

Dialogical approaches

Reflecting on choices and engaging in open discussions about intercultural responsibility, tolerance, and consequences aid ethical decision-making (Ferri, 2014). Integrating ethics into the classroom can be approached in various ways to help students develop ethical mindsets and address any issue they may encounter.

The purpose of ethical conversations in software engineering

Issues can arise in any project or system involving software or data, and three key aspects must be considered (van Veen & Visser-Knijff, 2022). The first involves the project's technological possibilities and feasibility, leading back to the SDLC. The second regards the legality of data storage and application (i.e., the DLC): *Is data's intended use and processing compliant with regulations?* These two aspects can easily be solved by personnel like privacy officers and can be assisted with a control check like a data protection impact assessment (DPIA).[1] However, even when all technical or legal questions are sufficiently answered in any project, a third aspect remains – concerning desirability. This aspect demands careful reflection regarding software and data projects at different points in time to ensure we consider future ethical implications a project might have in addition to complying with the rule.

One way to empower students is by incorporating ethical conversations into educational practice. Gaining experience in having ethical conversations about real-world case studies can help students gain what Aristotle calls *phrónēsis*, or practical wisdom, which to him means knowing how to act in a specific situation (Kraught, 2022). Gaining *phrónēsis* is a process of training to cultivate one's virtues or good traits. By applying principles of action to real-world situations, we will train ourselves to assess each situation carefully and then decide on our actions, which should align with a virtue. This way of looking at ethics is crucial because 'doing' ethics is not a matter of ticking boxes or choosing the right action from a predefined set, but rather requires us to assess each situation and its context on its own and decide our best course of action from there.

In addition to teaching students about ethical implications, conversations can aid decision-making. Facing ethical issues can potentially pressure students to make the right decision. At the same time, many students lack formal decision-making training and practice (Bond et al., 2008; Keeney, 2020) while also underestimating how their own biases affect decision quality. Discussing projects in a group may reveal diverse perspectives, offering invaluable input on project decisions.

Conversations may yield new insights, promoting ethically minded decision-making and risk mitigation. Below, we will summarize three conversation methods and how they can be used:

1. *Socratic dialogue*: The Socratic dialogue is a conversation method which, according to Knezic et al. (2010, p. 1105), is defined as:

> a philosophical group dialogue in which the participants guided by a facilitator and a number of ground rules strive to reach a consensus in answering a fundamental question on the basis of a real-life example or incident with the purpose of achieving new insights.

For those in computing roles, the Socratic dialogue may appeal for cultivating several skills:

a) Asking the right questions – as the dialogue is centered around one core question, students are pushed to carefully consider what questions should be asked in different stages of the ETHOS model.
b) Inquiring into underlying assumptions or ideas – at its core, the Socratic dialogue is focused on inquiring into one's ideas, beliefs, assumptions, and dispositions. It is thus pre-eminently suited to encourage students to examine their own attitudes and identify any bias they might unconsciously hold that could impact projects.
c) Promoting mutual understanding using the Socratic dialogue is a collective effort, pushing participants to reach a unanimous conclusion. This encourages students to adapt their viewpoints based on new information and sound arguments, fostering better teamwork and openness to diverse perspectives that can alter project directions.

2. *Moral case deliberation*: Moral case deliberation is a conversation method commonly used in medical and business ethics. It involves discussing a real-life case to move towards a concrete approach or action.

   As moral case deliberation aims to obtain concrete project actions, it proves valuable with case studies or as group intervention for students dealing with ethical challenges in internships. Discussing problems and extracting actionable suggestions boosts students' confidence to address such issues in projects, enhancing their ethical agency. As per the description in a 2012 article by Karssing (2012), students will also be encouraged to explore the case and thus increase their ethical awareness and sense of personal responsibility.
3. *Thought experiments and scenario thinking*: The last method to initiate a conversation is to use thought experiments or scenario thinking. Thought experiments (e.g., Philippa Foot's 'Trolley Problem') are widely used in philosophy. Philosophers often use thought experiments to visualize scenarios that cannot be carried out in the real world due to physical, financial, technological, or ethical limitations. The *Stanford Encyclopedia of Philosophy* (Brown & Fehige, 2019) describes engaging in a thought experiment as follows: "we visualize some situation that we have set up in the imagination; we let it run or we carry out an operation; we see what happens; finally, we draw a conclusion."

Using thought experiments and scenario thinking lends itself especially well to software engineering, as it focuses on trying to gain insight by:

a) Identifying issues by describing possible outcomes of deployment (by sketching utopian and dystopian scenarios).
b) Reflecting on the nature of the problem that serves as the basis for a project (by asking: *Is this solution the means or the goal?*).

c) Reflecting on the necessity of using specific data-driven or technological solutions and exploring alternative options (by asking: *What problem(s) does this solution solve? What problem(s) doesn't it solve? What new problem(s) does it create?*).

Reflection tools

Ethical reflection tools can provide further guidelines and resources to help identify and mitigate potential ethical issues that may arise. Although the ETHOS model can function as one such reflection tool, others have been developed which may offer additional resources and guidance. They adopt a variety of formats, that we broadly classify as: (1) ethics-by-design tools, (2) process tools, and (3) ethical impact assessments. What they all have in common is that they offer a way to start ruminating on a project in a structured way. They can help individuals in computing make more informed decisions, consider how to address ethical issues in their projects, and weigh the potential benefits and harms of different options. Some tools that may be helpful to consider are:

- The *Data Ethics Canvas*[2] is a reflection tool created to help identify and manage ethical issues in the DLC. Designed for project-wide use, it poses predefined questions to provide an overview of problems encountered in data-based projects. It can help acquaint students with using ethical tools. Alongside the canvas (i.e., matrix), the tool offers a comprehensive user guide that promotes using the tool for enhanced project planning and encourages a human-centered perspective towards the design from the project's outset.
- *Consequence scanning*[3] involves a set of activities that can be incorporated into the SDLC, to examine the potential impact on people and/or society. It consists of a manual, printed headings, a proposed structure to follow, and two sets of prompts focused on consequence and context. It provides opportunities for software teams to have conversations about the potential implications of what they have built, what they will build, and how they can mitigate potential harms before they happen. It can also help students cultivate awareness of ethical and societal problems arising when one fails to reflect on potential consequences of their designs.
- The *Data Ethics Decision Aid* (DEDA)[4] is a toolkit designed to be used by teams involved in data-based projects. With a worksheet designed like a board game and with a questionnaire and handbook, the DEDA offers an interactive experience. Using the DEDA is a structured process in which each team member has their place, with the worksheet serving as the foundation of a conversational session. Applying the DEDA in educational settings can help students gain experience with identifying values *and* value conflicts, and approaching software products using an ethics-by-design approach, while also learning to work as a team.
- The *Ethical OS Toolkit*[5] serves as a guide for predicting the future consequences of current technology. It aims to encourage various stakeholders involved in its creation, dissemination, and use to foresee potential issues. The tool includes 14 scenarios to start conversations, a risk mitigation manual, and a checklist for considering ethical questions and issues. Educators can use the toolkit in computing or design courses in multiple ways, for example,having students read about the proposed 'Risk Zones' and tasking them with collecting examples in the real world.

Case studies

Using case studies to promote ethical thinking

As educators, facilitating critical thinking via case studies can help students understand moral issues in real-world situations and prompt them to connect ethics to their practice. Analyzing case studies can stimulate ethical awareness and a sense of personal and professional responsibility. Such activity can help students explore the values and interests of the various actors (Lim et al., 2011). Working with case studies can serve as problem-based learning, which may aid decision-making (Chowdhury, 2018) and improve critical thinking. By pairing reflection tools with the cases, teachers can provide students with further opportunities to internalize concepts and meta-cognition (Begley & Stefkovich, 2007). For specific advice on facilitating this type of discussion/activity, please see Chapter 20.

We note that 'flipped classrooms' can provide a viable alternative for educators concerned about finding time to integrate case studies into existing lessons. 'Flipped classrooms' is an instructional approach where traditional teaching methods are reversed, with students learning new material independently (outside of class) and engaging in activities, discussions, or case studies during class time. This approach can ensure material is covered while enhancing engagement and academic outcomes (Fuchs, 2021). Moreover, providing students with guidance as they solve problems (such as those described through hypothetical scenarios) can be aided via interactions with peers and/or instructors (Herreid & Schiller, 2013).

We now showcase three case studies to provide educators with possible starting points to engage students on topics related to current ethical concerns in software engineering. These case studies are hypothetical but are based on real-world examples. Although there are many ways to incorporate these cases into lessons, we encourage instructors to consider having ethical conversations (using the conversational methods described in the previous section) and providing students with reflection tools to tackle ethical issues arising from these scenarios. Alternatively, each could be integrated into small-group role-play (for more on that, see Chapter 20).

Case study 1: Algorithmic bias – fairness

Scenario

You head the Responsible AI division at SynGenAI, a software company based in Silicon Valley. SynGenAI is steadily gaining attention because it recently released Talk8ive, a free-to-access general-purpose chatbot that tries to answer users when they pose questions. Talk8ive is built on a large language model (LLM), a machine learning algorithm that can recognize, predict, and/or generate text based on large datasets of content, primarily text that the algorithm is fed as input training data. Talk8ive was released with much fanfare but within hours of its release, a number of users started flagging the chatbot for the content it was generating. Since Talk8ive is designed to answer any kind of question, all types of queries are treated the same way. Therefore, while queries with more deterministic responses (coding-related) generate less disputable answers, queries about predicting social outcomes (recidivism) produce untrue and racially biased responses. The behaviors displayed by the chatbot are attracting a lot of negative attention to the product and the company, and there are calls to revoke free access to the bot.

Discussion prompts:

1. *Did SynGenAI make the right decision to release Talk8ive? Why were the issues not flagged in the development stage itself?*
2. *Should the chatbot have been created to address all kinds of queries in the same manner? How did the algorithmic bias enter into the generated context?*
3. *As head of the Responsible AI division, how do you prioritize your responsibilities to the company, to the public, and to your team? What course of action should you adopt?*

Context:

Generative AI media, such as OpenAI's DALL-E or DALL·E2[6] and Speechify,[7] have gained popularity due to their ability to create realistic digital images or clone voices using machine learning algorithms (Chen & Lin, 2023; Pavlik, 2023). These algorithms are trained on vast amounts of data, allowing them to produce content that can be difficult to distinguish from human-created content. ChatGPT, developed by OpenAI, is one such example of a chatbot derived from the generative pretrained transformer (GPT) (Rudolph et al., 2023), that utilizes LLMs to generate responses in a conversational manner. Generative AI tools can be classified into three based on the kind of tasks they perform (Narayanan, 2019):

1. *Perception* tasks involve interpreting sensory data, such as images, audio, or text, to extract meaningful information.
2. *Judgment automation* involves making decisions or providing recommendations based on data analysis and predefined criteria.
3. *Social outcome prediction* involves predicting future events or results based on historical data and patterns.

An over-reliance and belief in generative AI's ability to solve problems can lead to complacency and additional problems. Although generative AI may positively contribute to society, it has limitations, because the underlying algorithms may be trained on datasets that reflect societal biases. If ignored, AI systems can 'learn' in ways that amplify incorrect points or replicate discriminatory patterns. For example, in areas such as hiring or loan approval, AI algorithms can reinforce existing prejudices and disadvantage certain groups, perpetuating social inequities rather than addressing them (Su & Yang, 2023). Broadly speaking, algorithmic bias can result in the following types of problems:

1. *Misrepresentation of minoritized communities*: Minoritized individuals in the tech industry often lack the influence to shape software development and data usage. Consequently, content related to these groups tends to be biased and reflective of societal stereotypes or exclusionary practices.
2. *Misinformation*: The presence of pseudoscientific content in the training data may result in its regurgitation by the AI when presented to users. Misinformation is compounded by automation bias because humans tend to trust the confident tone of interactive AI like chatbots.
3. *Automation and intellectual property theft*: Generative AI platforms are affecting the occupations of artists and journalists, whose occupations are being automated by large corporations because it is cited as being cheaper. Often, these automated generations neglect factors such as copyright or licensing in their training data – and effectively commit digital theft.

Educators can discuss algorithmic accountability using the ETHOS model to raise awareness of the ethical implications of creating and using AI. By ensuring that data sources are properly curated, algorithms are tested rigorously, and AI-generated outcomes are accounted for by developers, students will learn to develop a practice of developing ethical AI systems. Educators can use this case study in the classroom by integrating the dialogue approaches and reflection tools mentioned earlier. For example, moral case deliberation could be used to discuss algorithmic bias. Issues with unwanted output are a common problem with chatbots but often only seem to become apparent following release. Accordingly, we suggest holding a moral case deliberation on the question: *Should SynGenAI take Talk8ive offline completely?* We also suggest using the consequence scanning tool in combination with this case study to encourage reflecting on more careful implementation. The main ethical issues in this case revolve around the (unforeseen or unintended) consequences of generative AI.

Case study 2: User interface design – empowerment

Scenario

Your current phone is falling apart – it is running out of storage, it has fallen to the ground multiple times – and as a result, it runs very slowly. So, you decide to buy a new phone. Although you do not have a clear idea of what phone you want to purchase, you do have a set budget. By querying a search engine, you find a list of shopping websites selling phones within your target range. You browse some of these websites, looking up different phones and their specifications. After some time, you pick a phone offered at a 30% discount. You read the reviews and are satisfied with the performance and other aspects of the phone. You decide to wait a few days to buy it since there is no real sense of urgency to purchase it immediately. However, as you try to navigate away from this page, you encounter a pop-up message, with a limited time offer for an additional 10% discount that is valid only until the end of the day. A countdown timer is also shown, ticking down on the hours left to buy the phone at this price. Since this seems like a good deal – and you do not want to regret buying it later at a higher price – you add it to your shopping cart. As you check out, you look at the final price and realize that the total is much higher than what was shown earlier. Upon closer inspection, you find that the final breakdown includes a screen guard and a one-year phone protection plan that you did not intend to purchase, and did not add yourself.

Discussion prompts:

1. *How does this experience make you, as a user, feel? What specific design aspects of the shopping website contributed to this feeling?*
2. *How does using interface design like those mentioned above affect your ability (as a user) to make a well-informed, autonomous choice?*
3. *Do you note any ethical issues in how the shopping experience has been designed? If so, what are they and what would your feedback be for the user-interface designer?*
4. *What might the different motivations be that govern the design choices for the website user interface? How might web designers redesign the interface to improve the user experience while addressing potential ethical issues?*

Context:

The scenario describes a not-uncommon shopping experience. Many users encounter similar situations where choices and decisions are manipulated during their interactions with a given website. Moreover, these types of manipulation are not restricted to e-commerce. Companies have tried to force certain actions on users engaged in unrelated pursuits. For example, at one point, users were automatically opted into eBay's marketing emails when they signed in with their Google accounts.[8] Some companies design their user interfaces to make it difficult to perform certain actions, particularly those that can harm the company's revenue. For example, users from Europe have complained about Amazon's confusing cancellation process regarding Prime service, which tries to distract users by issuing several warnings to deter them from canceling their subscription.

Some designers use knowledge of human behavior to implement deceptive functionality that is not in the user's best interest (Gray et al., 2018) to manipulate them (Haselton et al., 2015). The design interfaces and choices described in the scenarios mentioned above are commonly known as ‘dark patterns’ (also known as ‘deceptive designs’). Dark patterns are user interface design choices that nudge, manipulate, or deceive users into making unintended and potentially harmful decisions while using an online service (Mathur et al., 2019). The key issue in the above scenarios is that these services may sacrifice providing users with a positive experience for achieving the company’s business goals. Dark patterns are asymmetric (i.e., available choices are presented differently to users), covert (i.e., the effect of the design choice is hidden from users), deceptive (i.e., they portray a misleading/false belief); and restrictive (i.e., they hide, obscure, or delay presentation of information from users) (Mathur et al., 2019). Tactics of dark patterns include sneaking (i.e., concealing or delaying information users might object to if overtly displayed), creating false urgency, misdirecting (i.e., using words or visuals to steer users), providing false ‘social proof’ (i.e., displaying fictitious user experiences to coerce a purchase), implying scarcity (i.e., falsely indicating product unavailability), causing obstruction (i.e., making it hard for users to exit situations), and forcing action (i.e., requiring tasks unrelated to current activities) (Mathur et al., 2019). Concerning the ETHOS model, such dark patterns can be associated with both the SDLC and the DLC, especially those regarding the design and development of software systems. Moreover, data from user interactions can be applied to design dark patterns, violating the professional and ethical codes of conduct that necessitate considering the well-being of the software’s end-users.

Considering the prevalence of dark patterns, students are likely to have encountered such scenarios online. Hence, drawing from students’ experiences can be an excellent way to introduce the ethics involved in user interface design. Instructors can describe examples of deceptive designs and ask students to classify designs into different categories. Researchers have suggested that interventions such as ‘spot the dark pattern’ and ‘design bright patterns’ can help create awareness regarding design patterns (Bongard-Blanchy et al., 2021) and hence can be used in classrooms. Instructors can also leverage software engineering projects to help students think of and design ethical user interfaces. Using dialogical approaches, such as the Socratic dialogue technique with a prompt like *What is autonomy?* can bring out diverse issues surrounding ethical user interfaces. DEDA can also be used as a reflection tool. Students can identify conflicting values between (a) companies that use dark patterns and (b) users confronted with dark-pattern user interfaces. As the DEDA encourages identifying (conflicting) values and actors, it can prompt helpful reflection on the implementation of dark patterns.

Although creating awareness of deceptive patterns is needed, a more crucial goal for educators is to help students integrate ethical reasoning into all aspects of the user interface design process. Elements of the ETHOS model, such as ethical awareness and accountability can prepare students to engage in critical debate about where to draw the line, based on the company’s values and their own sense of ethics. Students should realize the power that user interfaces have over people’s behavior and online interactions, and the social responsibility they as software designers have towards developing ethical user interfaces.

Case study 3: Privacy and surveillance – security

Scenario

You are a machine learning engineer in the computer vision department at Protos. This tech company manufactures virtual reality (VR) headsets and a social media space called Protospace for VR users’ avatars to meet for work and entertainment. To enhance the experience, Protos has developed a headset called VISOR with inward-oriented cameras to obtain real-time information about users’ gaze and non-verbal facial expressions. In addition, external, outward-oriented cameras emulate users’ physical body movements in the virtual world. Protos states these sensors allow for deploying customized, photo-realistic avatars and adaptive content based on the users' motion and reactions. Users must accept Protos' terms to use VISOR. Critics of the company are skeptical about this justification – they believe that Protos will store and apply the data captured by the sensors as training for future projects. Moreover, the purported immersive nature of Protospace will grant Protos access to users' behavioral patterns, which can then be monetized by manipulating advertising and/or selling the data to third-party enterprises. Furthermore, there are growing concerns about the potential use of such a technology by autocratic governments that seek to surveil their citizens to quell civic unrest, repress all forms of opposition, and establish political dominance.

Discussion prompts:

1. *What are the potential risks of collecting this type of sensitive data from users? What measures can be taken to mitigate these risks?*
2. *What should be included as part of the informed consent terms? Are there other ways to help users recognize what they are agreeing to? How can users be made aware of future uses of the data collected?*
3. *As a machine learning engineer whose work involves processing the data gathered by the cameras, how could you improve the design's privacy and/or security?*

Context:

Ongoing attempts to improve facial recognition, non-verbal expressions, and gestures have resulted in data collection beyond what users have agreed upon or consented to. The example scenario mirrors real-life concerns around Meta's Quest Pro, a device collecting multi-dimensional data (Johnson, 2022). Although the technology was described as valuable for deploying customized, photo-realistic avatars, concerns arose around the detailed information collected about individuals and how it might subsequently be used for surveillance or otherwise infringe upon their privacy. Furthermore, Meta's privacy policy was vague regarding how the data would be used and shared with outside services.

Despite multiple calls to regulate the collection, storage, and application of user data, it appears that companies frequently collect more data than necessary. Personal information, preferences, and behaviors are often captured without explicit consent, leading to a loss of control over one's own data. This can result in unintended consequences for privacy, such as data breaches, identity theft, or the misuse of sensitive information. Scholars have highlighted the need for enhanced consideration of the interactions between technology and the law (Bernes, 2022). They have encouraged those involved to consider the purposes, implications, and uses of data collected; the transparency of tools developed; and ways to limit what is gathered, recorded, and disseminated for purposes more strictly aligned with goals. The European Union's General Data Protection Regulation (GDPR) has specific guidelines on this that students should be made aware of; readers are encouraged to see Chapter 13 on law in engineering ethics education.

There are several approaches educators can take to deliver lessons on privacy and security. For instance, they can introduce concepts, such as the benefits of sharing; privacy protection and perceptions; codes of conduct; cryptographic protocols; and international laws and regulations. The ETHOS model could help align discussion around design, deployment, development and documentation, or maintenance or monitoring when referencing components of the SDLC. The scenario above can also fit into discussions about the DLC throughout its phases. Concrete examples and discussion scenarios can raise awareness of possible issues and promote the need for greater accountability. For example, students could be asked to consider the trade-offs of retail loyalty cards, which can offer discounts but may also track shoppers' habits and sell data to third parties. Another example could be related to concerns regarding healthcare and genetic mapping and/or ancestry websites. By spanning multiple fields and topics, for example, from electronic payment systems to educational records, instructors can help illustrate the necessity of awareness, protection, and action to rectify issues.

Instructors can use dialogical approaches such as Socratic dialogue and scenario thinking in conjunction with this case study. For the Socratic dialogue, we suggest evaluating the concept of ‘privacy by design’ as a realistic development approach. For scenario thinking, imagining idealistic and problematic scenarios for collecting sensitive user data may shed light on data ethics and values. The Data Ethics Canvas can be an appropriate reflection tool introduced along with this case study. One of the central issues here is the collection and use of personal user-data to further the business interests of the company doing the collection. By utilizing the Data Ethics Canvas, students can engage with questions that address multiple facets of this case, including the legal implications of collecting and utilizing data on this scale; they can reflect on reasons for using this data and the need for transparency.

Conclusions

As technology evolves, so do concepts and definitions; engineering educators must continually update the definitions, pedagogical content, and activities we use. Although today’s society is increasingly recognizing the urgent need to address ethical issues and the need for responsible software development, this recognition brings its own problems – problems that have an impact beyond the classroom. We face the very real danger of ‘ethics’ becoming a buzzword and another trend to be monetized by companies (the way corporate ‘greenwashing’ undermines sustainability initiatives, see Chapter 6). Ready-made courses or tools that seem able to solve our ethical issues rather than focus on imparting awareness and ethical thinking could exacerbate that potential problem. A criticism that has been raised regarding the Sustainable Development Goals (SDGs), which some organizations tout without achieving results in specific target areas, as mentioned in Chapter 6. Another possible concern involves the increasing reliance on tools; while ethics assessment tools can be very helpful in guiding conversations and steering us toward relevant questions, they can be used in more nefarious ways. They may be used to help companies identify or define what is easiest or cheapest rather than what is morally desirable (and desirability can also vary depending on who is asked). Using ethics assessment tools could devolve into merely another box-ticking exercise, rather than being used to prompt software developers to reflect on the possible implications of their decisions and the products they help create. Ongoing efforts must thus be made to provide meaningful examples, to articulate where potential issues may arise, and to help students make relevant and meaningful connections – to help ensure future generations of engineering practitioners are committed to ethical principles and that software engineers develop products and manage issues as equitably and justly as possible.

Notes

[1] https://gdpr.eu/data-protection-impact-assessment-template/
[2] https://www.theodi.org/article/the-data-ethics-canvas-2021/
[3] https://www.tech-transformed.com/product-development/
[4] https://dataschool.nl/en/deda/
[5] https://oecd-opsi.org/toolkits/ethical-os-toolkit/
[6] https://openai.com/dall-e-2
[7] https://speechify.com/
[8] https://twitter.com/darkpatterns/status/1470399874147438594

References

Andrus, M., Dean, S., Gilbert, T. K., Lambert, N., & Zick, T. (2020). AI development for the public interest: From abstraction traps to sociotechnical risks. In *2020 IEEE International Symposium on Technology and Society (ISTAS)* (pp. 72–79). IEEE.

Ashok, M., Madan, R., Joha, A., & Sivarajah, U. (2022). Ethical framework for artificial intelligence and digital technologies. *International Journal of Information Management, 62,* 102433.

Begley, P. T., & Stefkovich, J. (2007). Integrating values and ethics into post secondary teaching for leadership development: Principles, concepts, and strategies. *Journal of Educational Administration, 45*(4), 398–412.

Bernes, A. (2022). Enhancing transparency of data processing and data subject's rights through technical tools: The PIMS and PDS solution. In: Senigaglia, R., Irti, C., Bernes, A. (Eds.), *Privacy and Data Protection in Software Services. S*ervices and Business Process Reengineering. Springer, Singapore. https://doi.org/10.1007/978-981-16-3049-1_17

Bond, S. D., Carlson, K. A., & Keeney, R. L. (2008). Generating objectives: Can decision makers articulate what they want? *Management Science, 54*(1), 56–70.

Bongard-Blanchy, K., Rossi, A., Rivas, S., Doublet, S., Koenig, V., & Lenzini, G. (2021). "I am Definitely Manipulated, Even When I am Aware of it. It's Ridiculous!" – Dark Patterns from the End-User Perspective. In *Designing Interactive Systems Conference* (pp. 763–776).

Brown, J. R. & Fehige, Y. (2023). Thought experiments. In Zalta, E. N. & Nodelman, U. (Eds.), *The Stanford encyclopedia of philosophy*. Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/entries/thought-experiment/

Brown, S., Davidovic, J., & Hasan, A. (2021). The algorithm audit: Scoring the algorithms that score us. *Big Data & Society, 8*(1), 2053951720983865.

Chen, J. J., & Lin, J. C. (2023). Artificial intelligence as a double-edged sword: Wielding the power principles to maximize its positive effects and minimize its negative effects. *Contemporary Issues in Early Childhood*, 14639491231169813. https://doi.org/10.1177/14639491231169813

Cheong, M., Leins, K., & Coghlan, S. (2021). *Computer science communities: Who is speaking, and who is listening to the women? Using an ethics of care to promote diverse voices*. In Proceedings of the *2021 ACM Conference on Fairness, Accountability, and Transparency* (pp. 106–115).

Chowdhury, M. (2018). Emphasizing morals, values, ethics, and character education in science education and science teaching. *MOJES: Malaysian Online Journal of Educational Sciences, 4*(2), 1–16.

Cohen, L., Precel, H., Triedman, H., & Fisler, K. (2021). A new model for weaving responsible computing into courses across the cs curriculum. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (pp. 858–864).

Connolly, R. (2020). Why computing belongs within the social sciences. *Communications of the ACM, 63*(8), 54–59.

Dainow, B., & Brey, P. (2021). *Ethics by design and ethics of use approaches for artificial intelligence*. European Commission.

Ellul, J. (1964). *The technological society* (J.Wilkinson, Trans.). Vintage Books. (1954.

Ferreira, R., & Vardi, M. Y. (2021). Deep tech ethics: An approach to teaching social justice in computer science. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (pp. 1041–1047).

Ferri, G. (2014). Ethical communication and intercultural responsibility: A philosophical perspective. *Language and Intercultural Communication, 14*(1), 7–23.

Fiesler, C., Garrett, N., & Beard, N. (2020). What do we teach when we teach tech ethics? A syllabi analysis. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (pp. 289–295).

Fuchs, K. (2021). Innovative teaching: A qualitative review of flipped classrooms. *International Journal of Learning, Teaching and Educational Research, 20*(3), 18–32.

Gotterbarn, D., Miller, K., & Rogerson, S. (1997). Software engineering code of ethics. *Communications of the ACM*, *40*(11), 110–118.

Gotterbarn, D., Miller, K., & Rogerson, S. (1999). Computer society and ACM approve software engineering code of ethics. *Computer, 32*(10), 84–88.

Gray, C. M., Kou, Y., Battles, B., Hoggatt, J., & Toombs, A. L. (2018). The dark (patterns) side of UX design. In *2018 CHI Conference on Human Factors in Computing Systems* (pp. 1–14).

Haselton, M., Nettle, D., & Murray, D. (2015). The evolution of cognitive bias. In Buss, D. M. (Ed.), *The handbook of evolutionary psychology* (pp. 1–20). John Wiley & Sons, Inc. https://doi.org/10.1002/9780470939376.ch25

Heidegger, M. (1977). *The question concerning technology* (W. Lovitt, Trans.). Garland Publishing, Inc.

Herreid, C. F., & Schiller, N. A. (2013). Case studies and the flipped classroom. *Journal of College Science Teaching, 42*(5), 62–66.

Horton, D., McIlraith, S. A., Wang, N., Majedi, M., McClure, E., & Wald, B. (2022). Embedding ethics in computer science courses: Does it work? In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education* (pp. 481–487).

Hysa, X., D'Arco, M., & Kostaqi, J. (2023). Misuse of personal data: Exploring the privacy paradox in the age of big data analytics. In *Big data and decision-making: Applications and uses in the public and private sector* (A. Visvizi, O. Troisi, & M. Grimaldi, Eds.), pp. 43–57. Emerald Publishing Limited. https://doi.org/10.1108/978-1-80382-551-920231004

Janeja, V. (2019). *Do no harm: An ethical data life cycle*.American Association for the Advancement of Science (AAAS). https://www.aaaspolicyfellowships.org/blog/do-no-harm-ethical-data-life-cycle

Johnson, K. (2022). *Meta's new quest pro vr headset harvests personal data right off your face*. https://www.wired.com/story/metas-vr-headset-quest-pro-personal-data-face/

Karim, N. S. A., Ammar, F. A., & Aziz, R. (2017). Ethical software: Integrating code of ethics into software development life cycle. In *2017 International Conference on Computer and Applications (ICCA)* (pp. 290–298).

Karssing, E. D. (2012). Uit de boekenkast van de bedrijfsethiek 44: Stapvoets door een dilemma. opzet en achtergronden van een bewerkingsmodel en rob van es: Professionele ethiek: Morele besluitvorming in organisaties en professies. *Tijdschrift voor Compliance, 12,* 133–138.

Keeney, R. (2020). *Give yourself a nudge: Practical procedures for making better personal and business decisions*. Cambridge University Press.

Knezic, D., Wubbels, T., Elbers, E., & Hajer, M. (2010). The socratic dialogue and teacher education. *Teaching and teacher education, 26*(4), 1104–1111.

Kraught, R. (2022). *Aristotle's ethics*. In Zalta, E. N. & Nodelman, U. (Eds.), *The Stanford encyclopedia of philosophy*. Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/entries/aristotle-ethics/

Lim, M. Y., Leichtenstern, K., Kriegel, M., Enz, S., Aylett, R., Vannini, N., Hall, L., & Rizzo, P. (2011). Technology-enhanced role-play for social and emotional learning context–intercultural empathy. *Entertainment Computing, 2*(4), 223–231.

Mall, R. (2018). *Fundamentals of software engineering (5th ed.)*. PHI Learning.

Martin, C. D., Huff, C., Gotterbarn, D., & Miller, K. (1996). Implementing a tenth strand in the CS curriculum. *Communications of the ACM, 39*(12), 75–84.

Martin, J. P., Desing, R., & Borrego, M. (2022). Positionality statements are just the tip of the iceberg: Moving towards a reflexive process. *Journal of Women and Minorities in Science and Engineering, 28*(4), v–vii.

Mathur, A., Acar, G., Friedman, M. J., Lucherini, E., Mayer, J., Chetty, M., & Narayanan, A. (2019). Dark patterns at scale: Findings from a crawl of 11k shopping websites. In *Proceedings of the ACM on Human-Computer Interaction, 3*(CSCW), 1–32.

Moore, J. (2020). Towards a more representative politics in the ethics of computer science. In *Proceedings of the 2020 Conference on fairness, accountability, and transparency* (pp. 414–424).

Mumford, L. (1934). *Technics and civilisation*. Harcourt, Brace and Company, Inc.

Mumford, L. (1967). *The myth of the machine: Technics and human development* (Vol. 1). Harcourt, Brace Jovanovich.

Narayanan, A. (2019, November 18). *How to recognize AI snake oil*. Arthur Miller Lecture on Science and Ethics, Massachusetts Institute of Technology.

O'Regan, G. (2008). *A brief history of computing*. Springer.

Pavlik, J. V. (2023). Collaborating with chatgpt: Considering the implications of generative artificial intelligence for journalism and media education. *Journalism & Mass Communication Educator, 78*(1), 84–93.

Rudolph, J., Tan, S., & Tan, S. (2023). ChatGPT: Bullshit spewer or the end of traditional assessments in higher education? *Journal of Applied Learning and Teaching, 6*(1), 342–363.

Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes, 35*(3), 8–13.

Shafer-Landau, R. (2018). *The fundamentals of ethics* (4th ed.). Oxford University Press.

Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.

Spiekermann, S. (2021). What to expect from ieee 7000: The first standard for building ethical systems. *IEEE Technology and Society Magazine, 40*(3), 99–100.

Spiekermann, S., & Winkler, T. (2022). Value-Based Engineering With IEEE 7000. *IEEE Technology and Society Magazine, 41*(3), 71–80.

Su, J., & Yang, W. (2023). Unlocking the power of ChatGPT: A framework for applying generative AI in education. In *ECNU review of education* (pp. 355–366).

van de Poel, I. (2015). The problem of many hands. In *Moral responsibility and the problem of many hands* (pp. 50–92). Routledge.

van Veen, M., & Visser-Knijff, P. (2022). 8. Ethics in local politics: A case study of the city of eindhoven. In B. Wernaart (Ed.), *Moral design and technology* (pp. 149–168). Wageningen Academic Publishers. https://doi.org/10.3920/978-90-8686-922-0_8

Wing, J. M. (2019). The data life cycle. *Harvard Data Science Review, 1*(1), 6.

Xu, C. K., & Tang, T. (2020). Closing the gap or widening the divide: The impacts of technology-enabled coproduction on equity in public service delivery. *Public Administration Review, 80*(6), 962–975.