Ethics in (Open)Software

Florien de Graaff[9588078] and Kiranmay van Amerongen[2777037]

Utrecht University, Utrecht, The Netherlands
{f.degraaff, k.g.h.vanamerongen}@students.uu.nl

Abstract. Software plays a critical role in daily life, driving innovation and efficiency across various domains. However, as its prevalence grows, so do the associated risks, including psychological, physical, economic, and environmental harms. Despite advancements in regulating data protection and artificial intelligence through frameworks like the General Data Protection Regulation (GDPR) and the European Union AI Act, legislation addressing the ethical development of software remains insufficient. This research examines the impacts of open and research software across multiple dimensions, investigating how community-driven development ensures ethical software use and maintenance. Through a comprehensive literature review, the study reveals that software is more than a technological artifact—it is a complex socio-technical phenomenon with implications for both human experiences and global systems. The research further supports the need for a regulatory framework that can address the sophisticated challenges posed by increasingly pervasive digital technologies, positioning community-driven development as a critical mechanism for ethical (research)software production.

Keywords: Software harms · Regulation · Ethics · Open Source · Community Development

1 Introduction

Concerns surrounding data protection have been a key focus for policymakers, leading to the introduction of numerous regulations aimed at ensuring responsible handling and security of data. Prominent examples include the *General Data Protection Regulation (GDPR)* [1], the *California Consumer Privacy Act (CCPA)* [2], and the *Personal Data Protection Act (PDPA)* [3]. These regulations prioritize secure data storage, ethical collection practices, and obtaining explicit user consent.

However, while significant progress has been made in regulating data and artificial intelligence (AI), a notable gap remains in legislation addressing the development of ethical and open software. Existing frameworks, such as copyright laws [4] and cybersecurity laws [5], provide limited guidance in this area. Despite the introduction of the European Union (EU) AI Act [6], which establishes comprehensive regulations for the ethical use and governance of AI systems, broader guidelines for creating ethical and open software remain largely undeveloped; leaving an important area of technology regulation insufficiently addressed.

2 Software defined

Software has been in development for a long time, however in the last couple of decades the ground terms have stayed the same, or have been eleborated on. In a paper regarding the question "*What is Software?*" by Peter Suber, the term software is defined as "a pattern that we use to do work"[7]. While this definition is intentionally broad, it highlights that software is not limited to computers; it only impacts them when it is physically embodied, enabling causation. Within the context of computer science, software is more concretely defined as "the instructions that control what a computer does" [8].

It is important to distinguish software from data. According to the Cambridge Dictionary, data refers to "*information, especially facts or numbers, collected to be examined and considered and used to help decision-making, or information in an electronic form that can be stored and used by a computer*" [9]. Therefore, distinction lies in their roles. Software comprises the programs or instructions that direct a computer's operations, whereas data represents the information that these programs process.

Artificial intelligence (AI) introduces additional complexity to these definitions. In a press release by the Council of the EU, AI software is described as *"systems as software developed through machine learning, logic-based, and knowledge-based approaches."* [10] This definition sets AI apart from traditional software by emphasizing its capacity for advanced functionality, ensuring that regulatory efforts specifically target applications with AI capabilities.

To further specify, a distinction will be made between open-source software (OSS) and open research software (ORS).

2.1 Open Software vs Open Research Software

Research software encompasses a broad spectrum of computational tools and programs specifically developed to facilitate scientific research. The FAIR for Research Software (FAIR4RS) working group defines research software as *"source code files, algorithms, scripts, computational workflows, and executables that were created during the research process or for a research purpose"* [11]. This software is intentionally designed to address specific scientific challenges and plays an important role in modern research.

Research software is widely used in various stages of scientific workflows, including data collection, analysis, simulation, and visualization [12]. It enables researchers to handle complex datasets and perform computations that would otherwise be impractical or impossible manually. Moreover, research software contributes to the reproducibility of scientific findings by allowing others to replicate and validate results using the same tools [13].

Open-source software refers to software with source code that is made publicly available under permissive licenses, enabling users to study, modify, and redistribute it [14]. Examples of open software include general-purpose programming languages and frameworks like Python, R, and TensorFlow, which are used across industries and research. Open-source software is typically designed to meet broad technical needs or solve general-purpose problems [15].

Open research software, on the other hand, is a subset of open-source software specifically designed for use in scientific research [16]. It incorporates domain-specific features tailored to address research questions, process scientific data, or simulate phenomena. For example, software such as LAMMPS for molecular dynamics simulations or R packages developed for statistical modelling directly supports the scientific process [17].

Additionally, open research software often adheres to the FAIR principles (Findable, Accessible, Interoperable, and Reusable), ensuring that it can be easily discovered, applied, and adapted by other researchers [11]. Unlike general-purpose open software, research software requires comprehensive documentation and metadata that connect the code to its scientific context, enabling reproducibility and transparency in scientific workflows [18].

While open-source and open-research software share foundational principles of openness, the latter is deeply embedded in the scientific process. By addressing specific scientific challenges and adhering to principles like FAIR, open research software fosters reproducibility, transparency, and collaboration within the research community.

3 Motivation

Software is an integral part of daily life, influencing countless activities. From navigating to work and using public transportation to driving a car, listening to music, or scrolling through social media, software plays a crucial role in shaping everyday experiences. It has significantly enhanced life by enabling connections with friends, facilitating mobile payments, and boosting work efficiency.

However, while there are many benefits of software there are also many risks, including in critical areas such as healthcare, education, employment, law enforcement, and essential infrastructure. We found these dangers include psychological-, physical-, economic- and environmental- harm. Despite the multitude of these harms, there are no regulations for developing software in place.

4 Research question

Therefore, this research aims to answer the question: **"What role does community-driven development (through Open Science at UU) play in ensuring the ethical use and maintenance of open research software?"**.

5 Impacts of Software

5.1 Psychological Impact

Software development, particularly in open-source contexts, has large psychological implications for its contributors and users. According to research by Ge

(2023) [19], these collaborative platforms create spaces that extend beyond traditional work environments, fundamentally transforming how individuals experience professional engagement and personal fulfillment.

One of the most significant positive psychological impacts is the sense of community and belonging. Fitzgerald and Findlay (2011) [20] highlight that when individuals contribute to open-source projects, they experience a powerful feeling of "collective intelligence". Contributors often find deep satisfaction in working towards shared goals, developing a strong sense of purpose that goes beyond individual tasks. This collaborative environment creates meaningful social interactions where participants feel they are part of something larger than themselves, significantly boosting intrinsic motivation [21].

Altruism represents another crucial psychological benefit. A study by Gerosa et al. (2024) [22] reveals that many open-source contributors are driven by a desire to make a meaningful difference, which enhances personal satisfaction and psychological well-being. The study of Isham and Jackson (2022) [23] further demonstrates that this leads to participants experiencing "flow" which is a state of complete immersion in challenging, yet meaningful work. This intrinsic motivation transforms software development from 'just' a technical task into a deeper and more rewarding personal journey.

However, the open-source landscape is not without psychological challenges. Zhang and Ke (2008) [24] shed light on burnout as a significant negative impact, particularly for voluntary contributors. The expectation of continuous participation can lead to overwhelming stress, especially as projects grow in scope and complexity. Many passionate developers find themselves overcommitted, experiencing feelings of inadequacy and emotional exhaustion.

Moreover, the psychological strain of a lack of recognition is an effect of developing open software. Contributors may feel their efforts are underappreciated, leading to frustration and potential disengagement. The voluntary nature of many open-source projects means that individuals invest significant emotional and intellectual energy without guaranteed acknowledgment or support [24].

Another critical psychological harm emerges from the potential for identity conflict and imposter syndrome. Open-source communities often have intense peer review and collaborative processes that can create significant psychological pressure. Participants may constantly feel their work is under scrutiny, leading to self-doubt and anxiety about their technical competence and professional worth. This persistent evaluation can create a psychologically challenging environment where contributors may experience heightened feelings of inadequacy and fear of judgment [22].

These psychological dynamics reveal that open software is more than just a technological phenomenon. It is a complex human experience that affects various areas, such as, individual motivation, well-being, and professional identity. It is important to understand these psychological nuances so communities can create more supportive environments that nurture both technological innovation and individual growth.

5.2 Physical Impact

The growth of software in daily life extends beyond digital interactions, significantly impacting human physical well-being. Moreno-Llamas et al. (2020) [25] provides insights into how digital technologies fundamentally alter physical health and lifestyle patterns.

One of the most immediate physical concerns is the impact of digital interfaces on human sensory systems. Extended screen time, particularly with software applications using intense blue light, can lead to substantial physiological stress. A study by Chang et al. (2015) [26] in the Proceedings of the National Academy of Sciences demonstrated that exposure, in the evening, to light-emitting digital devices dramatically suppresses melatonin production, disrupting circadian rhythms and potentially compromising sleep quality and overall physiological recovery [25].

Sedentary behavior represents another critical physical consequence of increasing software dependency. Owen et al. (2010) [27] highlight the metabolic risks associated with prolonged sitting, linking excessive sedentary time to increased risks of cardiovascular disease, diabetes, and reduced life expectancy. As digital technologies become more integrated into professional and personal lives, they inadvertently promote reduced physical movement [25].

Vision health represents a critical concern in the digital age. A study by [28] found that prolonged screen time is associated with digital eye strain, characterized by symptoms like dry eyes, blurred vision, and headaches. The researchers noted that the average adult spends over 7 hours daily interacting with digital screens, significantly increasing the risk of vision-related complications.

Simultaneously, software and digital technologies offer potential positive interventions for physical health. Smartphone applications and wearable technologies have demonstrated remarkable potential in promoting physical activity and health monitoring. A study found that digital health interventions can effectively increase physical activity levels and provide personalized health recommendations [29].

These physical dynamics reveal the complex relationship between software and human physicality. These are far from being a purely digital phenomenon; software profoundly mediates our physical experiences, presenting both significant challenges and innovative opportunities for maintaining and enhancing human physical health.

5.3 Economic Impact

The economic landscape of open and research software represents a complex ecosystem of innovation, cost dynamics, and transformative potential. Garzarelli (2003) provides insights into the economics of open-source software organization, revealing its impact on technological and economic structures [14].

Cost efficiency emerges as a primary economic advantage. Colombo et al. (2013) [30] demonstrates that open-source solutions enable organizations, particularly small and medium-sized enterprises (SMEs), to significantly reduce operational expenses. A different study by Russo and Succi (2020) shows that companies adopting open-source software experienced an average cost reduction of 20-30% in IT infrastructure expenditures [31].

Innovation represents another crucial economic dimension. Fitzgerald (2006) [15] highlights how open software lowers entry barriers for developers and researchers, facilitating rapid tool development across industries. A different study revealed that open-source models drive innovation by creating collaborative ecosystems that accelerate technological development [32]. This is particularly evident in specialized domains like bioinformatics, where Gentleman et al. (2004) [33] demonstrates how open software has revolutionized computational research capabilities.

Intellectual property dynamics add complexity to the economic model. Von Krogh and Spaeth (2007)[16] explore the unique characteristics of open-source software that challenge traditional economic frameworks. However, the economic model is not without challenges. Sustainability emerges as a critical concern. There are also potential hidden economic risks, including security vulnerabilities, software obsolescence and compatibility issues [34].

Economic challenges are also created when creating open source projects. Research by Gamalielsson and Lundell (2014) highlighted the economic challenges of maintaining open-source projects, demonstrating that long-term sustainability requires robust funding and community support to survive [35].

Market dynamics present additional complexities. While open-source software promotes innovation, it creates unique economic ecosystems. As the study of Lerner et al. (2006) [36] shows how open-source models create alternative value generation mechanisms, challenging traditional commercial software development approaches.

Commercial adaptation has been particularly interesting. Companies like Red Hat have successfully monetized open-source models by providing enterprise-level services, demonstrating the economic viability of open-source approaches [37]. These economic dynamics reveal open software as more than a technological trend. It is a transformative force reshaping how organizations innovate, compete, and create value in an increasingly digital global economy.

5.4 Environmental Impact

The relationship between software and environmental sustainability represents a critical yet often overlooked dimension of technological development. While digital technologies are frequently associated with environmental challenges, new research reveals their potential for positive ecological interventions [38].

Energy consumption emerges as a primary environmental concern in software development. Data centers and computing infrastructure contribute significantly to global carbon emissions, with estimates suggesting that information and communication technologies account for approximately 2-4% of worldwide greenhouse gas emissions [39].

However, open-source software models present innovative pathways for mitigating these environmental impacts. Open-source software can drive energy efficiency through collaborative optimization. Enabling multiple developers to collectively improve code performance, these collaborative platforms can reduce computational resource requirements.

A study by lsayyah and Ahmed (2020) [38] demonstrated that open-source optimization techniques could potentially reduce energy consumption in computing systems, presenting a significant environmental benefit.

Moreover, open research software plays a crucial role in environmental monitoring and climate research. Specialized scientific software enables researchers to model complex environmental systems, track biodiversity changes, and simulate climate scenarios with unprecedented precision [40]. Tools like OpenStreetMap and climate modeling software exemplify how open-source approaches directly contribute to environmental understanding and conservation efforts [41].

Circular economy principles are increasingly reflected in open software development. Promoting software reusability, modular design, and long-term maintenance, open-source models challenge the traditional "develop-and-discard" technological paradigm. This approach not only reduces electronic waste but also extends the lifecycle of computational resources [42].

Interestingly, the collaborative nature of open software communities creates environmental consciousness. Developers often integrate sustainability considerations into their design principles, creating a culture that values resource efficiency and ecological responsibility. This community-driven approach transforms software development from a purely technical endeavor into a holistic practice considering broader environmental implications.

While challenges remain, the potential of open software to drive environmental sustainability is large. Through the promotion of energy-efficient design, supporting critical environmental research, and cultivating a culture of technological responsibility, open software communities are emerging as key stakeholders in global environmental conservation efforts.

6 Limitations

While offering insights, this study acknowledges several inherent limitations that limit its generalizability and depth. The research is predominantly contextualized within a European academic framework, which potentially constrains the global transferability of its findings. The temporal specificity of the investigation presents another methodological challenge, as the rapid evolution of technological ecosystems means that contemporary observations may rapidly become obsolete.

The study's reliance on extensive literature review, while providing a robust theoretical foundation, simultaneously represents a limitation in empirical depth. Direct primary data collection could potentially yield more nuanced and contextually grounded insights into the complex dynamics of software development communities. Moreover, the focus on open research and open-source software necessarily narrows the research's scope, potentially overlooking broader software development paradigms. The selected sources and studies may inadvertently introduce interpretative biases, either by overemphasizing positive technological narratives or by presenting overly critical perspectives. However, by highlighting both sides an attempt has been made to diminish the chance of this happening.

7 Conclusion & Future Work

This research has examined the impacts of software, particularly open and research software, across psychological, physical, economic, and environmental domains. From this research it has become clear that software is far from being a mere technological artifact. It emerges as a complex socio-technical phenomenon with implications for human experience and global systems.

This review reveals a landscape where software development intersects with human experiences. Psychologically, open software communities represent more than collaborative platforms; they are intricate social ecosystems that generate collective intelligence. These environments simultaneously offer opportunities for personal fulfillment and professional engagement, while also presenting significant psychological challenges such as contributing to a burnout and identity-related anxieties.

Physically, software has transcended its digital origins to become a mediator of human corporeal experiences. In this research, it becomes clear how digital technologies simultaneously pose risks, such as sedentary behaviors and sensory system disruptions and create innovative interventions for health monitoring and physical well-being. This dual nature, furthermore, underscores the complexity of software's relationship with human physicality.

Economically, open software models are transforming traditional organizational and innovation paradigms. By challenging established commercial frameworks, these models generate alternative value creation mechanisms, enabling cost efficiencies and lowering technological entry barriers. The economic dynamics reveal open software as a force reshaping competitive landscapes and technological development strategies.

Perhaps most critically, the environmental dimension of software development emerges as a crucial domain in open science. Open software communities are positioning themselves as potential key stakeholders in global sustainability efforts, with the capacity to reduce computational carbon footprints and support critical environmental research through collaborative optimization techniques.

Regarding the research question, **"What role does community-driven development (through Open Science at UU) play in ensuring the ethical use and maintenance of open research software?"**, the study reveals that community-driven development serves as a critical mechanism for ethical software governance. Through collaborative platforms, open science communities at Utrecht University and beyond create dynamic, self-regulating ecosystems that inherently address ethical challenges. These communities foster transparency, collective accountability, and continuous improvement through peer review, shared values, and collaborative optimization. Developers within these environments are motivated not just by technical objectives, but also by broader ethical considerations. This would include psychological well-being, environmental sustainability, and social responsibility.

The collaborative nature of open science enables rapid identification and mitigation of potential ethical risks, creates shared standards of practice, and creates a culture of technological stewardship that goes beyond traditional top-down regulatory approaches. Moreover, the community-driven model allows for adaptive, context-sensitive ethical frameworks that can rapidly respond to emerging technological challenges. As a result, making it a more flexible and responsive approach to software ethics in comparison to static and institutionalized mechanisms.

7.1 Future Works

Future research investigations should prioritize several key areas: developing more comprehensive global and cross-cultural research methodologies, creating an empirical framework for analyzing software development ecosystems, and constructing adaptive ethical guidelines that are able to keep pace with technological transformations.

Despite the limitations, this research provides a critical academic contribution to understanding the complex ethical dimensions of software development. Highlighting the various impacts of software across psychological, physical, economic, and environmental domains, this study challenges reductive technological narratives and advocates for a more holistic, interdisciplinary approach to understanding digital innovation.

The compelling evidence presented underscores an urgent imperative: the need for comprehensive, nuanced regulatory frameworks that can adequately address the sophisticated challenges posed by increasingly pervasive digital technologies. As software continues to reshape human experiences, interdisciplinary attention becomes not just an academic exercise, but a critical social responsibility.

Bibliography

[1] European Union. General data protection regulation (gdpr). Official Journal of the European Union, L 119, 1–88, 2016.

[2] State of California. California consumer privacy act (ccpa). California Civil Code, Section 1798.100 et seq., 2018.

[3] Personal Data Protection Commission. Personal data protection act (pdpa). No. 26 of 2012, 2012.

[4] European Union. Directive 2001/29/ec of the european parliament and of the council on the harmonization of certain aspects of copyright and related rights in the information society. Official Journal of the European Union, L 167, 10–19, 2001.

[5] European Union. Regulation (eu) 2019/881 of the european parliament and of the council on enisa, the european union agency for cybersecurity, and on information and communications technology cybersecurity certification, and repealing regulation (eu) no 526/2013. Official Journal of the European Union, L 151, 15–40, 2019.

[6] European Union. Regulation (eu) 2024/1689 of the european parliament and of the council of 16 august 2024 on harmonised rules on artificial intelligence and amending certain union legislative acts. EUR-Lex, 2024.

[7] P. Suber. What is software? *The Journal of Speculative Philosophy*, 2(2):89–119, 1988.

[8] Cambridge Dictionary. software. `https://dictionary.cambridge.org/dictionary/english/software`, 2024.

[9] Cambridge Dictionary. data. `https://dictionary.cambridge.org/dictionary/english/data`, 2024.

[10] European Council. Artificial intelligence act: Council calls for promoting safe ai that respects fundamental rights. Press release by the European Council, 2022.

[11] M. B. M. Bloemers and A. M. A. Montesanti. The fair funding model: Providing a framework for research funders to drive the transition toward fair data management and stewardship practices, 2022. [Dataset].

[12] Jo E. Hannay, Carolyn MacLeod, Janice Singer, Hans Petter Langtangen, Dietmar Pfahl, and Greg Wilson. How do scientists develop and use scientific software? In *2009 ICSE Workshop on Software Engineering for Computational Science and Engineering*, pages 1–8. IEEE, 2009.

[13] Roger D. Peng. Reproducible research in computational science. *Science*, 334(6060):1226–1227, 2011.

[14] Giampaolo Garzarelli. Open source software and the economics of organization. In *Routledge eBooks*, pages 63–78. Routledge, 2003.

[15] Brian Fitzgerald. The transformation of open source software. *MIS Quarterly*, 30(3):587–598, 2006.

[16] Georg Von Krogh and Susanne Spaeth. The open source software phenomenon: Characteristics that promote research. *The Journal of Strategic Information Systems*, 16(3):236–253, 2007.

[17] Michael A. Heroux and James M. Willenbring. Barely sufficient software engineering: 10 practices to improve your cse software. In *2009 ICSE Workshop on Software Engineering for Computational Science and Engineering*, pages 15–21. IEEE, 2009.

[18] Mark D. Wilkinson, Michel Dumontier, I. J. Aalbersberg, Gavin Appleton, Michael Axton, Aart Baak, Niels Blomberg, J. Boiten, L. B. Da Silva Santos, Philip E. Bourne, Joost Bouwman, Andrew J. Brookes, Timothy Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Susan Edmunds, Chris T. Evelo, Reiner Finkers, et al. The fair guiding principles for scientific data management and stewardship. *Scientific Data*, 3(1):160018, 2016.

[19] N. Ge. *Fulfilling the Psychological Needs of Open-Source Software Developers*. Doctoral dissertation, Manchester Business School, 2023.

[20] R. Fitzgerald and J. Findlay. Collaborative research tools: Using wikis and team learning systems to collectively create new knowledge. In *The handbook of emergent technologies in social research*, pages 300–319. SAGE Publications, 2011.

[21] D. Bandhu, M. M. Mohan, N. a. P. Nittala, P. Jadhav, A. Bhadauria, and K. K. Saxena. Theories of motivation: A comprehensive analysis of human behavior drivers. *Acta Psychologica*, 244:104177, 2024.

[22] Marco Gerosa, Igor Wiese, Bianca Trinkenreich, Georg Link, Gregorio Robles, Christoph Treude, Igor Steinmacher, and Anita Sarma. The shifting sands of motivation: Revisiting what drives contributors in open source. *ArxIv*, 2021.

[23] A. Isham and T. Jackson. Finding flow: exploring the potential for sustainable fulfilment. *The Lancet Planetary Health*, 6(1):e66–e74, 2022.

[24] P. Zhang and W. Ke. Motivations for participating in open source software communities: Roles of psychological needs and altruism. *SSRN Electronic Journal*, 2008.

[25] A. Moreno-Llamas, J. García-Mayor, and E. De La Cruz-Sánchez. The impact of digital technology development on sitting time across europe. *Technology in Society*, 63:101406, 2020.

[26] Anne-Marie Chang, Daniel Aeschbach, Jeanne F. Duffy, and Charles A. Czeisler. Evening use of light-emitting ereaders negatively affects sleep, circadian timing, and next-morning alertness. *Proceedings of the National Academy of Sciences of the United States of America*, 112(4):1232–1237, 2015.

[27] Neville Owen, Phillip B. Sparling, Genevieve N. Healy, David W. Dunstan, and Charles E. Matthews. Sedentary behavior: Emerging evidence for a new health risk. *Mayo Clinic Proceedings*, 85(12):1138–1141, 2010.

[28] Kanwal Kaur, Bhavana Gurnani, Satya Nayak, Neelam Deori, Simran Kaur, Jitendra Jethani, Dilraj Singh, Shail Agarkar, Jameel Rizwana Hussaindeen, Jaspal Sukhija, and Devendra Mishra. Digital eye strain - a comprehensive review. *Ophthalmology and Therapy*, 11(5):1655–1680, 2022.

[29] Kym J. Brickwood, Glen Watson, Joseph O'Brien, and Andrew D. Williams. Consumer-based wearable activity trackers increase physical activity participation: Systematic review and meta-analysis. *JMIR mHealth and uHealth*, 7(4):e11819, 2019.

[30] M. G. Colombo, E. Piva, and C. Rossi-Lamastra. Open innovation and within-industry diversification in small and medium enterprises: The case of open source software firms. *Research Policy*, 43(5):891–902, 2013.

[31] Barbara Russo and Giovanni Succi. A cost model of open source software adoption. *International Journal of Open Source Software and Processes*, 1(3):60–82, 2009.

[32] Joel West and Sean Gallagher. Challenges of open innovation: the paradox of firm investment in open-source software. *R & D Management*, 36(3):319–331, 2006.

[33] R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. J. Rossini, and J. Zhang. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5(10):R80, 2004.

[34] M. AlMarzouq, L. Zheng, G. Rong, and V. Grover. Open source: Concepts, benefits, and challenges. *Communications of the Association for Information Systems*, 16, 2005.

[35] Jonas Gamalielsson and Björn Lundell. Sustainability of open source software communities beyond a fork. *J. Syst. Softw.*, 89(C):128–145, March 2014.

[36] Josh Lerner, Prashant A. Pathak, and Jean Tirole. The dynamics of open-source contributors. *American Economic Review*, 96(2):114–118, 2006.

[37] X. Xie. The profitability and effectiveness of merger arbitrage: Evidence from ibm & red hat. In *Advances in Computer Science Research*, pages 308–313. 2023.

[38] A. A. Alsayyah and S. Ahmed. Energy-efficient software development techniques for cloud-based applications. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(5):8043–8053, 2020.

[39] J. C. Bieser, R. Hintemann, L. M. Hilty, and S. Beucker. A review of assessments of the greenhouse gas footprint and abatement potential of information and communication technology. *Environmental Impact Assessment Review*, 99:107033, 2023.

[40] Yogesh K. Dwivedi, Laurie Hughes, Arpan Kumar Kar, Abdullah M. Baabdullah, Purva Grover, Roba Abbas, Daniela Andreini, Iyad Abumoghli, Yves Barlette, Deborah Bunker, Leona Chandra Kruse, Ioanna Constantiou, Robert M. Davison, Rahul De', Rameshwar Dubey, Henry Fenby-Taylor, Babita Gupta, Wu He, Mitsuru Kodama, Matti Mäntymäki, Bhimaraya Metri, Katina Michael, Johan Olaisen, Niki Panteli, Samuli Pekkola, Rohit Nishant, Ramakrishnan Raman, Nripendra P. Rana, Frantz Rowe, Suprateek Sarker, Brenda Scholtz, Maung Sein, Jeel Dharmeshkumar Shah, Thompson S.H. Teo, Manoj Kumar Tiwari, Morten Thanning Vendelø, and Michael Wade. Climate change and cop26: Are digital technologies and

information management part of the problem or the solution? an editorial reflection and call to action. *International Journal of Information Management*, 63:102456, 2022.

[41] OpenStreetMap Contributors. Openstreetmap. Available at: `https://www.openstreetmap.org/#map=7/52.154/5.295`, 2025.

[42] Ahmed M. Elgarahy, M.G. Eloffy, A.K. Priya, Ahmed Hammad, Moustafa Zahran, Ali Maged, and Khalid Z. Elwakeel. Revitalizing the circular economy: An exploration of e-waste recycling approaches in a technological epoch. *Sustainable Chemistry for the Environment*, 7:100124, 2024.