TOWARDS ACTIONABLE INFORMATION SYSTEM ETHICS: REQUIREMENTS ANALYSIS AND NON-MALEFICENCE

Dimitrius Keykaan
School of Computer Science and Information Systems, North-West University
11 Hoffman Street, Potchefstroom, South Africa

ABSTRACT

Unethical Software development is the cause of many harms that emanate from information systems. This paper proposes the explicit use of the non-maleficence bioethical principle, to assess whether system requirements may have ethical implications. The assessment takes place before the requirement advances to a following phase in the software development life cycle. Some harm caused by unethical Information systems will be highlighted to show the reality of the harm caused. An unethical software development scenario is then created, and a systems requirement of the system is formulated. A single bioethical principle is then used, as criteria, to determine whether a particular system requirement may be advanced to the design and implementation phases of the information systems development life cycle. The chosen approach has indicated usefulness in working towards actionable ethics that are not simply a long list of rules and principles. This research advises on the exact point in the development process, where ethical considerations can be made. It also guides the developer on how to determine whether a system could potentially cause harm or not. The actionability of the approach is also considered beneficial because takes place during the design process instead of after the design stages, where little can be done to influence ethicality.

KEYWORDS

Ethics, Information Systems, Non-Maleficence, Harm, Requirements Analysis

1. INTRODUCTION

Organizations often implement information systems to improve effectiveness, efficacy and to assist in reaching organizational goals (Hevner *et al.*, 2004). These goals generally include but are not limited to sustainability, growth and profitability that can enhance an organizations competitive ability (Abualoush *et al.*, 2018). As a result of growing organizations and new technologies, information systems are implemented though the use of software processes i.e. software engineering and software development, which enables keeping up with evolving organizational needs (Gregg *et al.*, 2001). Going forward, for the purpose of this paper, this implementation will simply be referred to as information systems (IS). The implementation of IS has been adopted by various industries in modern / e-society i.e., health (Eltajoury *et al.*, 2021), accounting (Darma, 2018), transport (Shmeleva *et al.*, 2019) and hospitality (Sever & Kağnıcıoğlu, 2019), just to mention a few. Over time it has been seen that IS can indeed contribute to reaching organizational goals and to have reasonably improved effectiveness and efficacy. Despite these successes, concerns of harm causing IS have also been raised (Pirkkalainen & Salo, 2016). This study views harm of any kind, to human beings, society as a whole or the environment as unethical and therefore any act of harm that comes from the use or implementation of IS is also considered unethical.

To mitigate this problem, ethical intervention in the software development process is proposed.

The Association for Computing Machinery (ACM), in its *ACM Code of Ethics and Professional Conduct*, has pointed out that ethical issues in IS can be addressed by software developers (Gotterbarn *et al.*, 2018). However, it does not explicitly guide the developer on when or at what point of development ethics should be considered. This paper proposes the sole and explicit use of a single bioethical principle, *non-maleficence,* to be used as criteria on determining whether a given *system requirement* should advance to the *design and implementation phases* of the software development life cycle, or not. The *non-maleficence* principle considers the potential harm that can be caused on the basis of a particular choice made. The intended contribution of this paper is to make ethics in IS actionable by simplifying the ethical considerations of the developer and providing guidance on how to minimize the potential harm that may emanate from the implementation of an IS. Therefore, the following research question has been formulated:

How can non-maleficence, as a bioethical principle, be used to address the potential harm that information systems may cause?

The remainder of this paper will be structured as follows: In Section 2, examples of unethical behavior in information systems will be presented. In Section 3, the software development process will be considered to emphasize where ethics can be used. Existing ethical approaches will then be considered in Section 4 and Section 5 will detail the proposed approach to using *non-maleficence* in the development of IS software. Some limitations will be considered in Section 6 and then following a brief reflection, the paper will be concluded in Section 7.

2. UNETHICAL BEHAVIOUR IN INFORMATION SYSTEMS

Harm caused by information systems (IS) is not limited to human beings, unethical behavior of IS also extends to the environment. In this section, unethical behavior of IS will be discussed in terms of harm caused to people and the environment, examples of each will be presented. The examples do not seek to serve as a literature review, it is only to demonstrate the reality of unethical behavior in IS.

2.1 Harm on Human Beings

The use of IS in organizations, though popular today, is not a recent practice. In 1985 the US and Canadian governments implemented a system called *Therac-25*, in their respective healthcare departments (Leveson & Turner, 1993). This system was intended to assist in contributing benefit to the patients who it served and healthcare professionals who depended on it. Leveson and Turner (1993) however, reveal that the system contained software design problems that led to grievous harm, including the death of 4 patients and serious injury of 2 others. Since its occurrence, this incident has received a lot of attention and has been cited in many research studies (McQuaid, 2012; Porrello, 2012; Silvis-Cividjian, 2021; Zelkowitz, 2012). Despite software professionals having become aware of possible harm, recent reports confirm that this problem is still found in IS implementations. Different views on whether the developers of *Therac-25* were deliberately negligent when programming the system have been shared. However, a common view is that the harm caused was avoidable by design.

Recent implementations of harmful IS include persuasive information systems (PIS) that persuade people into doing things they would not normally have done such as repeated and dangerous online gambling (Benner *et al.*, 2021). In PIS the developers deliberately design a system that will persuade individuals in a certain way, without considering the harm it may cause the gambler or their families. Another implementation is that of addictive information systems (AIS) that employ psychological tactics to ensure users continue using the system (Cemiloglu *et al.*, 2020). Examples of AIS include Facebook that has caused people to present with behavioral addiction symptoms such as mood modification, withdrawal, too much focus on the IS which can lead to job loss etc. (Turel, 2015). Addiction that results in the loss of a job leads to financial difficulty in the individual’s life which causes harm to the individual and involved family members. An IS is therefore considered to have unethical effects if it is designed in a way that would harm human beings in the above-mentioned ways. Sometimes the negative effects of IS extend further than just individuals or groups of society, though they are also affected. The next subsection will show how IS harm can also affect the environment.

2.2 Harm on the Environment

Much research has been able to point out how unethical IS causes harm in the lives of human beings. In addition to human harm, research has shown that the environment can also be harmed by the implementation of IS.

Examples of this is covered in the emergence of a term often referred to as *green IS* (Gholami *et al.*, 2013) that address environmental issues such as energy consumption by IS, sustainability and related environmental challenges etc. (Dao *et al.*, 2011; vom Brocke *et al.*, 2013).

A very famous example of environmental harm deliberately caused by an unethical IS is the case of an automobile manufacturer (Volkswagen), that employed software designed to present false results during an emissions test (Mansouri, 2016). In 2015 Volkswagen was found to have designed its software in a way that it would disable certain emission control measures, making it seem as if the engine is emitting less harm causing gasses into the environment (Contag *et al.*, 2017). The act was unethical in the sense that it presented false values that would keep Volkswagen out of trouble with the Environmental Protection Agency (EPA) but also because of the environmental damage it caused with its excessive amount of nitrogen oxides and carbon monoxide released into the environment (Klier & Linn, 2016).

Volkswagen admitted that it knowingly had unethical software designed and installed in their engines and they paid a criminal penalty of $2.8 billion (Rogerson *et al.*, 2017). If this kind of software was not designed, then VW would not have been able to cheat. This research, therefore, contends that unethical software should be identified and stopped in the development process. In the next section, the information system development lifecycle will be discussed and an area for the consideration of ethical software development will be identified. Once the area for ethical considerations is established, ethical approaches to the problem can be looked at.

3. INFORMATION SYSTEMS DEVELOPMENT LIFE CYCLE

The information systems development life cycle (SDLC), also called the software development life cycle, is the process of creating and maintaining information systems. To date, various approaches to the SDLC exist (Kute & Thorat, 2014). In this study, the two majorly recognized approaches will be covered to demonstrate that different approaches are, in principle very much the same, in terms of collecting requirements from various stakeholders, before designing a system. The first approach is known as the *traditional approach* and the second approach is known as the *agile approach* (Leau *et al.*, 2012).

3.1 Traditional SDLC Approach

The traditional approach, also known as the waterfall approach (Kisielnicki & Misiak, 2017), is one with six phases. The phases must be followed in a sequential fashion, one phase must be completed before the next one is started with, and the phases must always be followed in the same order (Sinha & Das, 2021).

The phases in the traditional approach are:

- Requirements: user requirements for the system are gathered from stakeholders.
- Design: planning program architecture and logical flow of program before implementation.
- Implementation: actual coding of the software system, based on requirements and design.
- Verification: testing the software against the requirements that were documented in phase one.
- Deployment: deployment of the system, into production environment.
- Maintenance: fix production bug(s)/defect(s) and push further build in production with cleaner code

All the phases are completed only once, then after deployment, only maintenance is possible, this approach does not allow for the return to any prior phase,

3.2 Agile SDLC Approach

The major difference between the traditional and the agile approaches is that the agile approach allows for the return to previous phases. The development team meets with the stakeholders often and are able to deliver a small part of the final program. The team then has the customer review that small iteration, and subsequently return to implement changes for a newer version of the same system. This is obtained by splitting a large project into a set of smaller subprojects that are completed in short time periods (Usman & Ogwueleka, 2018). Simply put, the agile approach can follow the same phases of the traditional approach with the exception that, after implementation, it can be returned to requirements and design and iterate through those phases until all stakeholders are satisfied.

3.3 Post Requirements Ethical Considerations

From looking at the different approaches to the SDLC it is clear that, irrespective of the chosen approach, requirements are gathered from the stakeholder before coding is finalized. The design and implementation phases of the system are based on the IS requirements gathered i.e., functional, and non-functional requirements. This points to the fact that requirements determine how IS behave, therefore it is argued that unethical behavior stems from collected system requirements. On the basis of this argument, this paper proposes that there be a pause for ethical considerations of each requirement before it is allowed to advance to the design and implementation phases.

Figure 1 shows how each requirement, that would normally go directly from the requirements phase to the design phase, now first needs to go through a form of ethical consideration. Should the requirement not meet the given criteria, whatever the criteria may be, the requirement should be reviewed before advancing to the design and implementation phases. This new phase, for the purpose of this paper will be referred to as the ethical considerations phase (ECP).

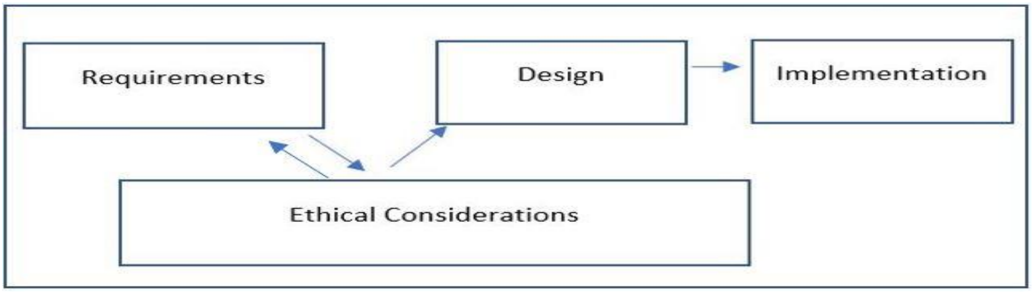

Figure 1. Post requirements ethical considerations phase (ECP)

In the next section various ethical principles or approaches will be discussed and one of those principles will be used as criteria in the ECP.

4. ETHICAL PRINCIPLES

To date, several attempts have been worked on to ensure ethical software development. One of the most famous efforts is found in the ACM code of conduct for software professionals (Gotterbarn *et al.*, 2018). The code of conduct lists a variety of principles that a software developer should follow to produce ethical software. In principle, the code of conduct addresses most ethical concerns that one may have, but it seems the code of conduct is not actionable. McNamara *et al.* (2018) did a research study on a large group of students and IT professionals (participants) and found that the explicit instructing of participants, to use the ACM code of conduct, had no effect. This code of conduct will therefore not be considered as criteria. The code of conduct also does not guide the developer to the point of the development process, where ethics should be applied. It also fails to, clearly, guide on the order in which the code should be followed. Such undetailed and lengthy codes of conduct or practice may be the reason why its ethical guidance is not actionable. It is the view of this research that minimal ethical principles may lead to actionable ethics for IS.

Out of the many minimal or less complex sets of ethical principles that exist, and have been applied in multiple fields, is bioethical principles. Among other fields, bioethics has been used in education (Nunes *et al.*, 2015), legal matters (Kirillova *et al.*, 2020) and in politics (Pelluchon, 2016).

In the absence of another IT related set of minimal or ethical principles or code of conduct, bioethical principles will be considered as criteria for the ethical considerations phase (ECP).

According to Kuhse and Singer (1998) Bioethics is a set of principles derived from the Hippocratic oath which is normally taken by medical professionals. Healthcare professionals often use these principles to guide them in making ethics related decisions. The four ethical principles of bioethics are beneficence, non-maleficence, justice, and autonomy. Each of the four principles can be summarized as follows:

(Aksoy & Elmali, 2002)

- Beneficence: doing good and promoting the welfare of others
- Non-maleficence: avoid causing harm unto others
- Justice: be fair to all party's involved
- Autonomy: value the freedoms and individual worth of others

Each of the four principles serves its own purpose and that is why they are initially listed as separate principles. This paper, however, contends that the principle of *non-maleficence* is an implicit and underlying idea in the remaining three bioethical principles. If this principle is adhered to then it will be difficult to go against any of the others. If one avoids causing harm to others (*non-maleficence)* it will be difficult to be unfair (*justice),* deprive others of their individual freedom (*autonomy)* or to deliberately disregard the welfare of others (*beneficence).* Based on this reasoning and an attempt to make information systems ethics actionable, through simplification, only the principle of *non-maleficence* will be used as criteria. It can then be determined whether system requirements have the potential to cause harm.

In the next section a system requirement will be judged using the criteria of non-maleficence. A summary of the required action will then be given, and this can be applied to other requirements as well.

5. NON-MALEFICENCE POST REQUIREMENTS CONSIDERATION

Due to page constraints the application of the non-maleficence principle will only be done based on one requirement of an IS that has already been discussed in section 2. The Volkswagen case of having developed software, that deliberately lies about the amount of harm causing gases released into the environment. It is assumed that one of the system requirements was similar to:

Display correct emissions only if it is below the legal limit, else manipulate the results to display emissions below the legal limits. Do this even if emissions are higher than the legal limit.

If a developer is faced with a requirement such as this one, using the non-maleficence principle, the developer must consider the following criterion question:

Will the use of this requirement possibly cause harm to any of the stakeholders or the environment?

The developer must then refer to the list of all the stakeholders (normally defined in the requirements phase) and consider whether this requirement can potentially harm any of them or the environment.

If, after consideration, the answer to the question is "NO", then the requirement may be advanced to the next SDLC phase. If the answer is "YES" or the answer cannot be "NO" for sure, then the requirement must be reviewed and may not be advanced to the next phase. This would mean that the requirement must stay in the requirements phase, where alternative avenues are discussed with stakeholders, until a newer version of the requirement is ready for another round of ethical consideration. Honest and comprehensive consideration of the criterion question is likely to increase the effectiveness of this approach. The answer will not always be a simple "yes" or "no", some alternatives may include "possibly", "maybe", "sometimes" etc., all of these responses are also not "NO" for sure and therefore the related requirement will need to be reviewed until the answer is "NO".

In the case of the assumed requirement above, the answer to the criterion question is "YES" because the environment will be harmed if emissions above the legal limit are released into it. The developer(s) must thus return the requirement to the requirements phase and alternatives must be considered.

6. LIMITATIONS

Actionable information systems ethics are expected to contribute to the minimizing of harmful software, however, optimal results would rely on more research in future. At this early stage of research on the topic, there are some limitations that can be addressed in future work. The process, though simplified and actionable, requires a software developer to be honest about the potential harm that a system requirement is likely to cause.

If the developer does not consider all stakeholders and the environment, then the process may be less effective. This study currently only focuses on the assessing of individual system requirements, which are allowed to advance to the design and implementation phases. There exists a possibility that the combination of two requirements, which were individually considered not to cause potential harm, may lead to unexpected unethical consequences once in design or implementation. This possibility, though unlikely, should also be considered and researched in future.

7. CONCLUSION

If not acted upon, from an ethical point of view, IS caused harm may be perpetuated for years to come. Based on this reasoning, a research question was formulated in Section 1, and it will now be readdressed:

How can non-maleficence, as a bioethical principle, be used to address the potential harm that information systems may cause?

In Section 2 it was shown that unethical IS does cause harm and that the harm caused can be serious. This makes it necessary to avoid IS related harm as far as possible. In Section 3, the post requirements phase of the SDLC was identified as an effective area for the use of ethical considerations. Section 4 presented non-maleficence as an ethical principle that covers concerns addressed in other bioethical principles and for that reason it was selected as the principle that will be used as criteria for system requirements. In Section 5 it was demonstrated that non-maleficence can be used to determine whether a particular system requirement is likely to cause harm to stakeholders or the environment. A non-maleficence criterion question was asked and based on the answer, the requirement will advance to the next SDLC phase, else it will be reviewed until it satisfies the criterion question. The advantage of this approach is that it is not complex or tedious, it is actionable due to its simplicity and the precise indication of where it should be applied.

This study has shown that the use of a single bioethical principle is useful in moving toward actionable ethics in IS. As actionable ethics for IS is still at a very early stage, possible application of the contribution made in this paper would be to serve as a basis for future research but also to supplement existing attempts in raising awareness of the harms caused by unethical IS. Future work will include actionable ethics in IS teaching and learning. Furthermore, actionable ethics in data management for Machine learning applications as well as more ethical approaches to IS development will also be investigated.

REFERENCES

Abualoush, S. H., et al. 2018. The role of employees' empowerment as an intermediary variable between knowledge management and information systems on employees' performance. *VINE Journal of Information and Knowledge Management Systems,* 48(2):217-237.

Aksoy, S. and Elmali, A. 2002. The core concepts of the four principles of bioethics as found in Islamic tradition. *Med. & L.,* 21(1):211-224.

Benner, D., et al. 2021. It is only for your own good, or is it? Ethical Considerations for Designing Ethically Conscious Persuasive Information Systems. In Proceedings of the *27th Americas Conference on Information Systems (AMCIS)*: Association for Information Systems, pp. 1-10.

Cemiloglu, D., et al. 2020. Towards ethical requirements for addictive technology: The case of online gambling. In Proceedings of the *2020 1st Workshop on Ethics in Requirements Engineering Research and Practice (REthics)*: IEEE, pp. 1-10.

Contag, M., et al. 2017. How they did it: An analysis of emission defeat devices in modern automobiles. In Proceedings of the *2017 IEEE Symposium on Security and Privacy (SP)*: IEEE, pp. 231-250.

Dao, V., et al. 2011. From green to sustainability: Information Technology and an integrated sustainability framework. *The Journal of Strategic Information Systems,* 20(1):63-79.

Darma, J. 2018. The role of top management support in the quality of financial accounting information systems. *Journal of Applied Economic Sciences,* 13(4):1009-1020.

Eltajoury, M. W., et al. 2021. Physicians' Attitudes towards Electronic Prescribing Software: Perceived Benefits and Barriers. In Proceedings of the *International Conference on Data Science, E-learning and Information Systems 2021*, pp. 47-53.

Gholami, R., et al. 2013. Senior managers' perception on green information systems (IS) adoption and environmental performance: Results from a field survey. *Information & management,* 50(7):431-438.

Gotterbarn, D., et al. 2018. Acm code of ethics and professional conduct.

Gregg, D. G., et al. 2001. Understanding the philosophical underpinnings of software engineering research in information systems. *Information systems frontiers,* 3(2):169-183.

Hevner, A. R., et al. 2004. Design science in information systems research. *MIS quarterly*:75-105.

Kirillova, A., et al. 2020. Bioethical and legal issues in 3D bioprinting. *International Journal of Bioprinting,* 6(3).

Kisielnicki, J. and Misiak, A. M. 2017. Effectiveness of agile compared to waterfall implementation methods in IT projects: Analysis based on business intelligence projects. *Foundations of Management,* 9(1):273-286.

Klier, T. and Linn, J. 2016. Comparing us and eu approaches to regulating automotive emissions and fuel economy. *accessed April,* 16:2018.

Kuhse, H. and Singer, P. 1998. What is bioethics? A historical introduction. *A companion to bioethics,* 2:1-3.

Kute, S. S. and Thorat, S. D. 2014. A review on various software development life cycle (SDLC) models. *International Journal of Research in Computer and Communication Technology,* 3(7):778-779.

Leau, Y. B., et al. 2012. Software development life cycle AGILE vs traditional approaches. In Proceedings of the *International Conference on Information and Network Technology*, pp. 162-167.

Leveson, N. G. and Turner, C. S. 1993. An investigation of the Therac-25 accidents. *Computer,* 26(7):18-41.

Mansouri, N. 2016. A case study of Volkswagen unethical practice in diesel emission test. *International Journal of Science and Engineering Applications,* 5(4):211-216.

McNamara, A., et al. 2018. Does ACM's code of ethics change ethical decision making in software development?, In Proceedings of the *Proceedings of the 2018 26th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering,* pp. 729-733.

McQuaid, P. A. 2012. Software disasters—understanding the past, to improve the future. *Journal of Software: Evolution and Process,* 24(5):459-470.

Nunes, R., et al. 2015. Education for values and bioethics. *Springerplus,* 4(1):1-8.

Pelluchon, C. 2016. Taking Vulnerability Seriously: What Does It Change for Bioethics and Politics?, *Human Dignity of the Vulnerable in the Age of Rights*: Springer, pp. 293-312.

Pirkkalainen, H. and Salo, M. 2016. Two decades of the dark side in the information systems basket: Suggesting five areas for future research. In Proceedings of the *24th European Conference on Information Systems*: European Conference on Information Systems, pp. 1-16.

Porrello, A. M. 2012. Death and denial: The failure of the therac-25, a medical linear accelerator. *Death and Denial: The Failure of the THERAC-25, AMedical Linear Accelerator*.

Rogerson, S., et al. 2017. Information systems ethics–challenges and opportunities. *Journal of Information, Communication and Ethics in Society*.

Sever, M. M. and Kağnıcıoğlu, C. H. 2019. A new information system for inventory management in hospitality industry. *İşletme Araştırmaları Dergisi,* 11(1):64-71.

Shmeleva, A. G., et al. 2019. Transport logistics management information system. In Proceedings of the *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*: IEEE, pp. 1471-1473.

Silvis-Cividjian, N. 2021. Awesome bug manifesto: Teaching an engaging and inspiring course on software testing (position paper). In Proceedings of the *2021 Third International Workshop on Software Engineering Education for the Next Generation (SEENG)*: IEEE, pp. 16-20.

Sinha, A. and Das, P. 2021. Agile Methodology Vs. Traditional Waterfall SDLC: A case study on Quality Assurance process in Software Industry. In Proceedings of the *2021 5th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*: IEEE, pp. 1-4.

Turel, O. 2015. An empirical examination of the "vicious cycle" of Facebook addiction. *Journal of Computer Information Systems,* 55(3):83-91.

Usman, A. V. and Ogwueleka, F. N. 2018. SDLC Models as Tools in the Development of MIS: A Study. *IUP Journal of Information Technology,* 14(4).

vom Brocke, J., et al. 2013. Green information systems: Directives for the IS discipline. *Communications of The Association for Information Systems,* 33(1):30.

Zelkowitz, M. V. 2012. What have we learned about software engineering? *Communications of The Acm,* 55(2):38-39.