Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025 ASEE
Paper ID #48493

Escalating Effects of Software Technical and Ethical Debt and Improvement Through Design Discipline and Social Constructs

Mr. James Wessel, Franciscan University of Steubenville (FUS)

James Wessel possesses over 40 years of major software development and acquisition management experience among varied DoD services and industry partners, providing expert consultation in software engineering best practice within major software intensive system program settings. This includes the emergent software engineering areas of the Cloud, Agile, Digital Engineering, DevSecOps and Cybersecurity. James Wessel has been a member of the Carnegie Mellon University Software Engineering Institute (SEI) for over the past 19 years. During this period he also provided software engineering instruction for Franciscan University of Steubenville students for over 9 years. Prior to the SEI, Mr. Wessel provided software system acquisition management support within U.S. Army Program Executive Offices and research programs as a MITRE Corporation employee. Prior to this period he worked within the energy sector. Mr. Wessel holds a MS degree in Software Engineering.

Escalating Effects of Software Technical and Ethical Debt and Improvement Through Design Discipline and Social Constructs

Abstract

As the exponential pace of technology continues to escalate, sadly so has technical debt and its counterpart ethical debt. This paper describes these persistent issues which negatively affect societies at a global level, with a focus upon software engineering design methods to expose and prevent these harmful debts early in the development lifecycle. As the users of software technologies are consequentially affected, social constructs are proposed to provide a feedback loop to both the developer and the consumer communities to better ensure continuous acceptable technical and ethical product quality. A historical review of technical and ethical debt concerns and their relationship is provided, with a focus upon the continuous tension between design rigor and time to market pressures. In terms of impacts, runaway modern day technical debt damages have been estimated at over $2.41 trillion. A recent example was a major global Automotive firm that had a large recall due to a safety related software problem in its cars. The cars were recalled and software updated at great expense, only to realize that the replacement software had not fixed the original problem and had new flaws. Clearly software industry investments to help prevent this type of phenomenon should pay substantial dividends.

The need for a more disciplined, engineering approach to identify and balance (e.g., tradeoff) technical and ethical attributes has been lamented. The proficient application of advanced design quality verification methodologies has shown to improve design quality and thus improved product performance for prioritized qualities. Including ethical quality attributes into the design verification process can reduce the probability of both technical and ethical debt. Software development tool sets can be leveraged to assist in the effectiveness and efficiency of this approach. These design methodologies should be strongly incorporated in education processes, software design guidelines and standards. Lastly the software development and user communities need to better organize awareness of this technology blight. The consumer community, independent agencies and government bodies can mount efforts to put a public spotlight upon these forms of software capability development malpractice resulting the mass distribution of harmful digital products.

Introduction

Historical accounts of technical inventions date back to the era of cave people, along with their shortfalls and misuses. In the modern era, these inventions most often involve complex software technologies that arise almost daily and affect peoples and societies on a global scale in a short amount of time. The design of software systems can have shortfalls, due to either an accidental or purposeful omission of engineering discipline. This is technical debt that needs to be repaired, which can lead to ethical debt and cause dreadful impacts on persons and society. This paper will focus on the subject of software in the domain engineering as software engineering is a relatively recent engineering disciple that has rapidly affected just about every aspect of our lives with software enabled capabilities. A specific focus is software engineering design methodologies to better verify proper quality early in the software development process to prevent technical and ethical debt. Still, anomalies can slip through the best processes. Bringing to bear independent bodies and social constructs that monitor and report software enabled capability performance can provide a vital feedback loop to continually improve quality product performance toward a better experience for peoples and societies. As software professionals are the developers of these capabilities, the education processes must continue to instill and strengthen the latest technical and ethical tenants, standards, methodologies and techniques to better manage the complexities of software enabled capabilities. Industry is a vital stakeholder who needs to prioritize proper operational qualities ahead of immediate monitory and schedule concerns. The relationships among technology, process, people and culture must be understood and managed to effectively and efficiently orchestrate the release of software capabilities that benefit all stakeholders, and minimize issues.

Early Technical Issues and Response Pattern

Toward the end of the nineteenth century homeowners excitedly purchased electric vacuum cleaners whose makers declared its amazing capability e.g., “the Domestic Cyclone” [1]. These early vacuum cleaners had fundamental engineering power management design and implementation faults (e.g., exposed metal parts, poorly insulated wiring, lack of grounding and safety features) that literally shocked (electrically) many users. This naturally resulted in a level of individual and public social distress, which in turn initiated public pressure for safety improvement. In response to this, and other alarming public technology product safety concerns, Underwriters Laboratories (UL) an independent agency for product assessment was created. UL professional engineers would study a product’s design regarding varied performance qualities with specific attention to safety, devised tests, and upon successful assessment, grant the famous UL approval certification. Standards for electric device safety were initiated and public trust in vacuum cleaners was significantly regained. Also, a cause-response pattern emerged of technical harm, public awareness and pressure on industry to improve, independent agency assessment and pronouncement, and better technical standards were instituted. The effect was that the vacuum industry improved its product design and operations. Today vacuum cleaner performance remains much safer, though not perfectly so [2].

Modern Technology and Ethical Quality Issues

Fast forward to current times, the Zoom video conference capability rapidly releases the Zoom app, advertising superior service at a great bargain, and users immediately adopt, trust and use this social communication technology. However, one day a church group using Zoom for a presentation was shocked when their session was hacked (aka Zoom bombing, a type of cybersecurity flaw) and inappropriate matter was broadcast, to the horror of church attendees [3]. The Zoom CEO was later interviewed, he stated in the rush to push their capability to the market, not enough security was designed into the technology [4], thus accruing a level of security and privacy issues that needed to be addressed (technical debt). In this case and many others, the flaw could be seen as an issue with quality of programming, quality of the review process, or perhaps it could even be debated as an ethical issue related to rushing a product to market before adequate test and evaluation.

Perhaps more alarming, Whitney Phillips, an assistant professor at Syracuse University suggests “When you see this kind of rampant abuse, it isn’t just a one-off thing,” [5]. This statement was prophetic as following years experienced several highly visible cyber security flaws yielding dramatic effects, a sampling include [27]: SolarWinds Orion, resulting damages estimated at $90 million (2020-21); T-Mobile data breach affected 50 million customers (2021); Colonial Pipeline’s ransomware attack resulted in a ransom of $5 million (2020-21); Tesla vehicle recall, affected 12,000 vehicles (2021); Log4j software bug that some estimate the worst security vulnerability in years (2021-22); and TX Crypto Hack that resulted in $473 stolen (2022).

In many cases, a cybersecurity flaw and resultant issue was the result of a decision with ethical components: the choice to deliver a product quickly instead of investing in more rigorous product quality verification practices. In the zoom bombing case, many users globally incurred long lasting disturbing personal effects or ethical debt (people cannot 'unsee' unwanted inappropriate materials). Unfortunately, this type of story is too common as the above examples show and Phillips stated, due in part to suboptimal attention to technical and ethical capability quality design.

Engineering Aspects to Understand

The above examples exhibit suboptimal management of modern technology that have led to negative outcomes for both consumers and developers. These outcomes are often associated to choices made within the engineering aspects of: technology, methodology (or process), people, culture [6], and drive to market. The perennial challenge for the developers of technology to deploy products that meet both business, technical and ethical objectives (tradeoff analysis). This is one of the greatest challenges of our day given consumer demand signal for technological advancement and the consequences poor technology releases to our global economy and peoples. To devise an effective improvement approach, it is worthwhile to understand and consider these aspects of engineering toward a holistic ethical improvement approach.

Technology in recent decades has exhibited rapid and almost unbridled growth of any sector [7]. This has produced great capabilities to bring us to the moon, perform precise eye surgery, facilitate education in desolate geographies, and much more. In modern times the scope of effects caused by modern technology include not only people, but just about every facet of our global society. This is expressed well in the Catholic Church' Pontifical Council for Social Communications publication *AETATIS NOVAE* "Today's revolution in social communications involves a fundamental reshaping of the elements by which people comprehend the world about them and verify and express what they comprehend. The constant availability of images and ideas, and their rapid transmission even from continent to continent, have profound consequences, both positive and negative, for the psychological, moral and social development of persons, the structure and functioning of societies, intercultural communications, and the perception and transmission of values, world views, ideologies, and religious beliefs" [8]. With respect to the negative effects, "technology can bring out our worst behaviors. Social media platforms can serve us content that enrages or depresses us…These platforms also can be used by bad actors to take immoral actions more easily" [9]. In short, we find ourselves on the back of an untamed technology-enabled tiger, without a proper bridle to more properly control, some would say.

Good engineering for decades has utilized process and methodology practices to instill repeatable rigor toward the development of products that behave in consistent fashion. In general, an overall methodology objective is to help enable the delivery of target technological capabilities on time, on budget, to prioritize quality goals. Good engineering methodology has the capacity to enable success, an example is how methodology literally transformed the automotive industry. Japanese car makers prior to the 1970s invested in continuous process improvement heralded in part by Mr. Deming to better design quality cars, for less capital, and in less time [10].

People (and the human condition) are a constant aspect of engineering, including our creative ideation ability for new products (e.g., better mouse traps and spaceships), desire for business success, and profit. This includes our imperfect human condition and proclivity for mistakes. An early technology pioneer Blaise Pascal offered his concern of technology given his observations of the human condition, stating issues emanating from inventions will unavoidably occur [11]. Pascal had a dual view of human nature, holding both a high and low regard of human nature. Today a similar sentiment used is 'trust but verify'.

People are the alpha and omega with respect to virtue improvement as the agents for technology design and its eventual users. Interestingly, designers and developers are a relatively small percentage of the workforce (e.g., computer scientists, software engineers) who are responsible for the creation of digital capabilities that flood the internet marketplace. "Computing professionals' actions change the world" [12]. As such engineers are a primary stakeholder to influence product quality improvement. The design methods they employ greatly affect the quality of experience for the users of delivered apps globally.

Culture is "An integrated pattern of human behavior that includes thoughts, communications, languages, practices, beliefs, values, customs, courtesies, rituals, manners of interacting, roles, relationships, and expected behaviors of a racial, ethnic, religious or social group; the ability to transmit the above to succeeding generations; is dynamic in nature." [13]. The digital culture "is the relationship between people and the use of technology. It describes how digital media and technology have shaped our everyday lives and interactions with society, people and work." [14]. Values are a "culture's standard for discerning what is good and just in society. Values are deeply embedded and critical for transmitting and teaching a culture's beliefs. Beliefs are the tenets or convictions that people hold to be true" [15]. Today, there seems to be a great culture tension among the deliverers of technology products and the user community in terms of quality of experience. Efforts to help the business community better understand the economic advantages of providing high quality, ethically orientated software products can reduce this tension toward better product quality.

Technical and Ethical Debt

The Technical debt (TD) metaphor is used to describe the long-term consequences of engineering decisions made to achieve a short-term benefit [16]. It can occur and accrue when good engineering discipline does not occur, or an unwitting tradeoff is made for perceived short term business profit goals. Often business stresses commonly delay the full completion of desired product quality attributes (QA). Management may intend to eventually fix known deficiencies, but this intent is often overcome by seemingly more urgent needs. This suboptimal practice of engineering design tends to become entrenched and normalized within the engineering culture over time. To quote Mark Zuckerberg "move fast and break things".

Software maintenance and sustainment activities encompass a wide variety of software related change vectors that can affect performance quality to include tooling changes, coding changes, algorithm changes, procedure changes, and documents changes. Software engineering discipline, to include design best practice, must be maintained in this phase of the software lifecycle to ensure quality integrity and thus prevent technical and ethical debt. The pressure to push updates or fixes into production is typically high, discipline in maintaining a good quality validation process is important. A recent example was a major global Automotive firm that had a large maintenance related recall due to a safety related software problem in its cars. The cars were recalled and software updated at great expense, only to realize that the replacement software had not fixed the original problem and had new flaws. [33]

A current popular technology engineering development approach is the Agile methodology. An intent of Agile is the delivery of working products quickly to satisfy the user demand signal. An undisciplined implementation of this method can prioritize speed of capability delivery over product quality assurance. The result can be a bow wave of technical debt over time, along with diminishing product quality and increased costs as shown in Figure 1.

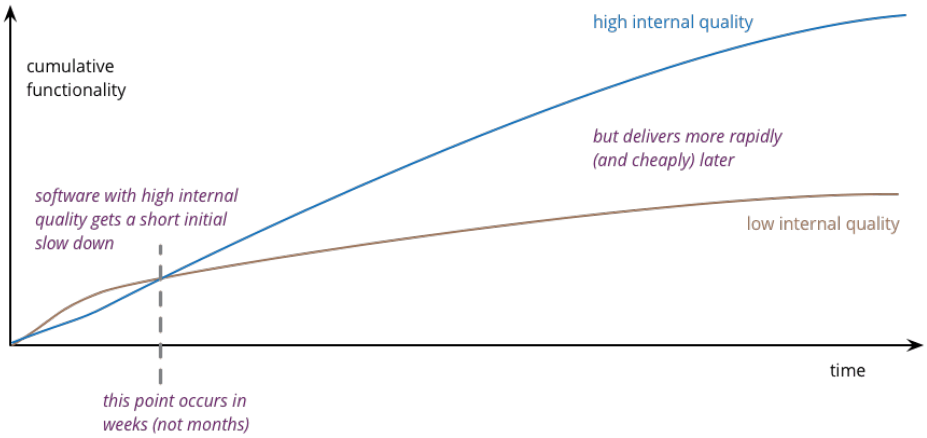

Figure 1 – Software Quality Curve [28]

The rapid advance of the use of Artificial Intelligence (AI) and Machine Learning (ML) during the design and development stage of products can create issues [34]. The concept of speed vs. quality vs. ethics of using machine generated designs is discussed along with the concept of inventorship of intellectual property. The speed and ubiquitous low cost of AI/ML may indeed make Agile programming obsolete and add yet another (the total lack of human content) layer to this complex issue of technical debt.

Ethical debt is the presence of poor ethics related product quality e.g., safety, health, and welfare of the public, due to failure to comply with good software design (as software is the focus of this paper) and development practices [17]. When the design process ignores or minimizes effort to address ethical design attributes, this debt begins to occur. Ethical debt is often related to or result of TD. For example, the lack of Zoom technical security led to unsuspecting audience members incurring poor content against their will, whose ethical negative effects may still be present. When AI/ML is involved, its information sources may be biased to a significant degree and amplified within AI/ML results [29]. AI/ML can generate products from sources without appropriate permission, additionally it has been known to produce inappropriate responses [30].

Ethical debt is often observable upon incurring TD. In the Zoom bombing case, the technical security shortfall led to ethical debt. In terms of cost, significant data exists with respect to TD however the cost is less deterministic regarding ethical debt. How can one put a price on the harm caused by exposing young children to extreme pornography or violence? In the Consortium for Information and Software Quality (CISQ) document ‘The Cost of Poor Software Quality in the US: A 20220 Report’, “the cost of poor software quality in the US has grown to at least $2.41

trillion”. The report goes on to state “by 2025, 40% of IT budgets will be spent simply maintaining TD, and it’s a primary reason that many modernization projects fail”.

In short, the software industry is drowning in TD, and by extension one can surmise ethical debt is in turn dramatically rising. Hence the quality of our global digital culture is degrading on several levels, a sobering unhealthy trend.

A number of improvement recommendations have been posed by CISQ, which revolve around a DevQualOps Model [27]. Areas for improvement include: 1. Quality standards/software problem taxonomies, 2. Tools for understanding, finding and fixing deficiencies/TD, and 3. AI / ML tools for software engineering. An intent of this paper is to bring to the forefront architectural evaluation methodologies that can provide necessary software design rigor to improve software quality and thus reduce our compounding TD issues, and by extension ethical debt. This can encompass these CISQ recommendations for improvement.

Initial Engineering Ethics Improvement Efforts

The ethical issues resulting from ill designed and implemented technology capability that affect our daily life experience, on a global scale, is well recognized and lamented [18]. Good efforts to improve ethics in the engineering domain have occurred over recent years. Standards and codes of software technology ethics have been developed and published [12], engineering ethics course work have been proposed and implemented in many university engineering programs e.g., University of Denver, Wrexham University, Stony Brook University.

The improvement initiatives are good. However, a cursory observation of current significant sub-optimal ethical conditions easily encountered on the internet point to the need for additional, more rigorous engineering practices to improve the technical and ethical behaviors of software technologies.

Software Design Quality Improvement

Design is a central element of software capability development, often called the blueprint for building technology capability. As poor-quality blueprints can result in inferior home structures, immature software design can result in sub-optimal app performance. Recent years have seen an increased demand signal (public distress) for ethical design improvement [20]. The need for a more disciplined, engineering approach to identify and balance (e.g., tradeoff) technical and ethical attributes has been lamented “it is certainly uncontroversial that the goal of software design should be a product that is efficiently optimizing the [ethical] values we wanted to consider, it is by no means clear or obvious which point on the line, that is, which one of the many possible trade-offs, should be implemented” [21]. Additionally, ”ethics in software systems is hardly ever considered as a first-class entity, its support of, or hindrance to, values of users and society is usually neglected” [19]. Clearly an increase of engineering design rigor is vital for an improved software product user experience, an approach to accomplish this follows.

A core tenet and driver of engineering design are QA, the non-functional attributes of a product. If the functions of a product are all that counted, a car design would focus upon the simple functions of movement, braking and steering. Instead, it’s the QA of a car e.g., safety, performance, security that significantly matter to buyers and thus are priority car design factors. The identification and management of QA are fundamental to a rigorous software capability design.

Toward this end, the addition of ethics-based QA [32] with traditional QA [31] should occur to better ensure ethical engineering design. This begins with the requirements management process where a champion for ethical engineering attributes is a key stakeholder. The requirements process should include ethical design factors to better promote a more positive product user experience.

Design Quality Improvement Methodologies

In the automobile industry nascent years, cars were generally considered safe, however car accident fatalities continually increased, along with public distress about safety. Eventually engineers designed a component to improve safety, the seat belt, tested their designs and measured improvements. Who can forget seeing car safety commercials that destroyed cars containing dummies as a public awareness campaign. Analogously, software quality must be identifiable in design (e.g., the seat belt in a car), varied use cases utilized to ascertain performance (e.g., car hitting varied objects) and metrics made (e.g., damage incurred to dummy driver) for improvement analysis. This is not a difficult concept, however cost and schedule constraints pressure organizations to sometimes avoid quality improvement activities. As with all investments, a cost-benefit tradeoff should occur properly to gauge quality improvement efforts. Software standards should minimally prescribe a reasonable level of quality improvement activity to pass subject inspection. The user community should also be involved, which will be discussed later in this paper.

Technical design methodologies have been available for decades with documented publications that attest to their effectiveness and efficiency, to assist engineers ‘bake in’ desired QA into a system’s design. The intent is to prevent technical and ethical debt from starting. Prime methodology examples are the Quality Attribute Workshop (QAW) [22], and the Architecture Analysis Methodology (ATAM) [23]. These provide disciplined, engineered and vetted avenues to increase the probability of a quality product.

The QAW is a facilitated method to solicit priority QA with project stakeholders, attention is focused upon developer organization mission intent, business strategic and tactical project objectives. This approach provides a sound basis to identify driving QA for a desired software capability. QA are most effectively characterized using scenarios, a core QAW activity. Each scenario has a stimulus, an environment, and a response measure. The results of the QAW include the set of priority scenarios and respective challenges.

This information is used to refine requirements, evaluate and improve the design, and document the design in greater detail. The overall benefits of conducting a QAW are clarified QA, increased stakeholder engagement and communication, improved basis for design decisions, better design documents and support for analysis and test throughout the system life cycle. By including ethical QA within a QAW, the product will likewise inherit the benefits of improved ethical design analysis and performance accountability.

The purpose of an ATAM is to assess the consequences of design decisions in light of identified QA and business goals, and to identify associated risks. An intent is to find trends e.g., correlations between architectural decisions and predictions of system properties. Use case scenarios are used to understand stakeholder priorities and associated QA. A team of evaluators and stakeholders then examine every aspect of each scenario to include risks, sensitivity points, and tradeoffs. A risk is a potentially problematic design decision, and a tradeoff is a property that affects more than one QA, for example performance and security in the context ‘Changing the level of encryption could have a significant impact on both security and performance’. The ATAM is a demonstrated approach to evaluate a capability design to identify the consequences, good or bad, of design decisions. These decisions can be recorded in design artifacts, along with their rationale, to help justify why a proposed design promotes priority capability QA. This increases the likelihood that the delivered capability will meet quality requirements and thus reduce the likelihood of technical and ethical debt.

Software Design Patterns

Software design patterns [24] are design constructs for commonly occurring design challenges. They allow software designers to create more flexible and ultimately reusable designs without having to rediscover the design solutions themselves. This allows the design process to be more expedient, consistent and reliable and as such are popularly used by designers. They are often characterized by certain QA, designers who target certain capability attributes in an application will use a corresponding behaving design pattern. This presents an opportunity for design patterns to be classified by their ethical performance qualities as well. Developers can take advantage of patterns known to provide desired ethical outcomes. A catalog of design pattern ethical attributes can allow developers to avoid known non-ethical patterns [25, 26].

Useful Social Constructs

The use of more rigorous design methodologies alone will not likely prevent all technical and ethical design and implementation issues for the simple reasons of (lack of adequate) time and budget. Additionally, the software system environment is typically in flux due to perturbations in hardware, operating system, and cybersecurity updates for example. The expanded and more focused use of social processes that encourage and assist the production of ethically designed and performing products should be addressed to accompany good engineering discipline.

- The development community can improve the socialization of good ethical design intent within their products using communications and feedback loops

Design and development communities can use various tools, such as a company web site that advertises desired product ethical QA, ethical design/architecture features and ethical test scenarios with test results to push the quality and ethics agenda and learning. This can provide a performance feedback loop between developing organizations and users for on-going product improvement. As discussed, an example of this approach occurred around the 1950s and 1960s in the automotive industry with car safety and seat belts with better awareness advertising. This quickly saw the advent and soon full adoption of seat belts, with the help of government policy, in a few years. Soon seat belts became a hallmark car design feature and saved many lives.

Independent verification organizations can provide and publish the results from a review of a product’s ethics-based performance, much like current day consumer report car model reviews. Car reviews have a safety rating that consumers consider in making purchase decisions. Including an ethical design and performance rating for apps for instance, will better inform consumers and motivate developers to ensure superior technical and ethical product performance.

Social user groups can solicit app user experience to further monitor and ensure proper ethical performance. Apps with a good ethics performance history can be prominently advertised to enable better consumer download decisions. This occurs in the travel industry where, for instance, well-performing hotels are published in travel books, and remain there as long as users continue to verify a good hotel experience. This is another vector to help ensure technical and ethical product performance.

Lastly, technology developers can be placed in precarious situations where an organization presses an unethical agenda upon engineer designers and implementors. A resource can be made available to provide assistance in making the proper technical and ethical decisions in terms of guidance and even legal assistance, e.g., a hotline to anonymously request support. The resources to provide this type of service can be sourced from major capability providers who ‘sign on’ to help make the world’s internet experience more technically and ethically responsible.

Education

As the exponential pace of technology continues to escalate, the software education process must be agile to continually absorb these updates to include advanced design quality verification rigors. The effort to keep curriculums current is the challenge of every software related education program. At a similar level, best practices and lessons learned of technical and ethical debt in parallel must be imparted throughout the education process. This must not be limited to a single class during the senior year of a program. Forming ethical character is a process that takes longer than one course. Core technology classes should provide a level of ethical tenets, for example, that are built into exercises and software projects. This will be a challenge to fit into technical programs that are already stressed to accommodate existing high levels of learning. The value of improved ethics is a powerful driver to accomplish this task.

Continual learning is an important element in industry to keep staff skills current. This process should also consider the subjects of technical and ethical debt. The current condition of TD in terms of cost [28], and the long-term benefits in improved quality design should be a motivating factor.

Conclusion

History has long demonstrated the consequences of suboptimal attention to ethics in technical capability design. Imperfect product design is part of the human condition, no one is perfect. Complex development technologies and products have increased the difficulty in managing product design in terms of their technical and ethical impacts on individuals and society. As technology has greatly grown in capability power, so have its consequential ethical effects, on a global scale. Many positive efforts have occurred to stem the tide of subpar technical and ethical effects, to include ethical engineering venues, standards, guidance and education. These efforts, as good as they are, have not produced the desired effects.

Good disciplined, engineering design and verification methodologies e.g., QAW and ATAM have been developed and shown to help ensure desired capability technology and ethical quality are present within a capability. These types of methods should be promoted in good software engineering guidance e.g., Software Engineering Body of Knowledge (SWEBOK) and included in software engineering education programs. This will help to ensure their inclusion in the design and verification process.

Many social constructs are available to further aid and encourage the adoption of good ethical engineering design and performance. A software ‘UL’ can be an independent assessor that regularly publishes its results. The user community can also participate through venues that aggregate feedback to the development community for improved technology and ethical product performance. Major app developers have the resources to lead the way for ethical design and verification improvement to better enable a more ethical internet experience by all persons, and global cultures.

Acknowledgements

The author wishes to acknowledge the contributions of Mr. John Cogliandro and Mr. Bryce Meyer for their expert advice and review of this document. Their academic and professional industry experience and acumen provided significant benefits throughout the writing of this paper.

References

[1] “Invention and History of Vacuum Cleaners,” vacuumcleanerhistory.com. https://www.vacuumcleanerhistory.com/ (accessed Nov. 3, 2024)

[2] United States Consumer Product Safety Commission, “Shark Vacuum Cleaners Recalled by Euro-Pro,” cpsc.gov. https://www.cpsc.gov/Recalls/2015/Shark-Vacuum-Cleaners-Recalled-by-Euro-Pro (accessed Nov. 3, 2024)

[3] A. Khan, “Zoombombing' comes for houses of worship.” religionnews.com. https://religionnews.com/2020/03/30/zoombombing-epidemic-comes-for-houses-of-worship/ (accessed Nov. 3, 2024)

[4] CNN, “Zoom CEO addresses 'Zoombombing:' We had some missteps.” yahoo.com https://www.youtube.com/watch?v=xk992LJ4N9M (accessed Nov 3, 2024)

[5] T Lorenz and D Alba, “Zoombombing Becomes a Dangerous Organized Effort.” nytimes.com. https://www.nytimes.com/2020/04/03/technology/zoom-harassment-abuse-racism-fbi-warning.html (accessed Nov. 3, 2024)

[6] P Buttles-Valdez, S Agapi, and F Valdez, *A Holistic Approach to Process Improvement Using the People CMM and the CMMI-DEV: Technology, Process, People, and Culture, the Holistic Quadripartite* Software Engineering Process Group 2008, March 2008, Tampa, Florida.

[7] Peter J. Denning and Ted G. Lewis, “Exponential laws of computing growth,” *Commun. ACM,* vol. 60, no. 1, January 2017. [Online]. Available: https://doi.org/10.1145/2976758. [accessed Nov. 3, 2024].

[8] “Pastoral Instruction Aetatis Novae on Social Communications on the twentieth anniversary of Communio et Progressio,” www.vatican.va https://www.vatican.va/roman_curia/pontifical_councils/pccs/documents/rc_pc_pccs_doc_22021992_aetatis_en.html (accessed Nov. 3, 2024)

[9] L. Kugler, “Technology's impact on morality,”. *Commun. ACM,* vol. 65, no. 4, April 2022, p.15–16. Available: https://cacm.acm.org/news/technologys-impact-on-morality/. [accessed Nov. 3, 2024].

[10] M. Cusumano, “Manufacturing Innovation: Lessons from the Japanese Auto Industry,*”* *MIT Sloan Management Review,* Fall 1988. [Online]. Available: https://sloanreview.mit.edu/article/manufacturing-innovation-lessons-from-the-japanese-auto-industry/. [Accessed Nov. 3, 2024].

[11] B. Pascal, *Pascal’s Pensées*. New York: E.P. Dutton, 1958.

[12] J Gogoll, N Zuber, S Kacianka, et al., “Ethics in the Software Development Process: from Codes of Conduct to Ethical Deliberation.” *Philos. Technol.,* vol. 34, p. 1085–1108, April 2021. [Online]. Available: https://doi.org/10.1007/s13347-021-00451-w [Accessed Nov. 5, 2024].

[13] National Center for Culture Competence, *“*Bridging the Cultural Divide in Health Care Settings,” nccc.georgetown.edu. https://nccc.georgetown.edu/culturalbroker/8_Definitions/index.html (accessed Nov. 3, 2024).

[14] Digital Society, “What is Digital Culture,” digitalsociology.org.uk. https://digitalsociology.org.uk/digital-culture/ (accessed Nov. 3, 2024).

[15] T. Rothschild, *“*Rothschild's Introduction to Sociology.” rwu.pressbooks.pub, https://rwu.pressbooks.pub/rothschildsintrotosociology/ (accessed Nov. 5, 2024).

[16] H. Kleinwaks, A. Batchelor and T. H. Bradley, "Ontology for Technical Debt in Systems Engineering," *IEEE Open Journal of Systems Engineering*, vol. 1, pp. 111-122, Sep. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10254240. [Accessed Nov. 7, 2024].

[17] IEEE Policies, “IEEE Code of Ethics,” ieee.org. https://www.ieee.org/about/corporate/governance/p7-8.html (accessed Nov 10, 2024).

[18] B. Johnson and J. Smith, “Towards Ethical Data-Driven Software: Filling the Gaps in Ethics Research & Practice,” in *IEEE/ACM 2nd International Workshop on Ethics in Software Engineering Research and Practice (Sethics), Madrid, Spain, 2021,* pp. 18-25

[19] Vallor, Shannon and Arvind Narayanan. “An Introduction to Software Engineering Ethics.” scu.edu. https://www.scu.edu/media/ethics-center/technology-ethics/Students.pdf (accessed Nov 10, 2024).

[20] I Ozkaya, “Ethics Is a Software Design Concern,” *IEEE Software,* vol. 36, pp.4-8, May-Jun. 2019. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8693077. [Accessed Nov 10, 2024].

[21] Gogoll, J., Zuber, N., Kacianka, S. et al., “Ethics in the Software Development Process: from Codes of Conduct to Ethical Deliberation,” *Philos. Technol*., vol. 34, pp.1085–1108, April 2021. [Online]. Available: https://doi.org/10.1007/s13347-021-00451-w [Accessed Nov 10, 2024].

[22] M. Barbacci, R. Ellison, A. Lattanze, J. Stafford, C. Weinstock, and W. Wood, “Quality Attribute Workshops (QAWs),” Carnegie Mellon Software Engineering Institute, Technical Report CMU/SEI-2003-TR-016, Aug. 2003.

[23] R Kazman, M Klein, and P Clements, “ATAM: Method for Architecture Evaluation,” Carnegie Mellon Software Engineering Institute, Technical Report MU/SEI-2000-TR-004, Aug. 2000.

[24] E Gamma, R Helm, R Johnson, and J Vlissides, *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., 1994.

[25] C. Nguyen, “X7 Dark Patterns in UX Design: A Guide To Ethical Design.” uxplaybook.org. https://uxplaybook.org/articles/ux-dark-patterns-and-ethical-design (accessed Nov 10, 2024).

[26] Jerome X, "Ethical vs Dark Patterns in Design: Navigating the Fine Line." medium.com. https://medium.com/@jeromexjamals06/ethical-vs-dark-patterns-in-design-navigating-the-fine-line-eec98c475b10 (accessed Nov 10, 2024).

[27] CISQ Report, "The Cost of Poor Software Quality in the US: A 2022 Report." it-cisq.org. https://www.it-cisq.org/wp-content/uploads/sites/6/2022/11/CPSQ-Report-Nov-22-2.pdf (accessed Nov 10, 2024).

[28] Falcon, "How to Deal with Technical Debt: strategies and tips." Dev.to. https://dev.to/gelopfalcon/how-to-deal-with-technical-debt-strategies-and-tips-5blh (accessed Nov 10, 2024).

[29] C. Petrozzino, "Who pays for ethical debt in AI?." link.springer.com. https://link.springer.com/article/10.1007/s43681-020-00030-3 (accessed Nov 10, 2024).

[30] International Organization for Standardization (ISO), "Building a responsible AI: How to manage the AI ethics debate," iso.org. https://www.iso.org/artificial-intelligence/responsible-ai-ethics (accessed Nov 10, 2024).

[31] Codoid, "The Basics of Software Quality Attributes," codoid.com. https://codoid.com/software-testing/the-basics-of-software-quality-attributes/ (accessed Nov 10, 2024).

[32] X- Team, "Ethical Issues in Software Development," team.com. https://x-team.com/magazine/ethical-issues-in-software-development#:~:text=At%20its%20core%2C%20ethics%20in,and%20fosters%20a%20sustainable%20future (accessed Nov 10, 2024).

[33] L. Bell, "Ford Recalling 34,000 Bronco Sport, Escape SUVs Over Unresolved Issue from Previous Recall." roadandtrack.com https://www.roadandtrack.com/news/a64352733/ford-second-recall-34000-bronco-sport-escape-suvs-fire-risk/ (accessed Nov 10, 2024).

[34] J. Cogliandro, "IP law is dead: The perils of using AI in invention," *ISE Magazine*, vol. 57, no. 3, March 2025. [Online]. Available: http://cie.ed.asu.edu/volume6/number12/. [Accessed Apr. 2, 2025]