Non-Functional Requirements Orienting the Development of Socially Responsible Software

Luiz Marcio Cysneiros[1] and Julio Cesar Sampaio do Prado Leite[2](✉)

[1] School of Information Technology, York University, Toronto, ON, Canada
cysneiro@yorku.ca
[2] Departamento de Informática, PUC-Rio, Rio de Janeiro, Brazil
julio@inf.puc-rio.br

Abstract. Nowadays, software is ubiquitous and present in almost everything we buy and use. Artificial intelligence (AI) is becoming prevalent in software products. The use of AI entices consumer inquisitiveness, promising software products that can make our lives easier, productive, and in some mission-critical applications safer. Similar reasoning can be applied to systems exploring Internet of Things, cloud services, and mobile technologies. However, there is a trust deficit when it comes to accepting AI as well as the other above-mentioned features, as a reliable technology platform. This paper argues that the more critical the domain is, the less consumers seem to trust software to make decisions on their behalf or even to be used. Aspects such as safety, privacy, and ethics challenges the perception of trustworthy computing. In the past two decades, several works have suggested that Corporate Social Responsibility (CSR) may play an essential role in creating a trust paradigm between customers and businesses promoting loyalty, customer retention and thus enhancing customer trust and increasing corporate profit. We believe that the software industry will need soon rather than later to encourage trust in their embedded software. A promising approach lies in adapting principles associated with CSR to guide the software development processes. Such an approach could help to achieve two goals: Deliver trustworthy software and, if desired, deliver socially responsible software. We believe that Non-Functional Requirements (NFR) will play a crucial role in this endeavor. This paper highlights a first approach to establishing a basic set of NFRs that should always be carefully considered when developing software, as to aim socially responsible software.

Keywords: Socially responsible software · Non-Functional Requirements · Trust · Transparency · Ethics

1 Introduction

Today, software is embedded in almost everything we buy or use daily in our lives. In recent years, AI has been increasingly used to deliver solutions in many different commercial and regulatory domains, from personal assistance devices such as Alexa[1]

[1] https://developer.amazon.com/en-US/alexa.

S. Nurcan et al. (Eds.): BPMDS 2020/EMMSAD 2020, LNBIP 387, pp. 335–342, 2020.
https://doi.org/10.1007/978-3-030-49418-6_23

to face recognition technologies used by law enforcement agencies. However, the use of AI raises doubts in the mind of consumers regarding how much we can trust AI[2] to make decisions on our behalf. The lack of trust seems to be more prevalent in mission-critical systems where personal safety is in the care of the machine. Kolm shows [1] that 70% of Canadians are comfortable with AI scheduling appointments, but only 39% feel comfortable with AI piloting autonomous vehicles. Therefore, we believe that the software development process needs to address ways to assure consumers they can trust the software embedded in the products they are buying and/or using.

Applications utilizing concepts related to Internet of Things, cloud services, and mobile technologies will raise similar concerns aggravated with the expectation of privacy and safety, triggering ethical questions that will directly impact how much customers can trust their devices. Although there are works [2, 3] tackling trust related to machine learning and decision support systems, they look at trust in a single dimension and do not capture the consequences of trust from a social perspective.

Our work aims to consider trust of AI-based software from a citizen's viewpoint, using the metaphor of Corporate Social Responsibility (CSR)[3]. In the past two decades, many works have pointed out that CSR goes a long way in promoting positive outcomes such as loyalty, repeat business, and purchase intention [4, 5]. Furthermore, CSR efforts may also positively impact the market value of companies that are perceived to be committed to social responsibility [6]. One important aspect of CSR is that its adoption reduces information asymmetry [7], and as such, it brings out transparency. It is to note the tangling effect of CSR in the broader concept of Corporate/Company Reputation [30].

One of the main reasons for consumers to value CSR is because it promotes intrinsic trust in the company. One way of looking at trust is to measure how much a consumer thinks a company can be deemed reliable in situations entailing risk to the consumer. One critical factor is how much consumers believe the company's actions and behaviors have the consumers' interest and welfare in mind [9].

This work builds an initial argumentation of why using CSR knowledge does help software engineers develop trustful Software. The benefits of using CSR concepts would be twofold i) Develop trustworthy systems that would help to retain customers and to increase market share ii) Use this trustworthy as the basis for developing a socially responsible system that is likely to be in high demand in the near future.

In this idea paper, we present the foundation for our ongoing work. We are tackling what we believe will be the core requirements to deliver trustworthy software. We associate these requirements with the perspective of society, in general, to be able to opt for socially responsible software together with the goal of repetitive business and improving market share. We hope to inspire other researchers to explore similar paths.

[2] https://www.nytimes.com/2020/01/18/technology/clearview-privacy-facial-recognition.html.

[3] The Business Domain debates over the similitudes/differences among the acronyms CS (Corporate Sustainability) and CSR. We side with those that consider CS and CSR as synonyms.

2 Method

We carried out a brief literature review starting with the CSR domain to investigate the qualitative properties used in their experience to promote corporate social responsibility that could be adapted to the software domain. We used keyword searches such as (csr AND trust), (corporate social responsibility AND Trust) from 2015 to 2019. We recovered 112 publications. After removing duplicated references, snowballing, and examining the abstracts, we reviewed 22 references. Our choice was based on the linkage of CSR and Trust.

We elicited knowledge from these references to create a matrix with the most often mentioned *properties* expected to be present in companies adopting the CSR approach. Following, we analyzed which of those *properties* should be implemented in company's Information Systems, in general, to contribute to a trustworthy and socially responsible environment. Using an NFR (Non-Functional Requirements) perspective [21], we found three NFRs: **Trust**, **Ethics,** and **Transparency** in these *properties*. Using earlier knowledge [22, 27], we searched for NFRs which may interact either positively or negatively with these three NFRs, and we elicited **Privacy**, **Safety,** and **Security**.

Our aim was not to build software to support CSR adoption by companies. We studied the CSR approach as a basis to propose which would be the critical qualitative *properties* to develop trustworthy Information Systems software, based on the company's goal. As such, these qualities would stand as essential NFRs to develop software to support the delivery of socially responsible Information Systems either as a goal by its own or as part of the adoption of CSR. At the same time, companies that are interested on a stricter approach to CSR will also have a solid base to start from.

3 Results

As pointed out by Vlachos et al. [10] and others, trust is central for companies to be perceived as socially responsible. By the same token, we set trust as the primary NFR to be satisfied, i.e., satisfied within acceptable limits. Our reasoning follows the idea that safety-critical systems and any software using AI as well as advanced forms of technology such as internet of things and cloud systems, will inherently trigger fear in many customers who are forced to relinquish their safety to a machine or to face unwanted misuse of their behaviors and preferences and sometimes both

In corporate domains, trust would come when providers demonstrate ethical behavior enforced in their software. Bowen illustrates such a scenario for safety-critical systems [11]. In order to promote trust, software engineers need to take a bottom-up approach in developing domain-specific knowledge of elements that build the foundations of trust.

Trust is also frequently linked to the concept of ethics. Consumers tend to trust companies that they perceive as ethically sound [12]. Nevertheless, consumers also identify and believe that companies are following moral standards if they are transparent in the way they do business [6]. The ISO 26000 standard points out how businesses and organizations should handle ethical and transparent concerns to act responsibly [13]. We believe that a similar perspective could be applied to software development in general.

Ethics helps to promote trust [5, 14], together with the understanding that safety-critical systems need to demonstrate ethical behavior to be accepted by consumers.

However, Ethics is not a straightforward concept that can be easily applied. It is a human compass that shifts with the sand of time. Researchers studying the self-driving car domain have pointed out many ethical dilemmas that need to be addressed [15]. Furthermore, ethical decisions made either by the software or its users choosing an ethical scenario may entail legal implications [15]. The first step to building ethics into a machine is to recognize that all models will have their faults and limitations. There can be no single software architecture which proclaims to be the benchmark of moral goodness. Since human interest cannot always be satisficed due to inherent conflicts, ethics will likely be aggressively contested in the court of law between those who seek to exploit AI technology in the public domain and those who wish to confine it to a controlled environment. Insurance companies will likely use such types of arguments in many cases when AI making decisions are involved. Software developers should be prepared to counter-argument any accusations or even better, be pro-active, and not give space to them.

Even for systems that are more benign in terms of damage, such as Alexa, ethical behavior can become a fundamental requirement for retaining customers, since its use may threaten privacy. Another critical requirement is privacy. Customers are rediscovering the importance of privacy [16], and states are enacting privacy laws, as the General Data Protection Regulation (GDPR). There have been many breaches of privacy such as the Samsung voice-recognition being activated by default in smart television models to Alexa capturing conversations and sending them to a repository [17].

Additionally, safety will play an essential role in many different domains. Software, AI based, will demand safety to be mandatory. Smart homes, for example, can bring threats to safety if not extensively analyzed when dealing with heating systems. It also should avoid exposing residents to external monitoring. Huang et al. survey points out most essential qualities to be satisficed [18] as to attain safety.

Systems like smart home administration can attract consumers with their convenience features, together with the expectations of saving money on things such as energy control. However, they may open the door for hackers to exploit vulnerabilities and misuse information. A recent study carried out in England shows the industry providing smart home solutions has not yet offered consumers with enough evidence they can trust that their home solutions will comply with the security and privacy standards they expect [19]. Poorly managing security and privacy can expose vendors to *legal disputes*, which is the price to pay by not observing social responsibility.

We advocate that software engineers should start investigating how to operationalize[4] Ethics, Safety, Security, and Privacy requirements. They can either carry out literature reviews to identify possible operationalizations for each NFR or use existing knowledge bases that capture knowledge that can be reused [20]. One of our goals for future work is to produce or improve existing knowledge bases representing them in the form of Softgoal Interdependencies Goals (SIG) catalogues [21], for example, those proposed by Zinovatna for Privacy and Transparency together [22]. There is empirical evidence that the use of SIG catalogues helps to obtain an improved set of NFRs [20, 23, 24].

[4] Recent work [31] brings an operationalization perspective on how to use goal models to define systems considering privacy, security and trust.

All the NFRs mentioned above alone will not be enough to promote trust in software. Trust will only be achieved if the software is transparent enough to demonstrate its qualities to consumers and illustrate that the software is acting accordingly to what consumers expect from it. In fact, Bachmann et al. [25] point out that transparency is one of the critical issues for companies to recover from a situation that led to a lack of trust by consumers. Furthermore, no matter how well-designed software is, there will always be scenarios where failures, accidents, and damages will be inevitable. In these cases, transparent software can help to determine responsibilities and deal with liabilities and civil and criminal implications. Software Transparency implies that software will supply transparency not only of the data it has used and manipulated but also on the *process of development* behind this software. Leite and Cappelli provides a catalogue of transparency with a set of possible solutions to build transparent software [28]. Cysneiros et al. point out that transparency may play an essential role in the acceptance and willingness to buy and use self-driving cars. It also illustrates how legal disputes are an important requirement in the self-driving car domain, suggesting that transparency may play an important role in *legal disputes* [27].

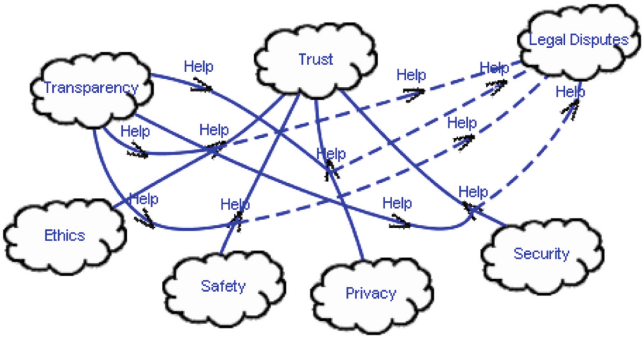

Fig. 1. A SIG with the Key NFRs for responsible software

Hence, we believe that *legal disputes* will certainly also be a requirement that will often be necessary to be tackled in a pro-active manner delivering systems that make use of AI, IofT, or cloud computing. Systems may have to embed ethical choices to guide the reasoning behind decisions made by the software. Injuries, damages, or even deaths resulting from these choices may result in both civil and criminal procedures [15]. On the other hand, in many cases manufacturers will be able to allow users to set their own scenario for ethical and safety decisions which could shift the liability away from software companies but may result in lower sales sparking fear in consumers to accept the risk without totally understanding, or more importantly, trusting how this feature would work. Systems that connect to several devices may be exposed to mal-function due to failures that can go from problems in communication to the lack of appropriate procedures to deal with mal-functions. This can also lead to *legal disputes*, and once again, systems prepared to be transparent can mitigate the consequences of legal actions.

Summarizing, it is our understanding that to support a socially responsible software development that can stimulate citizens/consumers to trust software embedded artifacts that they will be using or buying, we must develop the software with a close look at Ethics, Safety, Security and Privacy. These four NFRs should promote trust when the software is developed, and they must be transparent enough to demonstrate to consumers that the software will meet their expectations for these four NFRs. Each of these NFRs, when managed with Transparency, may have a positive impact on *legal disputes* that may

arise due to suspect software behavior. In Fig. 1, a SIG maps the interactions among Softgoals (NFRs) [21]; so Ethics, Safety, Privacy, and Security contribute positively (*Help*) for Trust, but it needs Transparency to allow the contribution to be effective in mitigating *legal disputes*. We certainly acknowledge that other NFRs will also play a relevant role in distinct types of applications, like Reliability, for instance. Nevertheless, we believe the NFRs illustrated in Fig. 1 is the anchor we need to carefully elicit and model operationalizations for developing software, that people can trust.

4 Conclusion

Society has been changing and evolving at a fast pace. Ubiquitous computing, massive social connection, and growing use of AI/ML (Machine Learning) quite often linked to IofT concepts have been pushing software development to a new paradigm. In a recent paper [8] Agrawal et al. stated: "Machine Learning models are software artifacts derived from data". More then ever, we can not afford to build software targeting one single scenario of use. New software may have immense social impact with legal implications. We need to move our practice to embrace this new scenario where we must build software that is trustworthy and can be accountable for behaving in a socially responsible way.

Our contribution relies on eliciting, from social sciences, basic qualities for socially responsible software to be represented as SIG catalogues, anchored on the NFR Framework [21]. We will be developing catalogues to capture as many as possible solutions (operationalizations) to each NFR illustrated in Fig. 1. We aim to research systematic ways to search and find satisficing solutions to each of the above NFRs and integrate these solutions into a software reuse processes, taking in consideration how each possible solution will impact other NFRs. We will revisit and extend existing catalogues such as Leite's transparency [28] and Zinovatna's privacy and transparency [22], as well as exploring existing operationalizations, such as [31]. We will also focus on better understanding the implications of ethical concepts in the development of software and how it would impact trust as well as its legal ramifications. That will lead to investigate personal and group values that are closely related to ethics aspects [29].

At the core of our research, trust is the primary goal to be achieved. If consumers can trust your company and, by extension, your products (software), they tend to become loyal to your brand and refer your products to acquaintances, which in a social network era can translate into benefits, avoiding *legal disputes*.

References

1. Kolm, J.: How comfortable are Canadians with AI? strategy. http://strategyonline.ca/2017/12/14/how-comfortable-are-canadians-with-ai/. Accessed 13 Nov 2018
2. Bussone, A., Stumpf, S., O'Sullivan, D.: The role of explanations on trust and reliance in clinical decision support systems. In: Proceedings - 2015 IEEE International Conference on Healthcare Informatics, ICHI 2015, pp. 160–169. Institute of Electrical and Electronics Engineers Inc. (2015). https://doi.org/10.1109/ICHI.2015.26
3. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should i trust you?" Explaining the predictions of any classifier. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144. Association for Computing Machinery, New York (2016). https://doi.org/10.1145/2939672.2939778

4. Chaudhuri, A., Holbrook, M.B.: The chain of effects from brand trust and brand affect to brand performance: the role of brand loyalty. J. Mark. **65**, 81–93 (2001). https://doi.org/10.1509/jmkg.65.2.81.18255
5. Park, E., Kim, K.J., Kwon, S.J.: Corporate social responsibility as a determinant of consumer loyalty: an examination of ethical standard, satisfaction, and trust. J. Bus. Res. **76**, 8–13 (2017). https://doi.org/10.1016/J.JBUSRES.2017.02.017
6. Kang, J., Hustvedt, G.: Building Trust Between Consumers and Corporations: The Role of Consumer Perceptions of Transparency and Social Responsibility. https://doi.org/10.1007/s10551-013-1916-7
7. Cui, J., Jo, H., Na, H.: Does corporate social responsibility affect information asymmetry? J. Bus. Ethics **148**, 549–572 (2018). https://doi.org/10.1007/s10551-015-3003-8
8. Agrawal, A., et al.: Cloudy with high chance of DBMS: a 10-year prediction for Enterprise-Grade ML (2019)
9. Delgado-Ballester, E., Munuera-Aleman, J.L., Yague-Guillen, M.J.: Development and validation of a brand trust scale. Int. J. Mark. Res. **45**, 35–56 (2003)
10. Vlachos, P.A., Tsamakos, A., Vrechopoulos, A.P., Avramidis, P.K.: Corporate social responsibility: attributions, loyalty, and the mediating role of trust. J. Acad. Mark. Sci. **37**, 170–180 (2009). https://doi.org/10.1007/s11747-008-0117-x
11. Bowen, J.: The ethics of safety-critical systems. Commun. ACM **43**, 91–97 (2000). https://doi.org/10.1145/332051.332078
12. Pivato, S., Misani, N., Tencati, A.: The impact of corporate social responsibility on consumer trust: the case of organic food. Bus. Ethics A Eur. Rev. **17**, 3–12 (2007). https://doi.org/10.1111/j.1467-8608.2008.00515.x
13. ISO - ISO 26000 Social responsibility. https://www.iso.org/iso-26000-social-responsibility.html. Accessed 22 Oct 2019
14. Bews, N.F., Rossouw, G.J.: A role for business ethics in facilitating trustworthiness. J. Bus. Ethics **39**, 377–390 (2002). https://doi.org/10.1023/A:1019700704414
15. Lin, P.: Why ethics matters for autonomous cars. In: Maurer, M., Gerdes, J., Lenz, B., Winner, H. (eds.) Autonomes Fahren, pp. 69–85. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-45854-9_4
16. Buck, C., Stadler, F., Suckau, K., Eymann, T.: Privacy as a part of the preference structure of users app buying decision. In: Proceedings of the Wirtschaftsinformatik 2017 (2017)
17. Thierer, A.D.: The internet of things and wearable technology: addressing privacy and security concerns without derailing innovation. SSRN **21** (2014). https://doi.org/10.2139/ssrn.2494382
18. Huang, F., Wang, Y., Wang, Y., Zong, P.: What software quality characteristics most concern safety-critical domains? In: 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), pp. 635–636. IEEE (2018). https://doi.org/10.1109/QRS-C.2018.00111
19. Wilson, C., Hargreaves, T., Hauxwell-Baldwin, R.: Benefits and risks of smart home technologies. Energy Policy **103**, 72–83 (2017). https://doi.org/10.1016/J.ENPOL.2016.12.047
20. Veleda, R., Cysneiros, L.M.: Towards an ontology-based approach for eliciting possible solutions to non-functional requirements. In: Giorgini, P., Weber, B. (eds.) CAiSE 2019. LNCS, vol. 11483, pp. 145–161. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21290-2_10
21. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering. Springer, Boston (1999). https://doi.org/10.1007/978-1-4615-5269-7
22. Zinovatna, O., Cysneiros, L.M.: Reusing knowledge on delivering privacy and transparency together. In: 2015 IEEE Fifth International Workshop on Requirements Patterns (RePa), pp. 17–24 (2015). https://doi.org/10.1109/RePa.2015.7407733

23. de Gramatica, M., Labunets, K., Massacci, F., Paci, F., Tedeschi, A.: The role of catalogues of threats and security controls in security risk assessment: an empirical study with ATM professionals. In: Fricker, S., Schneider, K. (eds.) REFSQ 2015. LNCS, vol. 9013, pp. 98–114. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16101-3_7
24. Cardoso, E., Almeida, J.P., Guizzardi, R.S., Guizzardi, G.: A method for eliciting goals for business process models based on non-functional requirements catalogues. In: Frameworks for Developing Efficient Information Systems: Models, Theory, and Practice: Models, Theory, and Practice, pp. 226–242 (2013)
25. Bachmann, R., Gillespie, N., Priem, R.: Repairing trust in organizations and institutions: toward a conceptual framework. Organ. Stud. **36**, 1123–1142 (2015). https://doi.org/10.1177/0170840615599334
26. Cysneiros, L.M., Yu, E.: Non-functional requirements elicitation. In: do Prado Leite, J.C.S., Doorn, J.H. (eds.) Perspectives on Software Requirements. SECS, vol. 753, pp. 115–138. Springer, Boston (2004). https://doi.org/10.1007/978-1-4615-0465-8_6
27. Cysneiros, L.M., Raffi, M., Sampaio do Prado Leite, J.C.: Software transparency as a key requirement for self-driving cars. In: 2018 IEEE 26th International Requirements Engineering Conference (RE), pp. 382–387. IEEE (2018). https://doi.org/10.1109/RE.2018.00-21
28. do Prado Leite, J.C.S., Cappelli, C.: Software transparency. Bus. Inf. Syst. Eng. **2**, 127–139 (2010). https://doi.org/10.1007/s12599-010-0102-z
29. Angela Ferrario, M., Simm, W., Forshaw, S., Gradinar, A., Tavares Smith, M., Smith, I.: Values-first SE: research principles in practice. https://doi.org/10.1145/2889160.2889219
30. Shim, K., Yang, S.: The effect of bad reputation: the occurrence of crisis, corporate social responsibility, and perceptions of hypocrisy and attitudes toward a company. Public Relat. Rev. **42**(1), 68–78 (2016)
31. Salnitri, M., Angelopoulos, K., Pavlidis, M., et al.: Modelling the interplay of security, privacy and trust in sociotechnical systems: a computer-aided design approach. Softw. Syst. Model. **19**, 467–491 (2020). https://doi.org/10.1007/s10270-019-00744-x