International Journal of Computer (IJC)
ISSN 2307-4523 (Print & Online)
https://ijcjournal.org/index.php/InternationalJournalOfComputer/index

Algorithmic Decision-Making Concerns for Software: Non-Functional Requirement Elicitation as a Solution

Nzechukwu Otunneme[a]*, Monday Eze[b], Kuyoro Shade[c], Ayankoya Folasade[d]

[a,b,c,d]*Department of Computer Science, Babcock University Ilishan Remo Ogun State*
[a]*Email: Nzegod4u@gmail.com ; otunemech@babcock.edu.ng*
[b]*Email: ezem@babcock.edu.ng*
[c]*Email: kuyoros@babcock.edu.ng*
[d]*Email: ayankoyaf@babcock.edu.ng*

Abstract

Reference [1,2] Millions of software are lunched yearly and this software depend on data to produce required output. Personal data privacy and security has been a source of public concern for some time, and is usually interpreted in terms of data obtained from interaction with software. It is difficult to know whether a software system's decisions are fair and what considerations were put in place in the system's internal decision-making process if the system's decisions are opaque. This has the potential to cause injustice and bias. In addition, a lack of openness may lead to a decrease in user acceptance and happiness. Algorithmic data-driven decision-making systems are becoming more automated, and they've had a lot of success in a lot of different applications. The General Data Protection Regulation of the European Union and other regulations limits algorithmic use of personal data and has fueled the dispute over the right to disclosure. This research adapted a crowd requirements elicitation model to develop a framework for the proper elicitation non-functional requirement. The developed model uses natural language processing integrated into a chatbot and a document extraction strategy since non-functional requirement exist also as government regulations and industrial standards. Proper and comprehensive elicitation of non-functional requirements will give accurate information on how the system performs its required task and such documents are best in terms of openness to the use of data by algorithms to avoid algorithm decision making concerns.

Key words: Non-functional requirement; Privacy; Software; Openness; Algorithm.

Received: 4/1/2023
Accepted: 4/30/2023
Published: 5/22/2023

---

* Corresponding author.

1. Background

Our modern society revolves around software. It drives our industries, feeds innovation, facilitates access to all digital information, is a cornerstone of modern scientific study, and has paved the way for the birth of new social and political structures-"code is law"[3]. Humans currently live software centered life as it determines how things are run in different sectors of life: health, security, entertainment, fashion, housing, rail management etc. [1,2]. Millions of software are lunched yearly and this software depend on data to produce required output. Personal data privacy and security has been a source of public concern for some time, and is usually interpreted in terms of data obtained from interaction with software [4].

However, in the age of machine learning, understanding how outputs and decisions are computed in these systems can be difficult, as the underlying algorithms can be sophisticated and opaque[5]. It is difficult to know whether a software system's decisions are fair and what considerations were considered in the system's internal decision-making process if the system's decisions are opaque. This has the potential to cause injustice and bias. In addition, a lack of openness may lead to a decrease in user acceptance and contravene data privacy lays . As a result, transparency is becoming more important as a non-functional requirements (NFR).

Institute of Electrical and Electronics Engineers (IEEE) 2430-2019 Trial-Use Standard for Software Non-Functional Sizing Measurements describes Non-functional Requirements as "Any requirement for a software-intensive system or for a software product, including how it should be developed and maintained, and how it should perform in operation, except any functional user requirement for the software. Non-functional requirements (NFRs) concern: the software system or software product quality, the environment in which the software system or software product must be implemented and which it must serve, and the processes and technology to be used to develop and maintain the software system or software product and the technology to be used for their execution" [6,7]. Requirements elicitation can be of various benefits including: to arrange your thoughts and ideas in a logical way, to arrange someone else's thoughts and ideas in a logical way, to understand what a software package must do before making a selection, to decide whether to buy or build a solution, as a point of reference throughout the project to provide a basis for testing, to reduce project cost, to avoid project failure etc. apart from the mentioned advantages of reequipment elicitation this research tries to establish that it can be a solution to privacy concerns relating to the use of software.

Algorithmic data-driven decision-making systems are becoming more automated, and they've had a lot of success in a lot of different applications. These technologies are increasingly being employed to make socially sensitive decisions in recent years [8].

General Data Protection Regulation of the European Union and other regulations limits algorithmic use of personal data and has fueled the dispute over the right to disclosure [5,9]. Because there will be a growing demand for openness in algorithmic decision making, computer scientists will have to take the lead in building algorithms and frameworks that permit explanations. The comprehensibility of a system can be improved by explanations . As a result, explainability—the ability to present justifications—can be seen of as a means of achieving transparency. Current machine learning techniques, in particular those based on deep learning, are

unable to make clear causal links between input data and final decisions [10].

2. Existing Models

Chazette & Schneider, (2020) used LimeSurvey to build and implement an online questionnaire based on the questions and metrics. It had 16 questions (11 multiple choice, 5 open-ended): three on demographics, one self-assessment question on software skills, four on software use, three on software difficulties, three on explanation needs, one on frequency, and one on presentation of explanations.

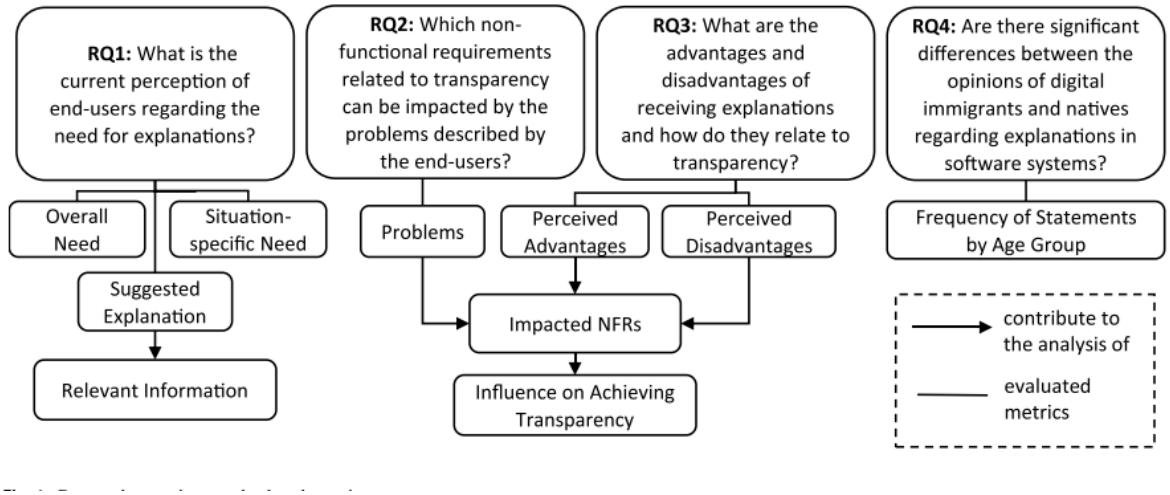

Figure 1: Research questions and related metrics [5]

Reference [11] built a model (see Figure 2) to elicit ERP requirements from non-German citizens for a German MyERP company in order to outperform their German competitors. This approach includes stages such as recognizing the crowd, as well as any cheaters who may be present. The crowd was kept active in the project by a policy that attempts to reward high-performing workers while penalizing low-performing workers. The crowd was asked to categorize the major activities in order to support easy and high-quality requirement elicitation. To resolve conflicts, the identified requirements are then prioritized. Duplicate requirements are also removed. This proposed model, on the other hand, drew crowd members from a single source (LinkedIn) due to the product's specialty, making it inflexible.

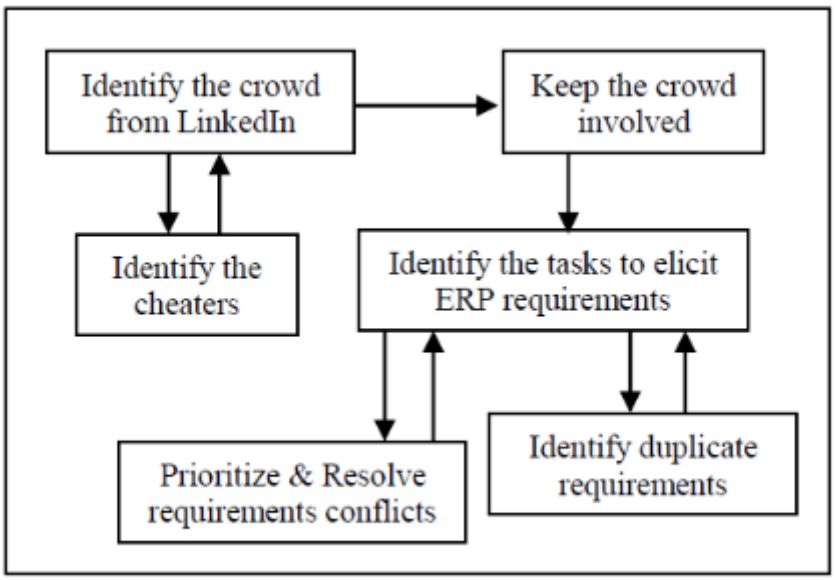

Figure 2: ERP Requirement Elicitation Model [11]

Crowd Requirement Engineering (CrowdRE) is a new method for gathering software needs that taps into the collective intelligence of the crowd. The strength of a crowd lies in its range of knowledge and talents; but, controlling the crowd, assessing, and annotating crowd requirements remains a difficulty. CrowdRE model known as the Crowd Requirement Rating Technique (CrowdReRaT) that enable annotation of requirements by different crowd members at various level as seen in figure 3.

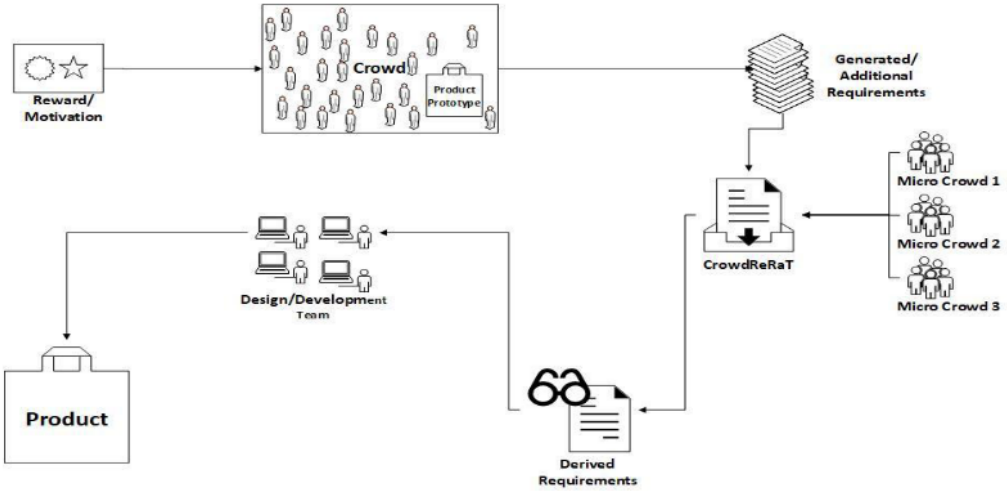

Figure 3: CrowdReRaT Model [12]

4. Proposed Model

Adopt a chat bot as a crowd requirement technique for gathering of non-functional requirement, stake holder using pre-elicited functional requirements. The crowd requirement gathering technique will make use of an

intelligent chat bot with dialogue management using Natural Language processing through a user interface to interact with the stake holder. The system should enable the administrator categorize the stakeholder according to the level of their impact or their influence on the system.

Explore archival material which include Industrial Standards, Government Policies and User Application Review for non-functional requirement of similar or previous versions of the system being specified. The archival documents will be read using text extractor and semantically analysed using Natural language processing as seen in figure 4.

The elicited non-functional requirements are integrated into software policy manual for all stakeholders.

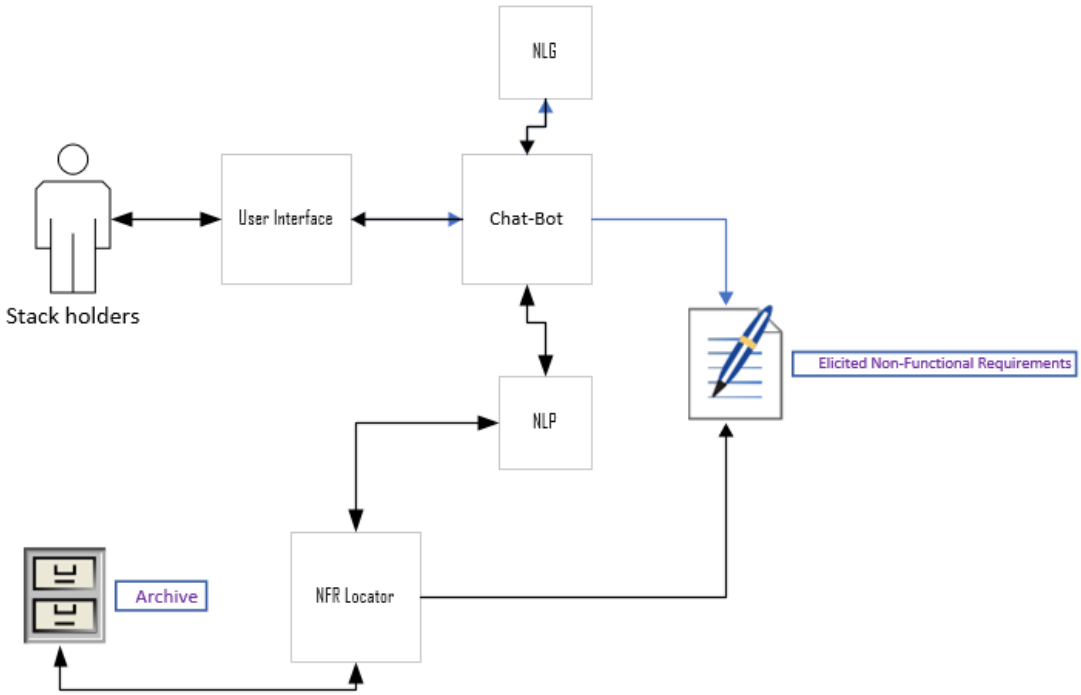

Figure 4: Non-functional Elicitation Framework (Researchers' Model)

4. Conclusion

Because of the verge and ambiguous nature of non-functional requirements, the elicitation is difficult. Proper elicitation of non-functional requirements involves large stakeholder participation which the proposed framework will effectively handle. This research focused on the openness of software development teams to overcome data privacy concerns as data is being manipulated by algorithm. Non-functional requirement efficiently describe how the software performs the takes specified by the non-functional requirements. Proper and comprehensive elicitation of non-functional requirements will give accurate information on how the system performs its required task and such documents are best in terms of openness to the use of data by algorithms to avoid algorithm decision making concerns. Requirements engineers need to explore the costs and benefits of presenting explanations to the user and what they mean in terms of functional and non-functional requirements.

Further work should be caried out develop a classification model for the elicited non-functional requirements and also check the effect of transparency of software development teams on software users trust.

References

[1] N. C. Otuneme, M. O. Eze, O. D. Adekola, A. O. Adebayo, and S. O. Ogunlere, "Railway Travel-Time Optimization System," vol. 7, no. 11, pp. 108–116, 2018.

[2] O. Akintunde, N. Otuneme, O. Adetunji, and A. Akinsanya, "Health Information Exchange Model for Nigerian Health Information Systems," *Article in International Journal of Computer Science and Information Security*, vol. 17, no. 2, p. 181, 2019.

[3] R. Di Cosmo, "Software Heritage: Why and How We Collect, Preserve and Share All the Software Source Code," in *40th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, IEEE/ACM, 2018, pp. 2–2.

[4] L. E. Buck and B. Bodenheimer, "Privacy and personal space: Addressing interactions and interaction data as a privacy concern," *Proceedings - 2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops, VRW 2021*, pp. 399–400, 2021, doi: 10.1109/VRW52623.2021.00086.

[5] L. Chazette and K. Schneider, "Explainability as a non-functional requirement: challenges and recommendations," *Requirements Engineering*, vol. 25, pp. 493–514, 2020, doi: 10.1007/s00766-020-00333-1.

[6] IEEE Computer Society, *IEEE Trial-Use Standard for Software*. The Institute of Electrical and Electronics Engineers Standards Assosiation, 2019. doi: 10.1109/IEEESTD.2019.8870263.

[7] IEEE, "IEEE 610.12 - Standard Glossary of Software Engineering Terminology | EnIEEE. 1990. IEEE 610.12 - Standard Glossary of Software Engineering Terminology | Engineering360.gineering360," 1990. Accessed: May 27, 2021. [Online]. Available: https://standards.globalspec.com/std/398645/IEEE 610.12

[8] S. Alzu'Bi, B. Hawashin, M. Eibes, and M. Al-Ayyoub, "A Novel Recommender System Based on Apriori Algorithm for Requirements Engineering," *2018 5th International Conference on Social Networks Analysis, Management and Security, SNAMS 2018*, pp. 323–327, 2018, doi: 10.1109/SNAMS.2018.8554909.

[9] Regulation(eu), "EUR-Lex - 02016R0679-20160504 - EN - EUR-Lex," *EU Open Data Portal*, Apr. 27, 2016. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A02016R0679-20160504 (accessed Apr. 08, 2022).

[10] R. Hamon, H. Junklewitz, S. Ignacio, G. Malgieri, and P. Hert De, "Bridging the Gap Between AI and Explainability in the GDPR: Towards Trustworthiness-by-Design in Automated Decision-Making," *IEEE Computational Intelligence Magazine*, vol. 17, no. 1, pp. 72–85, 2022, doi: 10.1109/MCI.2021.3129960.

[11] P. K. Srivastava and R. Sharma, "Crowdsourcing to elicit requirements for MyERP application," in *1st International Workshop on Crowd-Based Requirements Engineering, CrowdRE 2015 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Dec. 2015, pp. 31–35. doi: 10.1109/CrowdRE.2015.7367586.

[12] O. Adetunji, E. Oyenuga, and N. Otuneme, "Crowd Requirement Rating Technique (CrowdReRaT) Model for Crowd Sourcing," *International Journal of Computer Applications*, vol. 176, no. 22, pp. 9–14, 2020, doi: 10.5120/ijca2020920178.