Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 913–917
DOI: 10.15439/2023F9537
ISSN 2300-5963 ACSIS, Vol. 35

The Effects of Native Language on Requirements Quality

Fayona Cowperthwaite
0000-0002-7501-7048
University of Gothenburg
Sweden
facowperthwaite@gmail.com

Jennifer Horkoff
0000-0002-2019-5277
University of Gothenburg
Chalmers University of Technology
Sweden
jennifer.horkoff@gu.se

Sylwia Kopczyńska
0000-0002-9550-3334
Poznan University of Technology
Poland
sylwia.kopczynska@cs.put.poznan.pl

***Abstract*—[Context and motivation] More and more often software development projects involve participants of diverse nationalities and languages. Thus, software companies tend to use English as their business language. Moreover, to better prepare for future jobs, students consciously choose university courses in English. [Question/problem] As a result there is an increasing number of software engineers who are working or studying in a language which is not their native language. The question arises whether native language has an effect on the quality of natural language requirements. [Principal ideas/results] From the analysis of the requirements formulated by 44 participants of our empirical study, it follows that native language may have a negative effect on requirements quality, e.g., ambiguity, variability, and grammar issues. Furthermore, different native languages might drive to different quality issues. [Contribution] In order to prevent quality issues, our findings might be used by educators to adjust their materials to cater to different language groups, while practitioners might use them to improve their requirements review process.**

I. Introduction

SOFTWARE engineering is a diverse field, both in terms of research areas and worker backgrounds. This diversity is present in the industry, and companies are increasingly using English as their business language, no matter what country they are based in. University students are also globally mobile, with many who have the means often choosing to study all or part of their higher education abroad in English. This means that there is an increasing number of software engineers who are working or studying in a language that is not their native language.

Software engineers often use requirements specifications, either writing or developing systems from them, where the quality of the specification could determine the quality of the end product. The success of a software development project is said to depend on the quality of its requirements specification [1], [2]. Requirements are often written in natural language and, thus, the language used in that requirement could also have an effect on the quality of the specification.

The purpose of this study is to analyze natural language requirements written in English to determine (1) whether a author's native language has an effect on the quality of these requirements, and (2) which qualities are affected. In this paper, the term "native language" is defined as being the language of the country in which a person is born, raised, and receives early years of education. In an agile context, natural language requirements can either be written in the Software Requirements Specification (SRS) style or as user stories.

The findings from this study could support industry practitioners, research, and requirements engineering education. Targeted teaching and training could be developed to improve not only the overall quality of requirements but also to focus on the qualities that native speakers frequently have problems with. The study outcome could also help companies with requirements review processes, and quality checklists definition to identify or avoid requirements issues early on in development.

II. Background and Related Work

The IEEE Recommended Practice for Software Requirements Specifications [3] presents guidelines on how to produce "good" natural language SRS-style requirements. The guidelines detail eight characteristics that individual requirements should possess and five characteristics that a set of requirements should have. The recommended practice states that individual requirements should be: necessary; appropriate; unambiguous; complete; singular; feasible; verifiable; correct; and conforming (when applicable). A set of requirements should be: complete; consistent; feasible; comprehensible; and able to be validated. If an individual requirement or set of requirements violates one or more of these qualities, then it is not considered to be "good".

The INVEST criteria, originally discussed by Wake in 2003 [4], are specifically for evaluating the quality of user stories, rather than SRS-style requirements. According to the criteria, a user story should be: independent, negotiable, valuable, estimable, small, and testable [5]. If the story does not meet one or more of these criteria, then it is not of good quality.

There is a large body of work on requirements quality, with some focusing on specific qualities of a requirements specification and others giving a broader overview of what quality might be. Kiyavitskaya et al. [1] and Fabbrini et al. [6] take a detailed linguistic approach to identify ambiguity in requirements specifications. Antinyan et al. [7] focus on different requirements quality and developed a metric to measure the

complexity of a requirement. With a broader look at all of the potential qualities of a requirements specification, Knauss et al. [8] developed a GQM approach to improving requirements quality. Genova et al. [2] also had a wider view of which requirements qualities to consider when creating the framework and tool for improving the quality of a requirements specification. However, while these studies were conducted in English, none of them looked at the linguistic background of the participants.

III. Research Methodology

Research Questions. Our study aims to answer the following research questions:
RQ1: Does the native language have an effect on the quality of natural language requirements?

- RQ1.1: Which requirements qualities are affected?
- RQ1.2: Do any particular languages have greater effects on requirements quality?

Participants and Data Collection. We aimed to find participants who had a software engineering background, and who could potentially be asked to write requirements. The participants were selected on the basis of convenience sampling. Survey participants were reached via the REFSQ 2022 conference, LinkedIn, Facebook, Twitter, Discord, and email, and via sharing the survey link with the students studying software engineering at the Universities the authors work for. Thus, the participants were a mix of students, researchers, and industry practitioners within software engineering. We created an online survey hosted on sosci.de. The survey was piloted by two representatives of the study participants. We decided to ask two students (those who might have the lowest experience with requirements) who gave feedback which was used to refine the survey questions. The first five questions in the survey were demographic questions. The sixth question was a simple domain description after which the participant was asked to write five natural language requirements (either SRS style or user stories) for the example domain. The survey questions and study material are available online [9].

Data Analysis. The qualitative data was analyzed using thematic coding as per Saldana [10] with two coding iterations. The thematic coding process used a coding dictionary that we created, which covered violations of any of a selected subset of the IEEE characteristics of individual requirements [3] or four of the INVEST criteria for user stories [4], [5]. When analyzing SRS-style requirements, we used the 2018 IEEE guidelines [3] that detail what good individual requirements should possess: correct; ambiguous; verifiable; necessary; appropriate; complete; singular; and feasible. The characteristic of "conforming" was not included in the analysis as the participants in the case study were not given a set template or writing style to follow. We chose to exclude the five characteristics for a set of requirements as we only asked participants to provide a sample of requirements rather than a complete requirements specification, and we did the analysis on each individual requirement. We also looked at whether a requirement is vague because we felt that being imprecise might not necessarily mean the requirement is ambiguous or unverifiable – it may just need more details or explanation.

For user story analysis, we used the INVEST criteria [4], [5]. "Independent" was excluded as it would require evaluation of the user stories as a set, while analysis was conducted on individual user stories. We also made note of whether the user story was correctly formed according to the Agile Alliance user story template [5]. As SRS-style requirements and user stories have different purposes and quality criteria, we did not use the SRS-style characteristics to analyze user stories, and the INVEST criteria were not applied to SRS-style requirements. The requirements in this study are in written form, and so we also considered language quality as a contributor to the overall requirements quality. Therefore, we applied codes for typos and grammar issues.

After the first author completed the first analysis pass, a sample of 10 randomly-chosen responses (a total of 50 requirements) was analyzed by the second author. Then, we came together to discuss any differences and how to improve the coding book. Coding was redone by the first author based on these discussions. Tab. I shows three examples of requirements received in the survey and the final codes that were applied. The final coding book with examples is available online [9]. Fig. 1 gives an overview of the thematic codes.

IV. Results

47 people answered the survey. However, three respondents did not complete the requirements writing task sufficiently; therefore, 44 survey responses were considered for the analysis with 220 requirements in total. For simplicity, and to aid comparison, we report percentages over all collected requirements (user stories and SRS-style ones), even though not all errors are applicable to all requirements.

Respondent Demographics. Fig 2 shows the native languages of our respondents. The majority of respondents had Polish as a native language, due to the third author sharing the survey link with the Master students of software engineering specialty. Swedish, Chinese, and English were the next most common native languages of respondents. Although there are many dialects and languages, the participants are known as students of Beijing University of Technology where the language of instruction is Beijing Mandarin.

In terms of roles within software engineering, 22/44 respondents were students of master-level studies who might be treated as novice requirements engineers. Industry practitioners were the next largest group with 8 participants, and there were also 5 Researchers. 9/44 respondents had multiple roles within software engineering: 6 were both a student and an industry practitioner; 2 were both a student and researcher; one person was an industry practitioner and a researcher.

Among the 14 respondents who selected the industry practitioner role as either their only role or as one of their multiple job roles, 4 stated their roles as "Developer" and 3 "Software Developer". There was one answer each for the following roles: "Senior Software Engineer"; "software engineer"; "System Architect"; "Technical project manager";

the 44 participants) given in the online survey, 233 codes were applied. This means that multiple codes were applied to some requirements. The four codes that were applied the most were: unverifiable (25.91% of all codes); ambiguous (21.82%); grammar issue (18.64%); and incorrect format (11.82%). The codes with the fewest applications (but more than 0) were: incorrect (0.45% of codes); unfeasible (also 0.45%); and inappropriate (0.91%).

Looking at Table II, the native Chinese speakers had by far the highest percentage of occurrence of unverifiable codes (46.67%). The native Arabic speakers had the second highest percentage (30%), and the native Polish speakers had the third highest percentage of unverifiable requirements with 28.24%. Native Arabic speakers had the highest percentage of ambiguity occurrences with 50% of the requirements given being coded as ambiguous. The Polish native speakers had the second highest percentage of ambiguous code occurrences with 28.24%.

Observation 2: There are four requirements qualities that were affected the most that are: verifiability, unambiguity, grammar correctness, and correct format.

Observation 3: Native speakers of Polish, Arabic, and Chinese introduced the highest number of errors.

Other Factors. In our survey, we collected data on other factors such as level of education, number of languages spoken, and mother tongue. We found that holding a Bachelor's degree as the highest level of education and speaking four or more languages had a negative effect on requirements quality. This data is omitted for space reasons, but results are available online [9].

V. Discussion

All participants in the study did make requirements quality errors, regardless of their native language. However, being a native speaker of Chinese, Arabic or Polish may have a negative influence on the quality of requirements that are written by those speakers. Two of these three languages have a writing system that is entirely different from English, which uses the Roman alphabet.

Unverifiability was the most common error made by the study participants and is a quality that often concerns Non-Functional Requirements (NFRs). The second most common error was Ambiguity. Althouth, as mentioned in Section II, ambiguity is a widely-researched topic within software engineering [11], [12], [1], [13], [14], the results from the study in the present paper suggest that continuing research and education in this area seems still needed.

The third most common error—grammar issues—could also be considered to be connected to ambiguity in some cases. Introducing grammar-checking tools and proofreading into the requirements writing process might help in preventing these errors. Then, there was the incorrect format error type as the survey participants did not use what is considered to be the standard user story format [5], [4]. Thus, using such frameworks and tools for improving user story quality [15], [16] might be valuable.

Chinese, Arabic and Polish appeared to have a greater negative effect on requirements quality than the rest of the languages in our studies. However, we cannot claim what is the root cause of this observation. It is necessary to investigate whether requirements quality is affected by the native language itself (linguistic differences), the level of English education, education within software engineering, or other factors. Future studies that discover the root causes might deliver guidelines for requirements for engineers and educators.

VI. Threats to Validity

Internal: Thematic coding brings threats to validity due to being subjective in its nature and subject to the bias and experience of the person doing the analysis. In order to mitigate this and minimize the threat, the second author received the coding dictionary that we created and independently coded a sample of 20% of the requirements obtained in the study. The English level of participants was not taken as the variable in the study, but we had an inclusion criterion– the participants need to have enough knowledge and skills so that they are able to either study or work in English.

External: The study may not have a large scope of generalisability as even though the survey was shared with non-students, a large portion of the data collection was reliant on students. However, it could be argued that the results from student data could be indicative of the software engineering industry as they frequently work and might be treated as novice employees.

VII. Conclusion

This study investigates whether native language has an effect on the quality of requirements. The results from the analysis of the online survey data suggest that native language may indeed have an effect on requirements quality as well as on the type of error introduced by the requirements writer. It follows from our study that more work and education need to be carried out on improving verifiability and ambiguity within requirements. Moreover, more training is needed also on how to write user stories so that they are well-formed. Grammar issues were also quite prevalent across all requirements. Our results might be used by practitioners to include quality checks of the errors in their review process and by educators to draw the attention of students to errors they might introduce and teach them how to prevent making those errors. Moreover, researchers might use our results to investigate the root causes of why native speakers of some languages make more errors than native speakers of other languages.

References

[1] N. Kiyavitskaya, N. Zeni, L. Mich, and D. M. Berry, "Requirements for tools for ambiguity identification and measurement in natural language requirements specifications," *Requir. Eng.*, vol. 13, no. 3, pp. 207–239, 2008.

[2] G. Génova, J. M. Fuentes, J. Llorens, O. Hurtado, and V. Moreno, "A framework to measure and improve the quality of textual requirements," *Requir. Eng.*, vol. 18, no. 1, pp. 25–41, 2013.

[3] "Iso/iec/ieee international standard - systems and software engineering – life cycle processes – requirements engineering," *ISO/IEC/IEEE 29148:2018(E)*, pp. 1–104, 2018.

TABLE II
RAW DATA OF THE NUMBER OF ERRORS FOUND FOR EACH REQUIREMENTS QUALITY PER NATIVE LANGUAGE OF THE PARTICIPANTS

Native language [no. of participants]	Requirements quality code counts (% of requirements with quality code per native language)											
	Incorrect	Ambiguous	Unverifiable	Unnecessary	Inappropriate	Not singular	Unfeasible	Unnegotiable	Untestable	Incorrect format	Typo	Grammar issue
Polish [17]		24 (28.24%)	24 (28.24%)	2 (2.35%)	1 (1.18%)	7 (8.24%)			1 (1.18%)	5 (5.88%)	13 (15.29%)	23 (27.06%)
Swedish [7]		5 (14.29%)	5 (14.29%)					3 (8.67%)	5 (14.29%)	4 (11.43%)	1 (2.86%)	3 (8.57%)
Chinese [6]	1 (3.33%)	6 (20%)	14 (46.67%)			5 (16.67%)	1 (3.33%)			5 (16.67%)		4 (13.33%)
English [5]		4 (16%)	6 (24%)			1 (4%)		1 (4%)	4 (16%)	5 (20%)	3 (12%)	
Arabic [2]		5 (50%)	3 (30%)	1 (10%)							1 (10%)	2 (20%)
Cantonese [1]		3 (60%)	2 (40%)								1 (20%)	2 (40%)
German [1]										2 (40%)		
Greek [1]									1 (20%)	3 (60%)	1 (20%)	2 (40%)
Persian [1]				1 (20%)				1 (20%)		2 (40%)		2 (40%)
Brazilian Portuguese [1]			2 (40%)	1 (20%)		1 (20%)						
Amharic [1]		1 (20%)	1 (20%)								1 (20%)	1 (20%)
Korean [1]					1 (20%)	1 (20%)						2 (40%)

TABLE III
REQUIREMENTS QUALITY CODE OCCURRENCES OVER ALL 220 REQUIREMENTS RECEIVED IN THE ONLINE SURVEY

Requirements quality code	**Number (% of occurrence)**
Incorrect	1 (0.45%)
Ambiguous	48 (21.82%)
Unverifiable	57 (25.91%)
Unnecessary	5 (2.27%)
Inappropriate	2 (0.91%)
Not singular	15 (6.82%)
Unfeasible	1 (0.45%)
Unnegotiable	5 (2.27%)
Untestable	11 (5%)
Incorrect format	26 (11.82%)
Typo	21 (9.55%)
Grammar issue	41 (18.64%)
Total number of codes	233

[4] B. Wake, "INVEST in good stories, and SMART tasks - XP123," https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/, Aug. 2003, accessed: 2022-3-4.

[5] "What does INVEST stand for?" https://www.agilealliance.org/glossary/invest/, Dec. 2015, accessed: 2022-3-4.

[6] F. Fabbrini, M. Fusani, S. Gnesi, and G. Lami, "The linguistic approach to the natural language requirements quality: benefit of the use of an automatic tool," in *Proceedings 26th Annual NASA Goddard Software Engineering Workshop*. IEEE Comput. Soc, 2002.

[7] V. Antinyan, M. Staron, A. Sandberg, and J. Hansson, "A complexity measure for textual requirements," in *2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA)*. IEEE, 2016.

[8] E. Knauss and C. E. Boustani, "Assessing the quality of software requirements specifications," in *2008 16th IEEE International Requirements Engineering Conference*. IEEE, 2008.

[9] F. Cowperthwaite, J. Horkoff, and S. Kopczyńska, "The Effects of Native Language on Requirements Quality - Additional Material," Feb. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.7649140

[10] J. M. Saldana, *The coding manual for qualitative researchers*, 2nd ed. London: SAGE Publications, 2013.

[11] M. Bano, "Addressing the challenges of requirements ambiguity: A review of empirical literature," in *2015 IEEE Fifth International Workshop on Empirical Requirements Engineering (EmpiRE)*, 2015, pp. 21–24.

[12] V. Gervasi, A. Ferrari, D. Zowghi, and P. Spoletini, "Ambiguity in requirements engineering: Towards a unifying framework," in *From Software Engineering to Formal Methods and Tools, and Back*. Cham: Springer International Publishing, 2019, pp. 191–210.

[13] B. Gleich, O. Creighton, and L. Kof, "Ambiguity detection: Towards a tool explaining ambiguity sources," in *Requirements Engineering: Foundation for Software Quality*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 218–232.

[14] A. Bajceta, M. L. Ortiz, W. Afzal, P. Lindberg, and M. Bohlin, "Using nlp tools to detect ambiguities in system requirements - a comparison study," in *5th Workshop on Natural Language Processing for Requirements Engineering @ REFSQ*, March 2022. [Online]. Available: http://www.ipr.mdh.se/publications/6390-

[15] G. Lucassen, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper, "Improving agile requirements: the quality user story framework and tool," *Requir. Eng.*, vol. 21, no. 3, pp. 383–403, 2016.

[16] ——, "Forging high-quality user stories: Towards a discipline for agile requirements," in *2015 IEEE 23rd International Requirements Engineering Conference (RE)*. IEEE, 2015.