

The Significance of Cultural Veracity in Software Systems: A software practitioners' perspective

FAITH CULAS, The University of Auckland, New Zealand

JUDITH PERERA, The University of Auckland, New Zealand

MATTHIAS GALSTER, University of Canterbury, New Zealand

KELLY BLINCOE, The University of Auckland, New Zealand

Software requirements is one of the main factors in the development of software systems. Gathering all the requirements correctly and completely is essential to deliver the expected product to the users. Requirements Technical Debt (RTD) is when software practitioners make sub-optimal decisions concerning requirements either deliberately or inadvertently. RTD related to veracity is known as Veracity Debt. This paper reports the findings of one type of veracity which is cultural veracity which is part of a main study which explores all five types of veracity. Interestingly, cultural veracity requirements was interpreted as "user specific requirements". The findings show that there were different opinions of what cultural veracity means and its significance when developing a software system. To the best of our knowledge this is the first ever study on cultural veracity related to software requirements.

Additional Key Words and Phrases: Requirements Engineering, Requirements Technical Debt, Veracity, Cultural Veracity

1 Introduction

"Requirements Technical Debt (RTD) captures the consequences of sub-optimal decisions made concerning requirements, either deliberately (for strategic gains) or inadvertently (due to changes in context), during the identification, formalization, and implementation of requirements" [6]. Some examples of sub-optimal decisions made concerning requirements are missing to capture all the user essential needs, insufficient documentation or no documentation at all, bad prioritization of requirements and inadequate implementations of requirements (could be functional or non-functional). Although RTD has been extensively researched, there is yet a scarcity of research on the quantification of RTD. Quantification of RTD can help better manage the RTD by making informed decisions based on the quantification. A Systematic mapping study was conducted to establish a theoretical foundation to improve the understanding of RTD quantification [6]. This has also resulted in a formal definition of RTD and a conceptual model, the Requirements Technical Debt Quantification Model (RTDQM) [7]. Software systems are becoming more and more concerned with truthfulness and trustworthiness, and software requirements related to veracity are vital to the development of software systems. "Software requirements concerning veracity are related to the trustworthiness, truthfulness, authenticity, provenance, integrity, and demonstrability of the software product" [2]. Perera and her colleagues presented five different types of veracity requirements: Regulatory, Process, Data, Financial and Cultural [8]. Software requirements concerning veracity can be either functional or non-functional or both, depending on the software system.

This study focuses on one type of veracity which is cultural veracity. A systematic literature review done by Ortega et al. [4] provides insight on how cultural veracity is viewed but there is still a void of research on how it is perceived in the industry. The aim of this study is to understand how practitioners perceive cultural veracity, and the impact of

Authors' Contact Information: Faith Culas, The University of Auckland, Auckland, New Zealand, fcul804@aucklanduni.ac.nz; Judith Perera, The University of Auckland, Auckland, New Zealand, jper120@aucklanduni.ac.nz; Matthias Galster, University of Canterbury, Christchurch, New Zealand, matthias.galster@canterbury.ac.nz; Kelly Blincoe, The University of Auckland, Auckland, New Zealand, k.blincoe@auckland.ac.nz.

RTD related to it and how quantifying could support decision-making for better managing it. Our study answers the following research questions:

- RQ1: What are practitioners' perspective on cultural veracity in software products?
- RQ2: What are practitioners' perspective on the impact of cultural veracity requirement technical debt in software products?
- RQ3: How could quantifying Cultural Veracity RTD support informed decision-making?

This paper is divided into five sections: Section II explains the methodology we used to understand how cultural veracity and how RTD related to it is perceived in the industry; Section III outlines our results; and Section IV is a discussion on the findings as well as highlights any threats to validity.

2 Methodology

This section describes the study design and the coding methods that were followed to find themes.

2.1 Study design

The findings of this paper are part of a main study conducted by Perera and her colleagues [5]. They conducted semi-structured interviews to have more in-depth conversations with software practitioners to better understand quantification of RTD and veracity RTD and how it can support the decision-making for managing RTD. To investigate the perception of cultural veracity (RQ1), the impact of RTD related to cultural veracity (RQ2) and the benefit of quantifying it (RQ3) we added a section in the interviews on cultural veracity. One of the most popular methods for gathering case study data for software engineering is through interviews [9]. We first conducted two pilot interviews and then interviewed six experts in the industry.

2.2 Coding methods

The interview transcripts were coded using thematic analysis [1]. Themes were identified across the eight interviews conducted using NVivo14.

Interviewee	Company Size	Service (•) or Product (•) based	Experience in years			Development Method	Country of Work
			2-10	11-20	20+		
P1	large	•		•		Agile	Argentina
P2	large	•	•			Agile	Sri Lanka
E1	large	•	•			Agile, Kanban	Australia
E2	large	•			•	Agile, Waterfall	New Zealand
E3	large	•		•		Agile	New Zealand
E4	small	•			•	Agile	New Zealand
E5	large	•	•			Agile	New Zealand
E6	small	•		•		Agile, Lean	New Zealand

Table 1. Demographics of the interviewees among the categories of company size (i), service or product based (ii), experience in years (iii), development method (iv) and country of work (v)

3 Results

3.1 Demographics

This study had a total of eight interviews: two being pilot and six being expert interviews from the industry. As shown in Table 1, we interviewed practitioners of various experience levels from different domains, across different scales of companies. For company size we categorized companies with less than 50 employees as small [3]. Most of them practised Agile and were located in New Zealand. Tables 2 and 3 are the demographics of the interviewees particular to their role and domain and also also whether they were in a position to make decisions or not. To determine whether they were in a position to make decisions or not, we analyzed the interview responses, considering their roles and involvement in the decision-making process.

Table 2. Demographics of the pilot interviews among the categories of positions (i) and their domains (ii) and whether they are a decision maker (iii)

Interviewee Name	Position or Role	Domain of work	Decision Maker
P1	Project manager	Finance, Arms	NO
P2	Software Engineer	Security	NO

Table 3. Demographics of the expert interviews among the categories of positions (i) and their domains (ii) and whether they are a decision maker (iii)

Interviewee	Position or Role	Domain of work	Decision Maker
E1	Software developer	Identity and Access management	NO
E2	Retired Analyst/Project manager	IT hardware, Automotive	NO
E3	Engineering lead	finance, ERP, Medical	YES
E4	CEO of a startup	Exports, Commercial software	YES
E5	Software developer	IT Network, Shipping	NO
E6	Automation test architect	Finance, Aerospace, SaaS	YES

3.2 Perspective of cultural veracity in the industry

RQ1: What is practitioners' perspective on cultural veracity in software products?

One of the questions in the interviews asked was what cultural veracity means to them in a software product. While there are different opinions among them as shown in Table 4, most of them(62.5%) had one common interpretation which was "user specific requirements". Interviewee E6 even quoted "people have so many interpretations of culture anyway".

3.3 Perspective of the impact of cultural veracity RTD in the industry

RQ2: What is practitioners' perspective on the impact of cultural veracity requirement technical debt in software products?

In terms of impact, three incidents emerged from the interviews (Table 5). The impacts of each of them were different depending on the scenario. The impacted party also can be the user or the company or even both. The first incident was not much of an impact as it was a "one development day change" in terms of "rework cost" associated with it while the third incident led to changing the user interface completely which incurred a significant amount of "rework cost"

Table 4. Interpretation of cultural veracity requirements by the interviewees

Interviewee	Emerged Theme	Interpretation of cultural veracity requirements
P1	User specific requirements	Requirements that need to be addressed that affect the end user
P2	Organizational culture	Team culture
E1	User specific requirements	Accommodating diverse group of people and looking at different needs that the end users would have
E2	–	No experience with cultural veracity requirements
E3	User specific requirements	People are culture specific, so user specific requirements
E4	User specific requirements	More to do with design and User Interface (UI) and human interface
E5	User specific requirements	Accessible application, get feedback from community, cater to different users
E6	Organizational Culture	Align software requirements with the cultural values, team culture

and had to be validated with the users frequently. This is another theme which emerged from the interviews which is "user-feedback loop". This confirms the findings of Perera et al. [7] that unlike code related TD, user-feedback loop is very important for RTD. The second incident was an enhancement which was added later on to cater to more customer groups which indicated a financial impact on the company. The impact level being low or high is not determinable as only three incidents were identified and this also subjective based on the context. Although cultural did not emerge as the veracity requirement with the most impact, it did have significant impact to cause rework.

Table 5. Examples of cultural veracity RTD incidents that interviewees discussed

Cultural Veracity RTD Incident	Requirement problem associated with the incident
1. The UI had to be changed to incorporate users with only one single name as currently the UI required both first name and last name	Not capturing essential gathering requirements user needs while
2. The learning platform developed had to be changed to be made accessible for users of different languages, who read right to left, who had visual disabilities	Changes in context
3. The UI presented to the users was not accepted in the arms industry culture and needed to be changed	Lack of requirements validation with the user, Miscommunication

3.4 Benefit of quantifying cultural veracity RTD and the relevant concepts from the RTD quantification model

RQ3: How could quantifying Cultural Veracity RTD support informed decision-making?

The quantifiable factors considered when rectifying RTD were mostly effort/time and financial. While informally one of the key factors considered while rectifying RTD was end-user satisfaction and the relationship maintained with the users/customers. So the "User Need" from the RTD model [6] was an essential concept considered when making decisions to manage RTD. "Benefit" and "Cost" were the two high level concepts discussed when fixing the RTD incurred.

"Cost of Rectifying RTD", "RTD Interest", "New Code Cost associated with RTD", "Rework Code Cost associated with RTD" are the model concepts related to "Cost" and "Benefit of Rectifying" and "End user satisfaction level" are the model concepts related to "Benefit" that emerged from the model. Table 6 presents a more detailed mapping of the three incidents to the RTD model concept. The RTDQM thus helps to identify what concepts are relevant for cultural veracity related RTD and this lays a foundation to help understand better what concepts of the model are the most relevant to help practitioners make more informed decisions based on the quantification of those concepts. While the relevant concepts that emerged from the interviews confirm the usefulness of the model for quantification of cultural veracity RTD, this is based on three incidents and future work is needed to validate the relevant concepts and even relate to more model concepts that are not captured.

4 Discussion

In this section we interpret the results and also present any limitations and implications for researchers and practitioners, and future work.

4.1 Discussion of the findings

4.1.1 Importance of user feedback loop. As seen in Table 5, two of the incidents are due to lack of input and validation from the users/customers. Also the most emerged theme being "User specific requirements" clearly depicts that even though everyone practices Agile (Table 1), the user feedback is something to work on to make better decisions and avoid losses. This research further confirms the role of users in RTD management as highlighted in [7].

4.1.2 RTD model mapping to Cultural veracity requirements. As seen in Table 6, many concepts of the model were captured while discussing about the cultural veracity incident which shows that the RTD model can be adapted to be applied for Cultural veracity RTD. While only effort/time was the only quantified factor across all interviews, other factors like user satisfaction, team culture and satisfaction, relationships were also highlighted as benefits or could result in loss if not handled well. It can also be seen that some concepts like "(Documented) Requirement", "Quality", "Design", "Priority" were not addressed. This could be because of the short duration the interviews that all concepts were not covered, or could be because these concepts are not part of the software development process, or could be because cultural veracity itself isn't formally recognized in the requirement step.

4.1.3 Impact of cultural veracity. The level of impact is dependant on the context. While the first incident was a small change which required only one developer day to be fixed and did not impact a wide range of users, the third incident needed a complete change of UI to better suit the culture as it was not accepted by the users in the arms industry. While some RTD can be carried on, some cannot and will need to be fixed with extra cost.

4.2 Threats to validity

This study does have its own limitations and faces validity challenges.

4.2.1 Construct Validity. Construct validity refers to the extent to which the study accurately measures the theoretical construct it intends to measure. The study's construct validity hinges on how well "veracity" was defined and explained to participants. If participants struggled to understand and relate to the concept, the data collected might not accurately reflect their true perceptions and experiences. Their understanding and interpretation of the concept might vary widely, affecting the consistency and reliability of the data.

Table 6. Mapping of the cultural veracity RTD incidents to the model concepts [7]

Concepts in the conceptual model [7] relevant to Quantification of Veracity RTD	Incident 1	Incident 2	Incident 3
User Need	x	x	x
Requirements Engineering Step (Documented) Requirement	x	x	x
Functional Requirement	x	x	
Non-Functional Requirement			x
RTD Item	x	x	x
RTD Rectifying Step	x	x	x
Context			x
Risk			
Scenario			
Quality Attribute (Prioritized) Requirement			
Design Step			
Design/ Architectural Decision			
Implementation cost of Architectural/ Design decision			
Total Cost of a RE Step			
Cost of Rectifying RTD	x	x	x
RTD Interest	x	x	x
New Code Cost associated with RTD			x
Rework Code Cost associated with RTD	x	x	
New Design Cost associated with RTD			
Rework Design Cost associated with RTD			
New RE Cost associated with RTD			
Rework RE Cost associated with RTD			
Documentation cost of a requirement			
Benefit of Rectifying	x	x	x
Benefit of taking RTD			
Benefit of not taking RTD			
Product Value			x
End user satisfaction level	x	x	x
RTD Interest Probability			
RTD Item Priority			
Priority of a requirement			

4.2.2 Internal Validity. Internal validity refers to the extent to which the design and conduct of the study accurately measure the intended variables and minimize the impact of confounding factors. The study's small sample size (8 interviews) can limit the ability to identify consistent patterns. With only a few participants, individual differences can significantly impact the results. One-hour interviews may not be sufficient to deeply explore complex topics, potentially limiting the depth of data collected.

4.2.3 External Validity. External validity is the extent to which the results of the study can be generalized to other settings, populations, and times. Most participants are from New Zealand, which may limit the generalizability of the

findings to other cultural or geographic contexts. The cultural context of New Zealand might influence the participants' perceptions and interpretations of "veracity".

4.3 Implications for researchers and practitioners, Future Work

The main outcome of this study is that cultural veracity in software systems is something that is valuable for the users and the developers of the system. The "User specific requirements" is very important in today's context where user-centered products are being developed. The RTD model illustrated how cultural veracity RTD relates to the different concepts and the relationships among them while "User Need" being the center of all.

In future work, we would like to investigate more into practitioners' experiences with cultural veracity RTD and how different tools and techniques are being used to help quantify and make informed decisions. This can lead to more adaptations of the RTD model specific to cultural veracity.

5 Conclusion

In conclusion, the integration of cultural veracity in software products is essential for creating meaningful and authentic user experiences. As the global software market continues to expand, the importance of cultural sensitivity and authenticity will only grow. While this study addresses the meaning, impacts and the benefit of quantifying (formally and informally), future research should expand on these findings with larger and more diverse samples, exploring the impacts of cultural veracity across different cultural settings and software domains. When software accurately reflects and respects cultural nuances, users are more likely to feel understood and valued.

Acknowledgments

This research was funded by the New Zealand Ministry of Business, Innovation and Employment via The Science for Technological Innovation (SfTI) National Science Challenge Veracity Technology Spearhead.

We want to thank Associate Professor Māui Hudson for his feedback on the section of the interview transcript on cultural veracity and Ernestynne Walsh for the wānanga held on cultural veracity.

References

- [1] Virginia Braun and Victoria Clarke. 2006/04//. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (Apr 2006 2006/04//), 77–101. <https://ezproxy.auckland.ac.nz/login?url=https://www.proquest.com/scholarly-journals/using-thematic-analysis-psychology/docview/223135521/se-2> Copyright - © 2006 Arnold; Last updated - 2024-03-22.
- [2] Markus Luczak-Roesch, Matthias Galster, and Kevin Shedlock. 2023. The Veracity Grand Challenge in Computing: A Perspective from Aotearoa New Zealand. *Commun. ACM* 66, 7 (jun 2023), 67–69. <https://doi.org/10.1145/3589154>
- [3] Innovation Ministry of Business and Employment. 2023. Small business. <https://www.mbie.govt.nz/business-and-employment/business/support-for-business/small-business>
- [4] Natalia Ortega, Judith Perera, Matthias Galster, and Kelly Blincoe. 2024. Ensuring Trust and Integrity in Software Systems: Identifying, Integrating, and Measuring Veracity Requirements Technical Debt: A Systematic Literature Review. (2024).
- [5] Judith Perera, Ewan Temeporo, Kelly Blincoe, Matthias Galster, and Yu-Cheng Cheng. 2024. In-depth Conversations on Requirements TD and Veracity Requirements TD Quantification in the Industry. (2024). *In writing*.
- [6] Judith Perera, Ewan Tempero, Yu-Cheng Tu, and Kelly Blincoe. 2023. Quantifying Requirements Technical Debt: A Systematic Mapping Study and a Conceptual Model. In *2023 IEEE 31st International Requirements Engineering Conference (RE)*. 123–133. <https://doi.org/10.1109/RE57278.2023.00021>
- [7] Judith Perera, Ewan Tempero, Yu-Cheng Tu, and Kelly Blincoe. 2024. Modelling the Quantification of Requirements Technical Debt. *Requirements Engineering Journal* (2024), In press. <https://doi.org/10.1007/s00766-024-00424-3>
- [8] Judith Perera, Ewan Tempero, Yu-Cheng Tu, Kelly Blincoe, and Matthias Galster. 2024. Towards Quantifying Requirements Technical Debt for Software Requirements concerning Veracity: A Perspective and Research Roadmap. arXiv:2407.00391 [cs.SE] <https://arxiv.org/abs/2407.00391>
- [9] P Runeson, M Host, and A Rainer. [n. d.]. *Case Study Research in Software Engineering : Guidelines and Examples*. John Wiley & Sons, Incorporated, Newark. Ebook Central.