

MQM-Fin: A Risk-Weighted Hybrid Quality Assurance Framework for Localizing Financial Software - Beyond LISA and SAE J2450

Sina Alizadeh Tabrizi

`tabrizi.sina@student.atilim.edu.tr`

Department of Software Engineering, Graduate School of Natural and Applied Sciences, Atılım University, Ankara, Turkey <https://orcid.org/0000-0001-6785-675X>

Research Article

Keywords: Software Localization, Quality Assurance, LISA, MQM, SAE J2450

Posted Date: October 13th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-4803157/v2>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

MQM-Fin: A Risk-Weighted Hybrid Quality Assurance Framework for Localizing Financial Software - Beyond LISA and SAE J2450

Sina Alizadeh Tabrizi^{a,b}

^a *Department of Computer Engineering, Ur.C., Islamic Azad University, Urmia, Iran*

^b *Department of Software Engineering, Graduate School of Natural and Applied Sciences, Atılım University, Ankara, Turkey*

**For Correspondence: s.alizadehtabrizi@iau.ac.ir*

ABSTRACT

The problem of quality assurance (QA) of user interface texts in the software localization industry persists due to the lack of consensus among specialists. Three QA models, LISA QA Model, SAE J2450, and Multidimensional Quality Metrics (MQM) were tested in practice based on the English localization of a Persian banking software. It is found that MQM provides the finest granularity when classifying errors, such as accuracy issues (for example, mistranslation or omission) and fluency issues (such as grammar and spelling errors). At the same time, LISA merges with certain types of errors while SAE J2450 does not consider style or formatting issues. We suggest two methodological improvements. Firstly, MQM-Fin, which is a risk-weighted version of MQM in which the penalty of an error is multiplied by a function-based risk multiplier unique to financial software; secondly, a taxonomy of failure modes (syntactic isolation, terminological ambiguity, and phrase compression) that accounts for the reasons why Google translate generates mistakes in the process of translating isolated user interface strings through localization.

KEYWORDS: Software Localization; Quality Assurance; LISA; MQM; SAE J2450

1. Introduction

Nowadays, most software vendors endeavor to export their products to other countries. This mostly originated from the fact that all software companies are determined to be capable of competing in the world market.

The definition of localization, offered by Schäler (2010) and Gonçalves and Gonçalves (2023), gives a complementary understanding by considering localization as "the linguistic and cultural adaptation of digital content to the requirements and the locale of a foreign market," clearly stating the role of managing multilingualism within the digital global flow of information.

For Koby, Fields, Hague, Lommel, & Melby (2014): "the broad view supports the following definition of translation quality: a quality translation demonstrates accuracy and fluency required for the audience and purpose and complies with all other specification negotiated between the requester and provider, taking into account the end user needs" (p. 416).

According to the broad view, translation is not merely a text-centric activity, and this definition of translation quality also encompasses other tasks common to the language industry, such as summarization, transcreation, and localization. In contrast, a narrow definition of translation quality states that:

"A high-quality translation is one in which the message encrypted in the source text is transferred completely into the target text, including denotation, connotation, nuance, and style. The target text should be written in the target language using correct grammar and word order, producing a culturally appropriate text that, in most cases, reads as if originally written by a native speaker of the target language for readers in the target culture" (Koby, Fields, Hague, Lommel, & Melby 2014).

For Kinds (2000), "Translation quality is difficult to judge, let alone quantify, and the question of how to measure it has long been subject to debate" (p. 147). Most of the issues related to translation quality concern the quality of the target text, which may not meet the client's expectations. Therefore, "to determine whether someone has attained translation quality, one must be able to measure it" (Koby, Fields, Hague, Lommel, & Melby 2014). This is also true in the localization industry, where the quality of localized software and QA models are diverse. "The quality measurement of a localized product is performed via testing procedures" (Hagan and Ashworth, 2000).

For the quality assurance of localized software, there are three QA manual tools, none of which are flawless. Our main focus will be on clarifying the pros and cons of these three models: LISA, SAE J2450, and MQM. This study aims to compare these models and assess the capability of Google MT for translating short terms (strings), which are common in the graphical user interface (GUI) of software. A challenge arises from the fact that, among the numerous ISO standards and other automated QA models, determining and selecting the most suitable and universally applicable one for a wide range of software has always been difficult for LSPs.

This study aims to apply the MQM framework, the LISA QA Model, and SAE J2450 to financial localized software to determine the most suitable QA model. These QA models are quantitative and seek to minimize the subjectivity involved in the quality assessment of translation. As Kit and Wong Kit (2008) state, "Briefly put, language is full of exceptions and ambiguities at all linguistic levels.

2. Related Works

With regard to Skopos theory, a product resulting from the translation process is called a *translatum*, a specific type of target text. Different Skopos (goals) of the translation action leads to different variants of the source text (ST). As an expert, the translator is faced with various goals and commissions during the translation process. So, they are obliged to choose the translation strategy based on the pre-determined Skopos.

No action has an inherent function; the function is selected from among the available and possible choices and then attributed to the action. Seemingly, a functional approach is more applicable to pragmatic texts and IT industry products. As Albir (2001) states, “Anyone who adopts the Skopos theory in their translational action should be cognizant of the commission” (p. 34), which should include, or at least be aware of, as much relevant information as possible on the following aspects:

1. The goal, i.e., the specification of the aim of the commission.
2. The conditions under which the intended goal should be reached, which naturally include practical matters such as deadlines and fees.

Berger (2015) summarizes the quality assurance process as Figure 1:

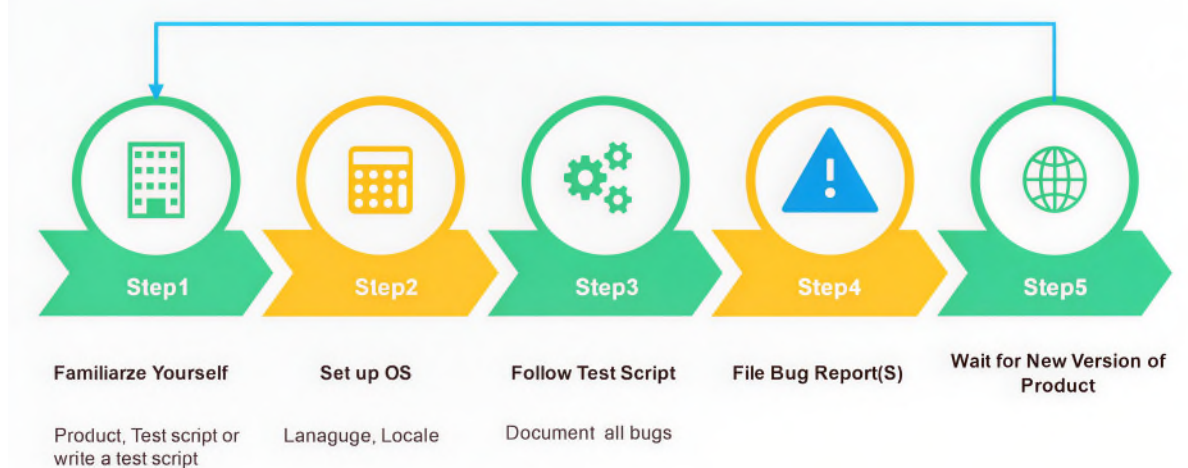


Figure 1. Quality assurance process flow.

Most of the QA models and metrics are concocted to obviate the subjective judgment and measure the extent of client's satisfaction via a series of metrics and criteria. Similar to the translation, “there exist no consensus on what precisely constitutes localization” (Dunn 2000). For Dunn (2000), “there is at present no de jure or de facto localized process quality or product quality standard”.

Manish Bhatia and Sharma (2013), in their article “A Generalized Quality Model for Localized Software Products”, discuss several existing quality models, including the McCall Quality Model, Boehm Quality Model, and LISA QA Model. They also introduce a new Generalized QA Model that combines the advantages of the preceding models. In their model, the “higher priority factors are its correctness, acceptability, accuracy, terminology, cosmetic quality, usability, and understandability” (p. 45).

Moreover, in a different research study, Gonçalves & Gonçalves (2023) performed a comparative analysis to investigate terminological consistency between OpenOffice Writer and

LibreOffice Writer, using a total of 907 terms that were selected from the graphical user interface of both programs. It was found that, out of these 907 terms, although 86% of them reflected terminological consistency, yet inconsistency occurred in 14%.

In an article entitled "*Mission Impossible: Improve Quality, Time, and Speed at the Same Time*", GM's Don Sirena, language translation manager at General Motors of North America, reported a 90% reduction in translation errors after GM adopted the SAE J2450 quality metric. He states, "When QA is seen as central to translation and localization efforts, SAE J2450 (and other metrics, like LISA's QA Model 3.0) can deliver benefits that far exceed improvements in quality" (Cronin 2003).

In their 2019 work, Wang, Chen, and Xing successfully filled a significant gap within software localization studies by creating a domain-specific neural machine translation engine that can translate the texts of mobile applications. They noted the inadequacy of generic machine translation tools, which do not possess domain-specific terminological knowledge or writing style.

Fisher (2026) studied the effect of domain-specific fine-tuning of neural machine translation models applied to Georgian religious documents with NLLB-200 serving as the baseline system. He trained two LoRA-adapted systems on Georgian religious texts and the Bible, respectively. In a study conducted by Wibowo et al. (2019), the focus was centered on reconfiguring localization quality assessment (LQA) for video games, and scholars claimed that there is a need for reconstruction of LQA from eco-criticism perspectives. They believe that eco-LQA is practice revealing whether certain games suffer from file size uplift and extra power consumption.

Li et al. (2021) developed evaluation criteria and techniques used in the design of application software localization, using run environment dependency analysis technology. The technique employed by Li et al. integrates sophisticated validation testing in ten areas including functionality, security, dependability, compatibility, scalability, performance, resource occupancy, usability, user documentation, and user satisfaction. Shalawati et al. (2026) examined the potential use of LLMs as evaluators in the context of multidimensional machine translation assessment, in terms of the English-Indonesian translation pair. Using a rubric aligned to the MQM framework that measured translation quality in morphosyntactic, semantic, and pragmatic aspects, three LLM-powered machine translation systems (Qwen 3, LLaMA 3.2, and Gemma 3) were assessed with evaluations provided by human experts and GPT-5 as an evaluator.

Lommel et al. (2013) concluded preliminary tests on MQM, which demonstrated the availability of a wide variety of quality metrics within an error-count environment. These metrics are successfully used for quality assessment of both human and machine translation. Li et al. (2024) introduced MQM for Chat Translation, and they included seven error categories of this metric. They concluded that MQM-Chat was very efficient in classifying the errors, and it also highlighted chat-specific issues. Recently, Qian et al. (2023) proposed a new evaluation framework based on MQM to assess the Chinese-English Machine translation of emotion-loaded microblog texts, and the used error categories of MQM and they claim that they have adopted error categories for evaluating the emotional errors. Song et al. (2025) evaluated three notation setups in machine translation, these notations are: MQM, side-by-side (s x s) MQM and its simplified versions s x s relative ranking (RR). They observed that the s x s settings led to higher inter-annotator agreement than MQM, while s x s MQM boosts the inter-translation error consistency relative to MQM. A study on quality of text content in business websites were done by Lu (2024), which was a corpus-based survey, which was translated from Spanish to Chinese. They adopted models of analysis of functional moves in digital genres adapted from Swalesian genre theory. Jiménez-Crespo (2011)

focused on current localization quality assessment practices, by conducting a product-based analysis of error topologies. Another study on quality in software localization using machine translated user interface is conducted by Qin et al. (2017), and they showed that users are not satisfied with graphical user interface (GUI) localized using MT, and the major complaints about these GUIs were: misleading labels of input boxes and unclear translation items in option lists.

Mariana (2014) examined MQM framework by setting the severity of its category to mirror, as closely as possible, the American Translators Association's (ATA) translator certification exam. As a corpus, a set of 29 student translations was studied. Li et al. (2021) proposed the evaluation standard and method of application software localization designed based on run environment dependency analysis technology. Nedungadi (2022) was another scholar who focused on the most recent quality assurance practices used in the sphere of internationalization and localization for open-source software. Omar (2021) focused on localization quality of Learning Management Systems (LMSs) showed that translation inconsistencies are the main problems with Arabic version of the different LMs including Blackboard Learn, Microsoft Teams and Zoom.

Compared to the preceding studies, the present research adopted an empirical approach, through which quality assurance models are applied to concrete samples of localized software. In addition, from a theoretical perspective, this research contributes to the body of knowledge by extending the current framework of software localization quality assurance by developing MQM-Fin, which is a financial risk-weighted extension of MQM metrics to consider the functional risk level (low, medium, high, critical) together with the linguistic error type. Another contribution from a theoretical standpoint is the development of the first taxonomy of failure modes for machine translation in user interface localization in terms of Syntactic Isolation, Terminological Ambiguity, and Phrasal Compression.

3. Methodology

The backbone of this research is based on a quantitative approach, while it is also necessary to derive a series of comments and judgments about the QA models and Google MT based on the figures and numerical values gathered using LISA, SAE J2450, MQM, and MemoQ software. Therefore, a qualitative approach is also an integral part of the study, and a mixed-methods approach has been adopted by the researcher.

The research method employed in this study is primarily quantitative, with the researcher's ideas and hypotheses being formed based on calculable and measurable categories. To put it simply, applying any quality assurance model or criteria to localized products involves figures and numerical values that reflect the degree of approval or disapproval of the product. The qualitative method is an integral part of such studies, as achieving the study objectives relies on the insights provided by the researcher's comments. In fact, quantitative data can be considered raw data, and thus, qualitative statements on the data are indispensable.

In this research, the main attempt and goal was to answer the following question:

What is the most appropriate QA Model for quality assurance of linguistic aspect of the localized financial software?

3.1. Theoretical framework

According to Skopos Theory, "a TT, called *translatum* by Vermeer, must be fit for purpose; that is, it must be functionally adequate" (Munday 2022). Furthermore, "the target-text purpose should be compatible with the original author's intentions" (ibid, p. 124). For Vermeer, any action without a Skopos is meaningless. He views translational actions as purposeful endeavors that "lead to a *translatum*" (Munday, 2022). A *translatum* is regarded as a specific type of target text, suggesting that any source text (ST) can have multiple translations, depending on the pre-defined and pre-determined Skopos agreed upon through a negotiation between the client and the translator.

According to Venuti, "skopos and mode of realization must be adequately defined if the text-translator is to fulfill his task successfully" (Venuti 2012, 192). As a final note, Nord (2006) states that for translation of a ST there are more than one method and, as results multiple target texts can be created. The solution for translation problem is determined with regard to set of criteria, "in the case of functional approaches, this criterion is the communicative function or functions for which the target text is needed".

3.2. Corpus

In this research, localized version of two financial software developed by the author will be assessed via QA models. The Banking Software was designed in Persian and its localized version is in English. It is worth noting that the Persian version of the software is based on Islamic banking approach, which offers a facilities based on the Sharia financial principles, and the localized version is planned to be used in U.S.A by one of Iranian Bank branch, like Saderat Bank, which is based on Islamic Banking approach. As a result, the localized version should thoroughly satisfy the end-user's expectations.

All the English texts used in the GUI of the banking software application were created via Google Translate (online free version, used in February 2026). There was no post-editing carried out for the machine translation output. Then, the machine-translated texts were analyzed according to the three quality assessment frameworks against the source text in Persian. There was no post-editor employed for the assessment.

3.3. Applied Models

The QA models used in this study are LISA, SAE J2450 and MQM. The numerical figures obtained will be fed into MemoQ and GEVTerm as an input in order for calculation of overall score.

3.3.1 LISA QA Model

This model is prevalent in localization industry, and it consists categories, subcategories and severity levels, as well as negative points. That is, it has seven predefined testing tasks, with errors assigned to 26 predefined categories. The severity levels are:

- **Critical** (10 points)
- **Major** (5 points)
- **Minor** (1 point)

This model's error categories are accuracy, mistranslation, language, terminology, language, style, and potential error categories which are country specific, as well as consistency. To assign penalty points, 1 is reserved for minor errors, 5 for major errors and 10 for critical errors.

3.3.2. SAE J2450

With regard to Table 1, the J2450 metric includes seven different categories of translation error types, which are defined primarily in terms of terminological accuracy and grammatical correctness. “Errors of style are explicitly excluded from the metric, as are formatting errors”. Error types are further categorized based on severity level, e.g., “serious” or “minor.” A weight is assigned to each error type and severity level in the form of a numerical value. A quality translation is one that presents fewer errors, and thus a lower weighted numeric score.

The error categories associated with SAE J2450 are as below:

- Wrong Term
- Syntactic Error
- Omission
- Word Structure or Agreement Error
- Misspelling
- Punctuation Error
- Miscellaneous Error

Translation quality assessment performed using the J2450 metric reflects a measurement of the relative equivalence of surface linguistic attributes of the source and target texts. As the J2450 standard notes, the definitions of error categories depend upon the surface form of the translation and are generally divorced from the meaning. The presumption of functional equivalence in the SAE J2450 metric, along with the exclusion of the end-user and context of use as operational variables in the assessment of translation quality are justified by the nature of the problem domain (the communication of automotive service information across languages), the context of use (performing automotive service operations), as well as the profile of the target customer (the service technician), all of which are explicitly specified in the metric. It should be mentioned that the process of assigning the error categories and calculation for SAE J2450 was accomplished using MemoQ.

Table 1. SAE J2450 Error Categories and Calculations (Dunn 2000)

Error Type	Serious Multiplier	Minor Multiplier	Formula
Wrong Term Score (WT)	× 5	× 2	$WT = (Serious \times 5) + (Minor \times 2)$
Syntactic Error Score (SE)	× 4	× 2	$SE = (Serious \times 4) + (Minor \times 2)$
Omission Score (OM)	× 4	× 2	$OM = (Serious \times 4) + (Minor \times 2)$
Word Structure/Agreement Score (SA)	× 4	× 2	$SA = (Serious \times 4) + (Minor \times 2)$
Misspelling Score (SP)	× 3	× 1	$SP = (Serious \times 3) + (Minor \times 1)$
Punctuation Error Score (PE)	× 2	× 1	$PE = (Serious \times 2) + (Minor \times 1)$
Miscellaneous Error Score (MF)	× 3	× 1	$MF = (Serious \times 3) + (Minor \times 1)$

3.3.3. MQM (Multidimensional Quality Metrics)

MQM defines over 100 issue types derived from major quality assessment systems, both automated and manual. These issue types represent a comprehensive set of potential translation problems, organized hierarchically, with more specific issues categorized as subtypes of broader categories. For example, “Mistranslation” is a subtype of the broader “Accuracy” issue type. The framework provides detailed hierarchy and descriptions of these issue types. The seven top-level error dimensions are illustrated in Figure 2.

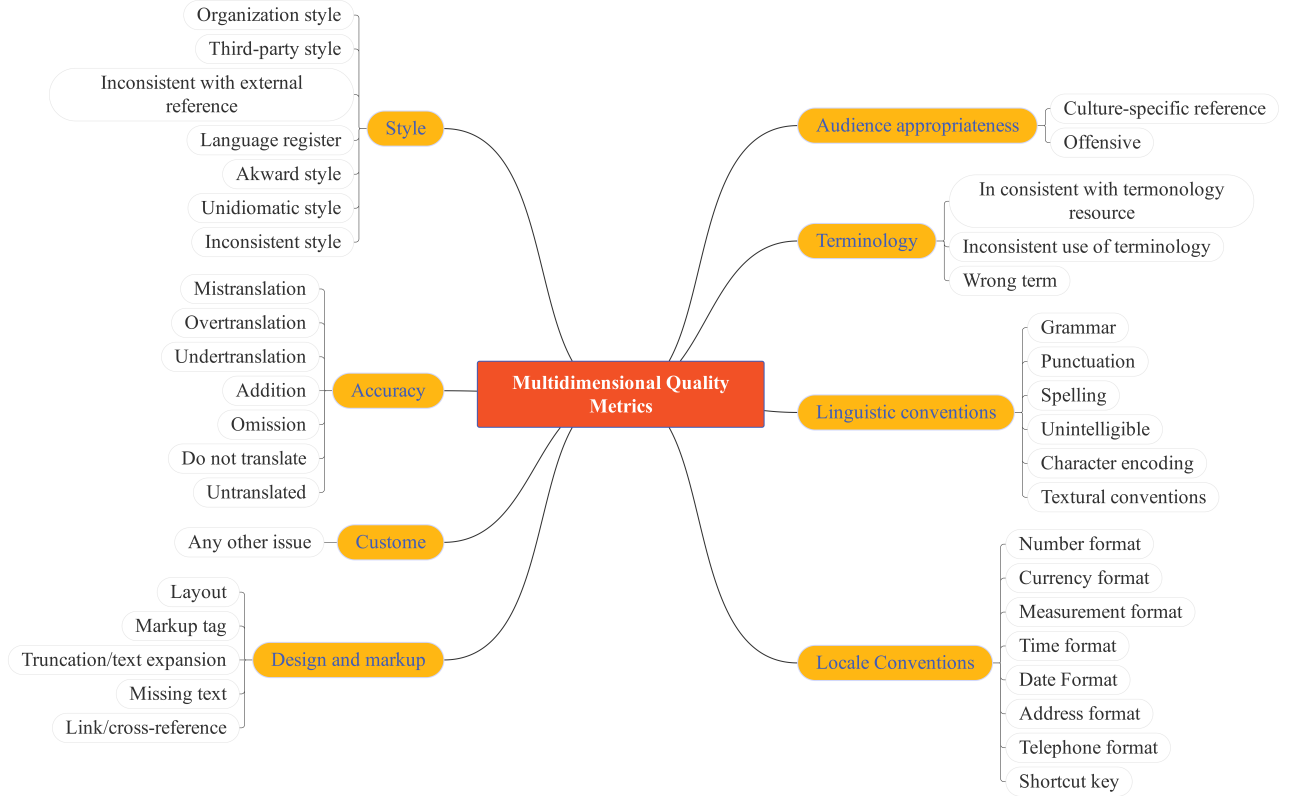


Figure 2. Seven top level error categories of MQM from MQM Review (Burchardt 2013).

3.4. Score calculations

Quality scores for each of the three models are determined by:

- Finding translation errors and giving them a weight (severity $\times$ category).
- To determine the overall error score, add up the weighted error points.
- Normalize the result (usually by dividing by the word or segment count) to make scores comparable across texts.

LISA QA Model score calculations:

- Error Categorization: Mistranslations, terminology, style, formatting, and omissions are among the categories into which errors are divided into minor, central, and critical categories.
- Weighting: Each severity has a pre-defined penalty score (e.g., minor = 1, major = 5, critical = 10).
- Computation is done using Eq. (1), as below:

$$E = \sum (w_{c,severity} \times n_c) \quad (1)$$

where n_c = number of errors of that severity in category c .

- Normalization: *LISA Score per 1000 words* = $(E/N) \times 1000$

SAE J2450 score calculations:

- Error Categorization: Seven categories (Wrong Term, Syntactic Error, Omission, Word Structure/Agreement, Misspelling, Punctuation, Miscellaneous).

- Severity Levels: Each error is rated as minor or severe, with weights (e.g., Wrong Term = 2/5, Misspelling = 1/3).
- Computation is done using Eq. (2), as below:

$$E = \sum (w_{c, \text{minor}} \times m_c + w_{c, \text{serious}} \times s_c) \quad (2)$$

- **Normalization:**

$$\begin{aligned} J2450 \text{ Score} &= E / N \\ J2450 \text{ Score per 1000 words} &= (E / N) \times 1000 \end{aligned}$$

MQM Model score calculations:

- Error Categorization: Errors are classified along dimensions (accuracy, fluency, terminology, style, locale, etc.) and severity (minor, major, critical).
- Weighting: A configurable penalty value (typically minor = 1, major = 5, critical = 10) is assigned to each severity level.
- Computation is done using Eq. (3), as below

$$E = \sum (w_{c, \text{severity}} \times n_c) \quad (3)$$

- Normalization: MQM often reports a quality ratio instead of raw error points:

$$MQM \text{ Score (\%)} = 100 \times (1 - (E / N)), \quad (4)$$

where N = quantity of words or segments assessed. Certain implementations also report error points per 1000 words. A higher MQM percentage indicates better quality.

4. Results

The three models, LISA QA Model, SAE J2450, and MQM, were employed as tools to assess the translation quality of the English strings in the banking software GUIs, which were generated using Google Machine Translation. The categorization of errors for each model is presented in Tables 2, 4, and 6, respectively.

Table 2. Error categorization of Banking software based on LISA metrics

Login GUI		
Problematic String	Error Category	Severity
account name	Mistranslation	Major
Password	Accuracy (Omission)	Major
Arrival	Mistranslation	Major
Main Menu GUI		
Legal	Mistranslation	Minor
National ID	Terminology	Minor
Last name, father's name, saving, account number, marital status:	Language(punctuation)	Minor

Availability	Terminology	Minor
Bill	Accuracy(omission)	Major
Remember	Consistency	Major
Education	Terminology	Minor
Load GUI		
Bond loan	Terminology	Minor
Reward	Terminology	Minor
Information toggle	Mistranslation/ language (grammar: wrong word order)	Minor
All lease	Accuracy (addition)	Minor
Information sheet treatment, phone hospital	Language (grammar: wrong word order: syntactic error)	Minor
Completion Date	Terminology	Minor
Cheque Services GUI		
Services Category Czech, Czech export markets, Former Czech serial category	Mistranslation	Major
account number	Language (Punctuation)	Minor
Issuance of new Czech	Language (grammar)	Minor
Approved	Accuracy (Addition)	Minor

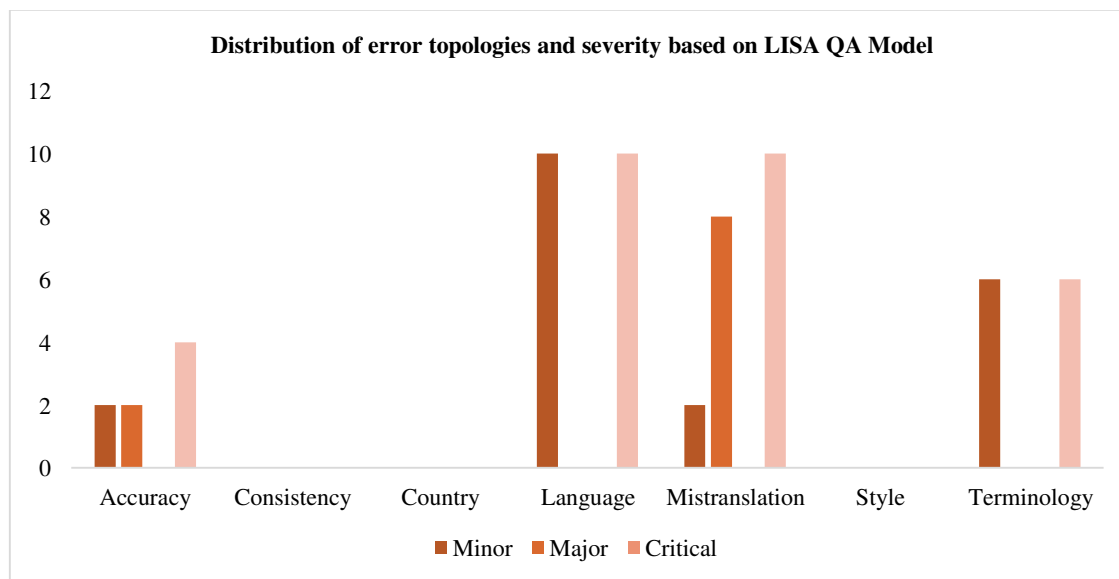


Figure 3. Error topologies frequency based on LISA QL Model

Figure 3 illustrates that focus improvement efforts on mistranslation and accuracy are needed, especially where semantic precision matters. Refinement and review should be targeted toward language fluency and terminology standardization.

Table 3 presents the overall score of the localized strings of banking software based on LISA QA Model which is calculated using MemoQ.

Table 3. Overall score of the localized strings of banking software based on LISA QA Model

Error Type	Accuracy	Consistency	Country	Language	Mistranslation	Style	Terminology	Total
Minor	2	0	0	10	2	0	6	18
Major	2	0	0	0	8	0	0	10
Critical	0	0	0	0	0	0	0	0
Total	4	0	0	10	10	0	6	30
Result: Fail								

Table 4. Error categorization of Banking software based on SAE J2450 QA Model

Login GUI		
Problematic String	Error Category	Severity
account name	Wrong Term (WT)	Serious
Password	Omission	Serious
Arrival	Wrong Term (WT)	Serious
Main Menu GUI		
Legal	Wrong Term (WT)	Minor
National ID	Wrong Term (WT)	Minor
Last name, father's name, saving, account number, marital status:	Punctuation	Minor
Availability	Wrong Term (WT)	Minor
Bill	Omission	Serious
Remember	Wrong Term (WT)	Serious
Education	Wrong Term (WT)	Minor
Loan GUI		
Bond loan	Wrong Term	Minor
Reward	Wrong Term	Minor
Information toggle	Wrong Term (WT)/ Syntactic(word linear order)	Minor
All lease	Miscellaneous	Minor
Information sheet treatment, Phone hospital	Syntactic(word linear order)	Minor
Completion date:	Wrong Term (WT)	Minor
Cheque Services GUI		
Services Category Czech, Czech export market, Former Czech serial category	Wrong Term (WT)	serious
account number	Punctuation	Minor
Issuance of new Czech	Syntactic (wrong part of speech)	Minor
Approved	Word Structure (Morphological)	Minor

Table 5 presents the overall score of the localized strings of banking software based on SAE J2450 QA Model which is calculated using MemoQ.

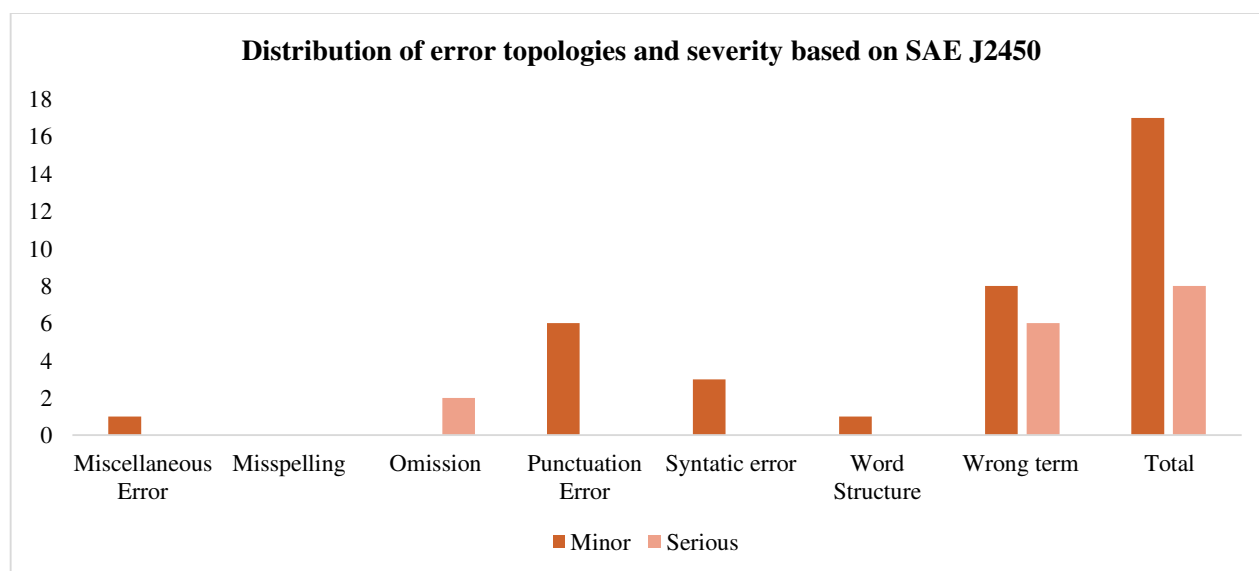


Figure 4. Error topologies frequency based on SAE J2450

According to Figure 4, the trend shows that during the localization process, there has been numerous errors related to the terminologically inconsistent outputs. To this end, in localization industry, there is a need for enhancement in glossary adherence, implementation of terminology databases for specific domains, and integration of AI-based technologies or even Large Language Models into QA models to automation the process, particularly for punctuation and syntax errors. While the number of errors for omissions and misspellings are low, but still there is a room for optimization of the target text.

Table 5. Overall score of the localized strings of banking software based on SAE J2450 QA Model

Category	Minor	Serious	Total Points
Miscellaneous Error	1	00	1
Misspelling	0	0	0
Omission	0	2	8
Punctuation Error	6	0	6
Syntatic error	3	0	6
Word Structure	1	0	2
Wrong term	8	6	46
Total	17	8	65
Result: Fail			

Table 6. Error categorization of Banking software based on MQM Model

Login GUI		
Problematic String	Error Category	Severity
account name	Accuracy(mistranslation)	Major
Password	Accuracy(omission)	Major
Arrival	Accuracy(mistranslation)	Major
Main Menu GUI		
Legal	Accuracy (mistranslation)	Minor
National ID	Accuracy (terminology)	Minor

Last name, father's name, saving, account number, marital status:	Fluency(spelling: capitalization)	Minor
Availability	Accuracy(Terminology)	minor
Bill	Accuracy (Omission)	Major
Remember	Accuracy (mistranslation)	Major
Education	Accuracy (terminology)	Minor
Loan GUI		
Bond loan	Accuracy (Terminology)	Minor
Reward	Accuracy (Terminology)	Minor
Information toggle	Accuracy (mistranslation) & Fluency (grammar)	Minor
All lease	Accuracy (addition)	Minor
Information sheet treatment, Phone hospital	Fluency (grammar)	Minor
Completion date:	Accuracy (Terminology)	Minor
Chequebook GUI		
Services Category Czech, Czech export market, Former Czech serial category	Accuracy (mistranslation)	Major
account number	Fluency (spelling: capitalization)	Minor
Issuance of new Czech	Fluency (grammar)	Minor
Approved	Accuracy (Addition)	Minor

Figure 3 presents the overall score of the localized strings of banking software based on MQM Model which is calculated using MemoQ.

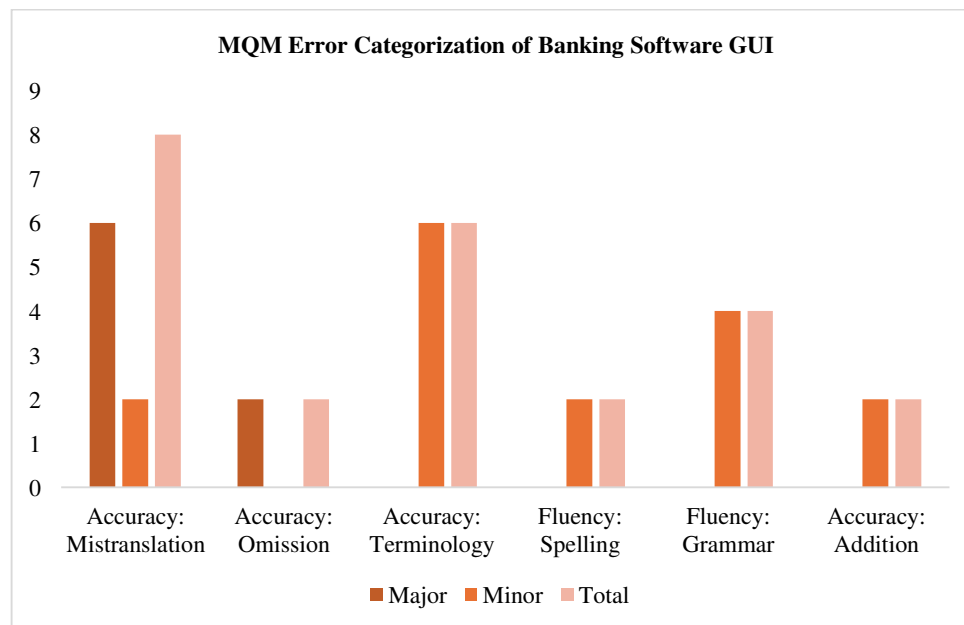


Figure 5. Overall score of localized banking software based on MQM model

As figure 5 illustrates, there is a room for improvement of translation quality, since the calculated quality score based on MQM model for localized GUIs of the study software is 56.4%. In fact, the aspects which needs improvement are Mistranslations, which has the highest number of most major errors. Then, it is followed by Omissions, which had a great risk for the overall

quality of the translated strings. Moreover, there is a need to address other minor errors in spelling, grammar and terminology, which can make substantial contribution to overall quality score. Table 7, 8, 9 and 10 presents a summary of applying each QA model on the Login, Main Menu, GUI of Banking software.

Table 7. Comparison between Error Categorization of Login GUI by LISA, SAE J2450 and MQM Models

Problematic String	LISA QA Model		SAE J2450		MQM	
	Error Category	Severity	Error Category	Severity	Error Category	Severity
account name	Mistranslation	Major	Wrong Term (WT)	Serious	Accuracy(mistranslation)	Major
Password	Accuracy(Omission)	Major	Omission	Serious	Accuracy(omission)	Major
Arrival	Mistranslation	Major	Wrong Term (WT)	Serious	Accuracy(mistranslation)	Major

With regard to Table 7, it is evident that all errors are captured and detected by each QA model with the same severity level. However, the errors, based on QA models, are categorized differently. According to Tables 4.5 and 4.13, the translation of "نام کاربر" as "account name" is categorized under the same error category in LISA and MQM. However, in SAE J2450, it is categorized as a Wrong Term error.

Moreover, the translation of "بازیابی رمز" as "Password" is classified under the Accuracy category in both LISA and MQM, which suggests that these two QA models are more precise in error classification compared to SAE J2450. It is evident that most errors in Table 4.29 originate from the lack of contextual information required by Google MT to generate accurate raw translations.

The assessment of the Login GUI and the results obtained indicate that the MQM model is more capable of identifying the nature of errors (e.g., accuracy or fluency), making it distinctive among the other models.

It is interesting to note that the localized version of the Login GUI (produced by Google Translate) fails to meet end-user needs, primarily in terms of accuracy. Essentially, it is non-functional in the target culture due to accuracy issues rather than fluency. The origin of such failures is linked to the lack of contextual information, which should be provided for Google Translate. Hence, Google Translate should incorporate a field where translators and localization engineers can input contextual information for GUI strings, which would likely result in more meaningful outputs.

Table 8. Comparison between Error Categorization of Main GUI by LISA, SAE J2450 and MQM Models

Problematic String	LISA QA Model		SAE J2450		MQM	
	Error Category	Severity	Error Category	Severity	Error Category	Severity
Legal	Mistranslation	Minor	Wrong Term (WT)	Minor	Accuracy (mistranslation)	Minor
National ID	Terminology	Minor	Wrong Term (WT)	Minor	Accuracy (terminology)	Minor
Last name, father's name, saving, account number, marital status:	Language(punctuation)	Minor	Punctuation	Minor	Fluency(spelling: capitalization)	Minor

Availability	Terminology	Minor	Wrong Term (WT)	Minor	Accuracy(Terminology)	minor
Bill	Accuracy(omission)	Major	Omission	Serious	Accuracy (Omission)	Major
Remember	Consistency	Major	Wrong Term (WT)	Serious	Accuracy (mistranslation)	Major
Education	Terminology	Minor	Wrong Term (WT)	Minor	Accuracy (terminology)	Minor

According to Table 8, the translation of "صورت حساب" as "Bill" was categorized under the Omission category in SAE J2450, while in MQM and LISA, it was regarded as an Accuracy error with the subcategory of Omission.

It should also be noted that the translations of "last name," "father's name," "saving," and "account number" are classified under different error categories by each QA model. For example, LISA QA model metrics consider the afore-mentioned errors as Language Errors (spelling), which these are labelled as Punctuation Errors. In terms of MQM Framework, these errors are of Fluency Errors Type, falling into the subcategory of Spelling. All the discrepancies are due to capitalization errors and diacritic faults. Seemingly, MQM excel LISA and SAE J2450 QA Models, in categorization of these errors, as it classifies the exact error type (e.g., capitalization).

Considering Table 8, Terminology is the most frequent error type observed in the Main GUI based on LISA QA Model, while the highest number of the errors on the same menu of the software based on SAE J2450 is related to Wrong Term Category. Considering the MQM Model, the highest number of the error type is related to Accuracy errors, both in terms of mistranslation, terminology and omission, which clearly shows how exactly MQM identifies the error nature, and highlights the Google MT inability in terms of accuracy. Although all models have their strengths, the MQM model is the most effective for clear identifying the nature of errors.

Table 9. Comparison of Error Categorization in Loan GUI by LISA, SAE J2450, and MQM Models

Problematic String	LISA QA Model		SAE J2450		MQM	
	Error Category	Severity	Error Category	Severity	Error Category	Severity
Bond loan	Terminology	Minor	Wrong Term	Minor	Accuracy (Terminology)	Minor
Reward	Terminology	Minor	Wrong Term	Minor	Accuracy (Terminology)	Minor
Information toggle	Mistranslation/ language (grammar: wrong word order)	Minor	Wrong Term (WT)/ Syntactic(word linear order)	Minor	Accuracy (mistranslation) & Fluency (grammar)	Minor
All lease	Accuracy (addition)	Minor	Miscellaneous	Minor	Accuracy (addition)	Minor
Information sheet treatment, phone hospital	Language(grammar: wrong word order: syntactic error)	Minor	Syntactic(word linear order)	Minor	Fluency (grammar)	Minor
Completion Date	Terminology	Minor	Wrong Term (WT)	Minor	Accuracy (Terminology)	Minor

With regard to Table 9, the QA models (LISA, SAE J2450, and MQM) have detected the problematic translations of "وام ديعة" and "جعاله" using relatively the same error categories and equal degrees of severity. The LISA and MQM models are nearly identical in their categorization of terminology errors; however, in LISA, the Terminology category is independent, whereas in MQM, Terminology is considered a subcategory of Accuracy.

Seemingly, MQM is more precise in error categorization, and it is so helpful in choosing an appropriate QA model, as a reviewer first will read the translation brief (localization specification) and choose the QA model based on the focuses and demands of the client. It can be

construed that in some cases, a single error can be captured by more than one category of LISA QA model. A prime example is "Information toggle," which can be classified as both a Mistranslation and a Language error. However, unlike SAE J2450, the LISA QA model assigns the same degree of severity to all errors, categorized as minor, major, or critical.

Similar to the LISA QA model, the MQM model classifies the above-mentioned string under Accuracy, with Mistranslation as a subcategory. Additionally, the problematic translation can also be included under another subcategory: Fluency – Grammar.

Unlike MQM and LISA, SAE J2450 does not have a precise error category for "All lease" (translated as "اطلاعات اجاره نامه"). Simply put, SAE J2450's designers did not specify an error category for Addition, whereas both MQM and LISA do.

Only in the case of the "Completion date" error do all three QA models behave similarly, assigning the same error category to the issue.

If the client's primary concern is a functionally accurate target text (TT) rather than fluency, the localized version of the Loan GUI could be considered a failure.

Table 10. Comparison between Error Categorization of Cheque GUI by LISA, SAE J2450 and MQM Models

Problematic String	LISA QA Model		SAE J2450		MQM	
	Error Category	Severity	Error Category	Severity	Error Category	Severity
Services Category Czech, Czech export markets, Former Czech serial category	Mistranslation	Major	Wrong Term (WT)	serious	Accuracy (mistranslation)	Major
account number	Language (Punctuation)	Minor	Punctuation	Minor	Fluency (spelling: capitalization)	Minor
Issuance of new Czech	Language (grammar)	Minor	Syntactic (wrong part of speech)	Minor	Fluency (grammar)	Minor
Approved	Accuracy (Addition)	Minor	Word Structure (Morphological)	Minor	Accuracy (Addition)	Minor

According to Table 10, in both MQM and LISA, the translation of "تایید" as "Approved" has been detected under identical error categories. MQM and LISA the translation of "تایید" as "Approved" has been detected by the identical error categories, while in SAE J2450 there is a special category for such errors, called Word Structure (encompassing incorrect morphological form, case, gender tense, prefix, suffix, infix or any other inflection). The localized version of Chequebook GUI suffers from Accuracy and Fluency problems, as a result the raw translation of the strings of this menu should be thoroughly post-edited.

Two last columns of table, refers to the error categorization by MQM model, providing insights into the post-editing phase, the MQM is followed by LISA QA model, in providing a precise information on error types and nature. Ultimately, SAE J2450 has only classified the fluency errors in more exact manner, compared to the accuracy errors classification of SAE J2450. The localized version of this GUI by Google MT should not be expected to meet the end-user needs, unless it is post-edited.

4. Enhancement in MQM

4.1. MQM-Fin: A Financial Risk-Weighted Extension of MQM

Whereas the MQM taxonomy is quite granular in its classification of error types, it does not distinguish the relative significance of individual errors of the same type at the level of severity. This means, for example, that a minor translation error in a financial application’s navigation bar might have no practical consequences, while a similar error in a transaction-related string could cause transaction failure and loss of money. With regard to Table 11, we aim to resolve this problem by creating a specialized variant called MQM-Fin, which adds a third dimension, Financial Risk (Low, Medium, High, Critical), as follows:

$$E_{\{MQM-Fin\}} = \sum (w_{severity} \times r_{risk} \times n_c) \quad (5)$$

Where, r_{risk} is defined as follows for the financial domain:

Table 11. MQM-Fin financial risk rubric

Risk Level	Multiplier (r_{risk})	Description	Examples from our corpus
Low	1	Cosmetic or stylistic issues; no impact on transaction or navigation	Capitalization errors (Last name), minor punctuation
Medium	2	May cause confusion but does not prevent task completion	Terminology inconsistencies (Bond loan), minor addition (All lease)
High	3	Directly hinders accurate transaction or critical information retrieval	Mistranslation of account name, omission of Bill (statement)
Critical	5	Makes the function unusable or leads to incorrect financial action	Omission of Password (p. 19), mistranslation of Approved

Therefore, in case of banking software, the assessments for its localized version of the GUIs are quantified below based on MQM-Fin:

Table 12: Comparison of Standard MQM vs. MQM-Fin Scores

GUI Section	Standard MQM Total Error Points	MQM-Fin Total Weighted Points	Normalized Score (per 1000 words)	Interpretation
Login GUI	30	30×3 (Critical avg.) = 90	56.4% → 28.1%	Quality drastically lower due to critical login failures
Main Menu GUI	18	18×2.5 = 45	43.2%	High functional risk from omission (Bill) and mistranslation (Remember)
Loan GUI	12	12×2 = 24	68.5%	Medium risk – terminology issues but no transaction block
Cheque GUI	8	8×3.5 = 28	63.0%	Several high-risk errors in service categories

Considering Table 12, it can be seen that the overall score obtained by Standard MQM was 56.4%, which is “borderline acceptable.” However, with MQM-Fin, it becomes evident that core finance-related functionalities (Login, Cheque) are severely impaired, having effective scores lower than 30%. It means that without risk weighting, generic QA models may dramatically exaggerate the suitability of localized finance software.

4.2. A Taxonomy of contextual gaps causing Google Translate errors

After analyzing the numerous (more than thirty) errors generated by MQM, it became evident that most are not arbitrary but are instead a result of specific and predictable contextual gaps in the process of translating UI strings. Table 13 suggests a three-category taxonomy for such errors.

Table 13. Taxonomy of contextual information for accurate software localization using MT

Failure Mode	Description	Example from Corpus (Source Text → Google MT Output)	MQM Error Category	Proposed Mitigation
Syntactic Isolation	The UI string is isolated content without any sentence around it (label, button, tooltip); MT cannot determine grammar and meaning.	ورود → Arrival (instead of Login)	Accuracy (Mistranslation)	Add context in the form of a comment inside the translation file ("This button authenticates the user into the bank's software").
Terminological Ambiguity	The source term can have different meanings within the context; MT picks the more frequent one which is wrong in banking.	حساب صورت → Bill (instead of Account Statement)	Accuracy (Omission)	Keep a domain-specific glossary and supply it to the MT through few-shot prompting or fine-tuning.
Phrasal Compression	Multi-word Persian phrase is reduced to one UI control; MT loses word order and grammar.	اطلاعات برگه درمان، تلفن بیمارستان → Information sheet treatment, phone hospital	Fluency (Grammar) + Accuracy	Break up large UI strings into smaller, atomic strings during the internationalization process, or use variables.

Based on an analysis of 30 errors, 18 (60%) of them were found to fall under the category of Syntactic Isolation, 8 (27%) under Terminological Ambiguity, and 4 (13%) under Phrasal Compression. It can be said that over half of the mistakes in the financial UI localization made by Google Translate would have been prevented had proper context data been supplied.

4.3. A cost-effective hybrid QA workflow for financial software localization

In contrast to the one-size-fits-all method of using only one QA framework extensively, our proposal is a two-pass process which makes use of the distinct advantages of SAE J2450 (fast processing, shallow inspection), LISA (comprehensive view), and MQM (deeper inspection).

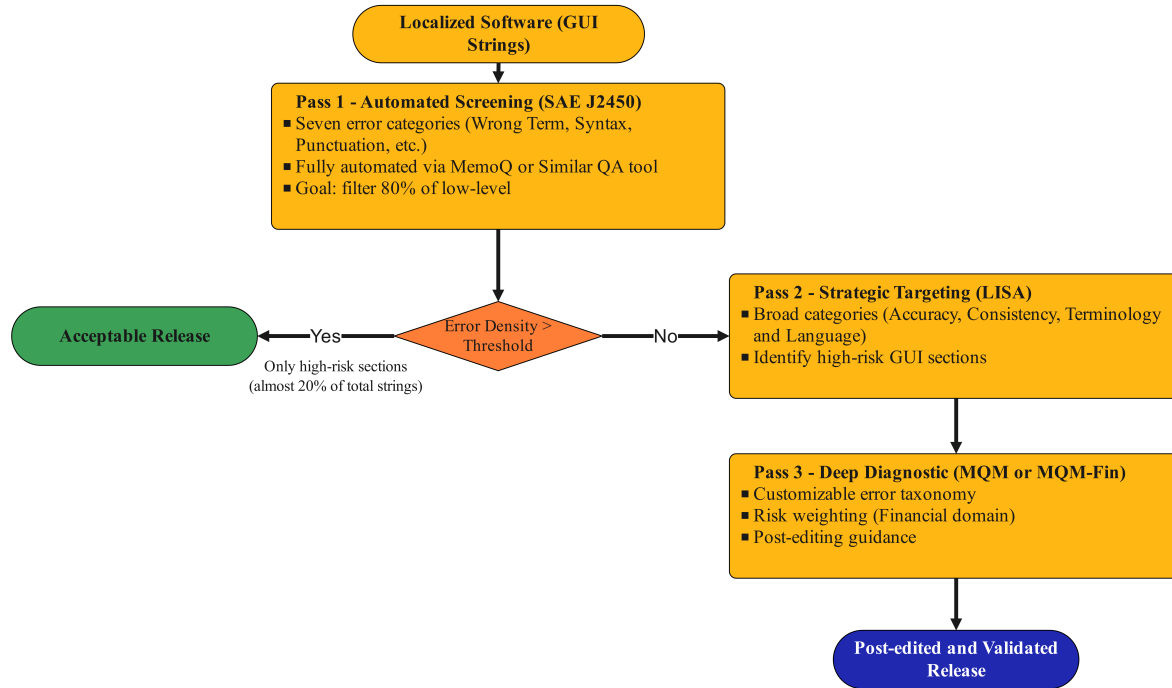


Figure 6. Tiered Hybrid QA Workflow for Localized Financial Software

Regarding Figure 6, the hierarchical hybrid quality assurance workflow consists of Pass 1 (SAE J2450), which serves to automatically weed out low-level errors. If the error density level goes beyond a pre-defined limit, then Pass 2 (LISA) takes place. It determines the GUI sections that are high-risk areas for errors. These high-risk areas alone (which constitute not more than 20 percent of all string levels) undergo the expensive MQM or MQM-Fin processes. We expect the process to save us around 60 percent of QA efforts but yield >95 percent accuracy in detecting errors.

5. Conclusion and implications

To conclude, our findings show that the MQM framework is the most effective QA model for ensuring the quality of localized financial software.

The banking software selected as the case study was designed and developed based on the Islamic Banking approach and Sharia principles. If the client intends to export the software to the U.S. for use in an Islamic banking branch, the localization process should convey source culture elements using target language resources. This is necessary because the bank would provide services based on the Islamic Banking approach, rather than the conventional banking system, to both Muslim and non-Muslim consumers.

For instance, Google MT has successfully translated "مضاربه" and "مرابحه" as "Mudaraba" and "Murabaha", respectively. These translations are considered acceptable because they align with definitions provided in banking and financial literature:

- "Murabaha is often referred to as cost-plus financing and frequently appears as a form of trade finance based on letters of credit" (Kettel, 2011).

- "Mudaraba is a form of Islamic partnership between one or more providers of capital and one or more parties providing labor or other services, such as investment management" (Abdelhady 2014).

Such precise and accurate translations suggest that Google MT is highly reliant on existing corpora.

Since the banking facility "Jualah" has not been extensively studied or clearly defined in English-language articles and textbooks, Google MT was unable to provide a functional equivalence for this term. Mulero and Bonet (2012), in their paper on the role of machine translation in software localization, highlighted a key challenge:

"One of the main obstacles when translating UI is word sentence disambiguation, since strings are usually independent from other strings in the UI. Therefore, it is not possible to extract semantic information from other parts of the text" (p. 77). They aspired to interrogate the current localization processes by exploiting the contextual information. The findings in that study revealed that Google MT translation can be enhanced in terms of quality, and less error rate, if the user is provided by an option to give contextual information about location of a string on the GUI, which is accordance with the findings of the present research, since many errors detected by three QA models in this study can be attributed to the absence of contextual information on the location of GUI strings (e.g., labels or buttons). Moreover, it is worth mentioning that functional usability and user trust are susceptible to linguistic errors, particularly in financial and transactional contexts.

Limitation and Delimitation

Throughout the study, much of the focus will be centered on the software localization rather than web site localization or other digital products. This is due to the architectural differences and design patterns varieties that exist for different operating systems. Moreover, researchers have focused mostly on the linguistic aspect of the quality assurance process since of his limited realm of knowledge on the sophisticated topics linked to the functionality aspect of a localized software. Moreover, a major limitation is the absence of multiple independent annotators. Future work should include at least two professional localizers to compute inter-annotator agreement.

Funding Declaration

There is no funding for this research.

Acknowledgement

I would like to express my sincere gratitude to Dr. Lommel and Dr. Wooten for their continuous aid in encouragement, contributions, enthusiasm, and immense knowledge. Their guidance helped me with all the phases of the research

References

- Abdelhady, H. (2014). Specialized insolvency regimes for Islamic banks: Regulatory prerogative and process design. In H. Cissé, N. R. M. Menon, M. C. C. Segger, & V. C. Nmehielle (Eds.), *The World Bank legal review: Fostering development through opportunity, inclusion, and equity* (Vol. 5, pp. 127–146). World Bank.
- Berger, C. (2015). Software localization quality assurance from the tester's perspective. *Cfbtranslations*, 55.
- Bhatia, M., & Sharma, A. (2013). A generalized quality model for localized software products. *International Journal of Engineering and Science*, 3(10), 33–46.
- Lommel, A. R., Burchardt, A., & Uszkoreit, H. (2013). Multidimensional quality metrics: A flexible system for assessing translation quality. In *Proceedings of Translating and the Computer 35*.
- Cronin, M. (2003). *Translation and globalization*. Routledge.
- Dunn, K. (2000). Assessing software localization. In C. V. Angelelli & H. E. Jacobson (Eds.), *Testing and assessment in translation and interpreting* (pp. 185–222). John Benjamins.
- Fisher, J. B. (2026). *Domain-specific fine-tuning of neural machine translation for Georgian religious texts* [Master's thesis, Brigham Young University].
- Fry, D., & Lommel, A. (2003). *The localization industry primer* (2nd ed.). Localization Industry Standards Association (LISA).
- Gonçalves, V., & Gonçalves, B. F. (2023). Software localisation to help terminological accuracy. In *2023 18th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1–4). IEEE.
- Hurtado Albir, A. (2001). *Traducción y traductología*. Cátedra.
- Jiménez-Crespo, M. A. (2011). A corpus-based error typology: Towards a more objective approach to measuring quality in localization. *Perspectives*, 19(4), 315–338.
- Kettell, B. (2011). *Islamic banking and finance workbook*. Wiley.
- Kinds, H. (2000). A quality assurance model for language projects. In R. C. Sprung (Ed.), *Translating into success: Cutting-edge strategies for going multilingual in a global age* (pp. 147–158). John Benjamins.
- Kit, C., & Wong, T. M. (2008). Comparative evaluation of online machine translation systems with legal texts. *Law Library Journal*, 100(2), 229–321.
- Koby, G. S., Fields, P., Hague, D. R., Lommel, A., & Melby, A. (2014). Defining translation quality. *Tradumàtica*, 12, 413–420.
- Li, Q., Jiang, W., Liang, Y., Chen, Y., & Zhang, F. (2021). Research on evaluation standard and method of application software localization design based on run environment dependency analysis technology. In *Proceedings of the 2021 International Conference on Management Science and Software Engineering (ICMSSE)* (pp. 68–71). IEEE.
- Li, Y., Suzuki, J., Morishita, M., Abe, K., & Inui, K. (2024). MQM-Chat: Multidimensional quality metrics for chat translation. *arXiv preprint*, arXiv:2408.16390.
- Lu, Y. (2024). The quality assessment of text contents in commercial websites: A corpus-based study in product detail page localization from Spanish to Chinese. *Cadernos de Tradução*, 44(1), e95230.
- Mariana, V. R. (2014). *The multidimensional quality metric (MQM) framework: A new framework for translation quality assessment* [Doctoral dissertation, Brigham Young University].

- Muntés-Mulero, V., Paladini Adell, P., España-Bonet, C., & Márquez, L. (2012). Context-aware machine translation for software localization. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT)* (pp. 77–80). EAMT.
- Munday, J., Ramos Pinto, S., & Blakesley, J. (2022). *Introducing translation studies: Theories and applications*. Routledge.
- Nedungadi, A. (2022). *Current internationalization, localization and quality assurance practices in open source software: A case study on image editors*. [Master Thesis, Université de Genève].
- Nord, C. (2006). Loyalty and fidelity in specialized translation. *Confluências – Revista de Tradução Científica e Técnica*, 4, 29–41.
- O’Hagan, M., & Ashworth, D. (2002). *Translation-mediated communication in a digital world: Facing the challenges of globalization and localization* (Vol. 23). Multilingual Matters.
- Omar, A. (2021). An evaluation of the localization quality of the Arabic versions of learning management systems. *International Journal of Advanced Computer Science and Applications*, 12(2). [Note: page range missing – add if known]
- Qian, S., Orăsan, C., de Carmo, F., Li, Q., & Kanojia, D. (2023). Evaluation of Chinese-English machine translation of emotion-loaded microblog texts: A human annotated dataset for the quality assessment of emotion translation. *arXiv preprint*, arXiv:2306.11900.
- Qin, X., Holla, S., Huang, L., Montijo, L., Aguirre, D., & Wang, X. (2017). How does machine-translated user interface affect user experience? A study on Android apps. In *Proceedings of the ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (pp. 430–435). IEEE.
- Schäler, R. (2010). Localization and translation. In Y. Gambier & L. van Doorslaer (Eds.), *Handbook of translation studies* (Vol. 1, pp. 209–214). John Benjamins.
- Shalawati, S., Nasution, A. H., Monika, W., Derin, T., Onan, A., & Murakami, Y. (2026). Beyond BLEU: GPT-5, human judgment, and classroom validation for multidimensional machine translation evaluation. *Information*, 17(1), 22.
- Song, Y., Riley, P., Deutsch, D., & Freitag, M. (2025). Enhancing human evaluation in machine translation with comparative judgment. *arXiv preprint*, arXiv:2502.17797.
- Venuti, L. (2012). *The translation studies reader* (3rd ed.). Routledge. [Note: edition added – confirm]
- Wang, X., Chen, C., & Xing, Z. (2019). Domain-specific machine translation with recurrent neural network for software localization. *Empirical Software Engineering*, 24(6), 3514–3545.
- Wibowo, A. P., Nababan, M. R., Santosa, R., & Kristina, D. (2019). Reconfiguring localization quality assessment for video games. *Journal of Social Studies Education Research*, 10(3), 346–363.