Translation Quality Assessment
in Computer-Assisted Translation Tools

DANIEL MARÍN BUJ – STUDENT ID: 16069218

Dissertation for the MSc in Multilingual Computing and Localisation

Department of Computer Science and Information Systems

Supervisor:
TABEA DE WILLE

Submitted to the University of Limerick, September 2017
Abstract

The increasing interest in measuring translation quality according to objective methods and standard metrics has led some translation software companies to start integrating assessment-enabling features into their products. Despite the adequacy of computer-assisted translation tools for providing support to human assessment tasks and tool makers’ claims about their success in its implementation, the actual capacities of the tools for quality assessment still need to be demonstrated. Formal translation quality assessment is a complex activity that requires the creation of issue typologies, severity-based penalties, quality score calculations and quality report generation, among other elements. Also, since one of the assessment goals is to improve the quality, it is crucial that metrics, annotations and reports can be consumed by all stakeholders without the need of additional transformation. In this sense, interoperability aspects and standards must also be respected for the assessment effort to be useful and efficient. In this study, a set of five translation tools integrating assessment features were examined according to two main aspects: assessment feature implementation and assessment data interoperability. To carry out the evaluation, a list of sixteen items were defined under those two aspects, from the capacity of customising a quality model to the exchangeability of the assessment data produced. These items were primarily inspired in one of the most comprehensive quality frameworks to date: the Multidimensional Quality Metrics. The results confirm that computer-assisted translation environments do support the most basic characteristics of translation quality assessment, such as applying custom quality metrics or generating scores and reports. However, there is still room for improvement in many other aspects and, in particular, those related to the exchangeability of the assessment data. Finally, if computer-assisted translation tools aim to be the perfect instrument for the assessment of translation quality, then existing quality frameworks must be fully understood, followed and enabled by tool makers, with special emphasis on the normative aspects of the models and the standard formats used to store and exchange quality assessment metrics and results.
Thanks

I want to thank my supervisor, Tabea De Wille, for her guidance throughout the programme and during the writing of this dissertation. Many thanks also to my partner Lorena Valero Romeo for her support and encouragement.
Declaration

I hereby declare that this document is the result of my work and that I have not submitted it previously to any institution.
# Table of Contents

Abstract ................................................................................................................. 3  
Thanks ................................................................................................................... 4  
Declaration ............................................................................................................ 5  
Table of Contents ................................................................................................. 6  
List of Tables ........................................................................................................ 8  
List of Figures ....................................................................................................... 9  
List of Abbreviations ............................................................................................ 12  
Chapter 1: Introduction ....................................................................................... 13  
  1.1 Research scope and background ................................................................... 13  
  1.2 Research question ....................................................................................... 16  
Chapter 2: Literature Review ............................................................................. 18  
  2.1 Introduction .................................................................................................. 18  
  2.2 TQA in the translation industry .................................................................... 19  
  2.3 TQA and CAT tools .................................................................................... 23  
  2.4 TQA and XLIFF ......................................................................................... 25  
  2.5 CAT tools and XLIFF ................................................................................ 27  
Chapter 3: Methodology ..................................................................................... 30  
  3.1 Introduction .................................................................................................. 30  
  3.2 Setup ............................................................................................................ 31  
    3.2.1 Quality metric ...................................................................................... 31  
    3.2.2 Sample corpus ..................................................................................... 34  
    3.2.3 Analysis items ..................................................................................... 35  
    3.2.4 Translation tools ................................................................................ 37  
    3.2.5 Testing environment .......................................................................... 40  
    3.2.6 Analysis of test results ..................................................................... 40  
  3.3 Conclusion ................................................................................................... 41
List of Tables

Table 1. Overview of tools according to previous studies.............................. 29
Table 2. Analysis items................................................................. 37
Table 3. Translation tools with TQA features to be examined..................... 39
Table 4. Built-in quality models summary........................................ 48
Table 5. Customising a metric summary............................................. 53
Table 6. Issue hierarchy summary.................................................... 56
Table 7. Severity levels summary..................................................... 60
Table 8. Issue weighting summary.................................................... 63
Table 9. Issue flagging summary....................................................... 67
Table 10. Markup scope summary...................................................... 70
Table 11. Overlapping spans summary............................................... 73
Table 12. Split issues summary......................................................... 75
Table 13. Report generation summary................................................ 81
Table 14. Score calculation summary................................................ 84
Table 15. Intermediate format summary.......................................... 86
Table 16. Markup summary.............................................................. 89
Table 17. Report compatibility summary......................................... 92
Table 18. Import/export summary.................................................. 95
Table 19. MQM compliance summary............................................. 97
Table 20. TQA in CAT tools summary............................................. 98
Table 21. TQA implementation and interoperability in CAT tools............ 99
Table 22. TQA in CAT tools according to application type.................... 100
Table 23. TQA in CAT tools according to business model..................... 101
**List of Figures**

Figure 1. Diagram of main research areas.................................................................18
Figure 2. Quality categories hierarchy.................................................................33
Figure 3. Quality categories hierarchy (mind map).............................................33
Figure 4. MQM integration in Fluency Now.......................................................43
Figure 5. LQA as a project component in memoQ.................................44
Figure 6. Metric changes force documents to be reimported in MemoQ........44
Figure 7. TQA module in SDL Trados Studio.................................................44
Figure 8. MQM pane in translate5..............................................................45
Figure 9. Global LQA settings in XTM Cloud...............................................45
Figure 10. Loading default MQM definition in Fluency Now......................46
Figure 11. Built-in quality models in MemoQ..................................................47
Figure 12. Custom quality metric in Fluency Now...........................................50
Figure 13. Customising a quality model in MemoQ........................................51
Figure 14. Definition of an error type in XTM with a read-only MQM name....53
Figure 15. Issue hierarchy in Fluency Now.......................................................54
Figure 16. Issue hierarchy in SDL Trados Studio 2017...............................55
Figure 17. Issue hierarchy in XTM 10.4..........................................................56
Figure 18. Issue severities in Fluency Now.......................................................58
Figure 19. Severities and penalty values in MemoQ.......................................58
Figure 20. Severities and penalty value in SDL Trados Studio......................59
Figure 21. Severity levels in XTM Cloud..........................................................59
Figure 22. Flagging an issue with no weight set in Fluency Now..................61
Figure 23. Issue weighting in SDL Trados Studio..........................................62
Figure 24. Issue weighting in XTM Cloud.........................................................63
Figure 25. MQM pane in Fluency Now..............................................................64
Figure 26. Flagging an issue in MemoQ. ................................................................. 65
Figure 27. Categorising an issue in MemoQ. .......................................................... 65
Figure 28. Flagging an issue in SDL Trados 2017. .................................................... 66
Figure 29. Categorising an issue in SDL Trados Studio .......................................... 66
Figure 30. Flagging an issue in translate5 ............................................................... 66
Figure 31. Categorising the issue in translate5 ....................................................... 67
Figure 32. Flagging an issue in XTM 10.4. ............................................................... 67
Figure 33. Flagging scope in MemoQ. .................................................................... 69
Figure 34. Overlapping issues in MemoQ. .............................................................. 72
Figure 35. MemoQ's View pane. ............................................................................ 72
Figure 36. Overlapping spans in translate5 ............................................................. 72
Figure 37. Marking an issue in distinct places in SDL Trados Studio. ................. 74
Figure 38. TQA pane showing a split issue in SDL Trados Studio ....................... 74
Figure 39. Split issues in SDL Trados Studio ......................................................... 74
Figure 40. TQA report in Fluency Now. ................................................................. 74
Figure 41. Project-based reports in MemoQ. ............................................................ 77
Figure 42. Quality report in MemoQ. .................................................................... 77
Figure 43. Error list in MemoQ. ............................................................................ 78
Figure 44. Generation of a TQA report in SDL Trados Studio ......................... 78
Figure 45. TQA report in SDL Trados Studio ......................................................... 79
Figure 46. Report's annotations in SDL Trados Studio ........................................... 79
Figure 47. MQM statistics in translate5 ................................................................. 80
Figure 48. XTM Cloud's quality report ................................................................. 81
Figure 49. MemoQ's Pass/Fail criteria ................................................................... 83
Figure 50. SDL Trados Studio's Pass/Fail criteria ............................................... 84
Figure 51. Overall quality score in SDL Trados Studio ........................................... 84
Figure 52. Exporting reports in MemoQ.......................................................... 90

Figure 53. MQM Core schema................................................................. 115
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>CAT</td>
<td>Computer-Assisted Translation</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma-Separated Values</td>
</tr>
<tr>
<td>DFKI</td>
<td>German Research Center for Artificial Intelligence</td>
</tr>
<tr>
<td>DQF</td>
<td>Dynamic Quality Framework</td>
</tr>
<tr>
<td>ITS</td>
<td>Internationalization Tag Set</td>
</tr>
<tr>
<td>LQA</td>
<td>Linguistic Quality Assurance</td>
</tr>
<tr>
<td>MQM</td>
<td>Multidimensional Quality Metrics</td>
</tr>
<tr>
<td>QA</td>
<td>Quality Assurance</td>
</tr>
<tr>
<td>QT21</td>
<td>Quality Translation 21</td>
</tr>
<tr>
<td>RTF</td>
<td>Rich Text Format</td>
</tr>
<tr>
<td>TAUS</td>
<td>Translation Automation User Society</td>
</tr>
<tr>
<td>TQA</td>
<td>Translation Quality Assessment</td>
</tr>
<tr>
<td>XLIFF</td>
<td>XML Localization Interchange File Format</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>XSLT</td>
<td>Extensible Stylesheet Language Transformations</td>
</tr>
</tbody>
</table>
Chapter 1: Introduction

Assessing translation quality may serve many purposes: screening freelance translators, providing feedback to language vendors, evaluating machine translation output or assessing student translations, among others. However, the main goal stays the same: "to ensure a specified level of quality is reached, maintained, and delivered to the client, buyer, user, reader, etc." (Doherty 2017). To achieve that goal, the methods are varied, but when it comes to technologies, two approaches stay quite prominent: automatic quality checkers and assessment scorecards (Lommel 2017). With respect to the software, Doherty et al. (2013) already revealed the popularity of built-in CAT tools for such purposes over other means. The advantages seem evident: the source and target text view, the integration of linguistic resources, the automated quality checks, and so on. As a matter of fact, this popularity has grown over time as translation software companies have been including more and more features—especially quality-oriented—to their tools, including specific functionalities to perform translation quality assessment in a normalised way, eventually bringing formal quality models into the picture of translation technologies.

1.1 Research scope and background

The perspective adopted in this study is primarily technical. Without ignoring that translation quality is an area where linguistic approaches have seen multiple and valuable contributions (Colina 2008; Williams 2009; Daems et al. 2013), the focus of this paper is placed on the translation technology scene. In this sense, the linguistic aspects, although considered, are only referred to explain and contextualise the characteristics examined in the software implementation of quality assessment features. That is to say, this study is not discussing the validity of the different linguistic approaches to translation quality, but only using them when necessary to illustrate the aspects of its implementation in translation tools.

Having said that, we cannot avoid admitting that quality assessment is often a fuzzy concept; and it is certainly difficult to find a consensus on what it involves in theoretical and practical terms, especially because of many years of lack of a normalised framework (Colina 2008; Hague et al. 2011). Therefore, it seems relevant to declare already that this paper deals with the human task of evaluating
and measuring the quality of translations, and does not refer directly to other aspects of quality assurance, such as automatic quality checks or other types of control and business processes.

Also, the ultimate object of this study is CAT tools, and, in consequence, another clarification may be necessary in relation to the ‘CAT’ label. The field of translation tools is vast, and the acronym ‘CAT’ is somehow vague, as Zetzsche (2017) recalls: “CAT […] is a great term for describing the numerous families of software tools that translators use for their work […] Unfortunately, we often use "CAT" as a synonym for so-called "translation memory tools," when the latter is really only a subcategory of the former”. This clarification is also valid here: this study relates to translation tools in a broad sense, including translation-specific elements, but also other related aspects such as TQA.

Concerning TQA support in CAT tools, compared to QA automated checks, which started to show a generalised penetration in translation workflows and systems some time ago (Makoushina 2007), TQA features are still seen as a novelty. In fact, many translation environments do not have yet formal mechanisms to perform quality assessment based on predefined metrics. However, the tendency is to integrate instruments for the measurement of the translation quality within the translation software, regardless the origin of the translation or the purpose of the assessment. A proof of this trend is the number of marketing and blog articles certain tool developers have dedicated to praise their new implementations in this regard. To give only some examples:

Introducing the concept of Linguistic Quality Assurance, measuring the quality of translations becomes easy. Very easy. […] Companies can measure the quality of their translations and translation vendors, while translation vendors can give feedback on the translation errors their reviewers marked.

Kilgray (2013)¹

MQM is here! What is MQM? MQM, or Multidimensional Quality Metrics, is a framework for defining metrics and scorecards used to assess the quality of translated texts. Users will notice MQM adopts a “functionalist” approach during translations. This is especially popular when the desired effect is to encourage an appropriate communicative purpose from text.

Western Standard (2015)

Through the development of the TQA, the principles and methodology we have introduced gives you the ability to measure objectively the quality and consistency of everyone in the translation supply chain, to ensure client requirements are achieved.

SDL (2016)

The launch of SDL Trados Studio 2015 saw the addition of the Translation Quality Assessment (TQA) tool which gives the user a simple way to perform quality evaluation either by using existing industry models or by creating their own framework within the tool.

SDL (2016)

LSPs can now access LQA assessments of their work which enables project managers and vendors to view and discuss quality trends objectively, with the aim of continuous improvement.

XTM (2017)

Smartling’s newest reporting functionality, Quality Evaluation with DQF is a complete measure of translation quality via comprehensive evaluation, tracking, and benchmarking data.

Smartling (2017)

Finally, we cannot forget that quality is a pervasive aspect of all stages of the translation cycle. Customers, vendors, and translators are all concerned with

_________________________

5  https://xtm-intl.com/blog/xtm-cloud-v10-3-live-better-collaboration-quality-productivity.
respect to quality, which requires back and forth communication among all actors. A proof of that important TQA angle are the expressions used in the above-quoted messages: “give feedback”, “encourage an appropriate communicative purpose”, “everyone in the translation supply chain”, “project managers and vendors to view and discuss quality trends”. At the same time, quality assessment tasks are linked to data in the form of metrics, quantifiable goals, annotations, or scores, to mention only the most evident items that need storage, transmission and visualisation across the chain. Therefore, not only adequate TQA implementations are important in terms of functionality, but also the fact that TQA-related data needs to be appropriately shared and exchanged among all stakeholders. A tool that integrates TQA functionality perfectly, but does not support or generate data in an open or standard format, might not be as useful as marketers announce in the end.

Therefore, following the recent but strong claims made by translation software companies with respect to TQA, CAT direct or indirect users could already expect that translation tools already offer an acceptable support to quality assessment activities. This support is precisely the area of this research, particularly TQA implementations as well as interoperability aspects, regardless the origin of the translation or the assessment context.

1.2 Research question

As we have seen, translation quality assessment is generating more a more interest in translation technology companies and translation-assisted tool makers. However, we have also seen that TQA integration is still in its infancy for most CAT tools, as only last releases of some of them are showing a serious vocation to include actual instruments for the human assessment effort. Considering this, our research question is the following:

*Do computer-assisted translation tools already provide a real support for the execution of translation quality assessment tasks in terms of implementation and interoperability?*

To answer this question, it seems important to delimit the extent of what we call *TQA tasks*. By understanding what hands-on TQA involves, we should be in a better position to examine the extent of the coverage translation tools offer
in this area. But, as it happens when the parent topic is wide and somehow controversial, exceptional care must be put on what is the essence of the TQA. That is, what are the necessary characteristics that TQA must have to be fully considered TQA in a CAT tool. How does it compile, compute and report measures? What are the means in place as to facilitate the exchange of the data? Etc. While the answers to these questions are still subject to evolution and development, the following chapter intends to present the foundations for TQA support in CAT tools so that we can face the analysis and the discussion under the best conditions.
Chapter 2: Literature Review

2.1 Introduction

This work involves three fundamental areas: TQA, CAT technologies and system interoperability. Although each one of these areas have been studied on their own, few academic studies seem to exist in relation to the execution of TQA tasks in CAT tools and the related data exchangeability as of today. As a result, this chapter presents the most revelant research carried out in the intersection of at least two of these fields, e.g. CAT tools and quality assessment, or CAT tools and interoperability. The priority and central focus stays nevertheless the authoritave references that deal directly with the intersection of all of them, as represented in Figure 1.

![Figure 1. Diagram of main research areas.](image)

One of the possible reasons academic literature is not abundant in the field of TQA and translation technologies is that there have not been real software support for human assessment until very recently. As we have seen in Chapter 1, the few tools that include TQA enabling features have integrated such features only in the last four years and that only in their most recent releases. The vast majority of CAT tools still do not have any explicit characteristic related to formal human assessment (see 3.2.4 for a non-exhaustive list of them). However, TQA is gaining importance thanks, in particular, to the development and implementation of machine translation solutions and the need of a standardised evaluation approach (Chunyu and Wong Tak-ming 2015), which explains partly the growing interest for integrating quality assessment methods in the translation systems and the corresponding research attention.

- 18 -
Given the complexity human assessment involves, some authors have observed for some time now “the need for objectivity (through precise measurement and quantification)” over other aspects (Williams 2009, p.7). As we examine further below, this need for objectivity, measurement and quantification pushed the industry to develop quality frameworks that included error typologies together with scoring mechanisms (Lommel et al. 2015). That is, as Martínez Mateo emphasises, “fixing a number of parameters or criteria as a yardstick for comparing real versus ideal performance” (2014, p.74). Ultimately, what these ideas seem to have in common is the underlying search for quality quantification based on formal methods.

Therefore, quality assessment’s main purpose seems to involve measuring the quality and producing a score based on known variables and inputs, such as the number of the issues compared to the volume of the translation, or the relevance certain issues have over the others, hence the opportunity CAT tools represent for such exercise (Yang et al. 2017). As a result, as Dohery (2017) points out, “more advanced and tool-based TQA methods have become more widespread, especially in the localisation and IT industries.”, which confirms the trend towards a logical integration of the assessment in the translation software itself.

However, although the evolution of the translation technologies is bringing a new landscape for quality assessment, the question of its maturity degree remains open. The impression received from studies like the one by Harris et al. (2016) is that translation technologies seem generally incomplete, if not precarious, when it comes to translation evaluation and interoperability. A certain agreement exists, nonetheless, on the “positive impact” (ibid., p. 54) that the development of environments for translation quality evaluation can have for the language industry, a judgement that seems justified given the TQA goals, the adequacy of the translation systems for dealing with such complexity, and the need to support the interoperability among the different systems.

2.2 TQA in the translation industry

From the practical side, the idea of TQA as a process enabled or not by specific tools is often subject to confusion. A sense of this confusion can be
perceived in the industry as well as in scientific studies that use labels like ‘quality control’ (O’Brien 2012), ‘quality assurance’, (Stejskal 2009; Martínez Mateo 2014; Filip 2017), ‘quality assessment’ (Doherty 2017) or ‘quality evaluation’ (Colina 2008; García 2014; Chan 2015; Harris et al. 2016) for the same or for different yet similar kinds of processes, such as automated quality checks, machine translation evaluation or quality assessment (Lommel et al. 2015, p.3; Korkas 2017, p.55). Although quality remains the core concept, the boundaries of each area in relation to translation technologies are not always clear or conclusive, and there is, in fact, overlapping among them, as reported by the academia and by specialised magazines (Chiocchetti et al. 2017, p.164; Korkas 2017, p.55).

According to Martínez Mateo (2014, p.75), “the process of evaluating translation quality is widely known as Translation Quality Assessment (TQA)”, observing that “many proposals for TQA have already been laid on the table, but none of them has proved to be a definite solution” (ibid., p.75). Despite the lack of agreement on what TQA involves (Williams 2009), there seems to be a certain consensus on that “quality assessment is the measurement of the extent to which a product complies with quality specifications”, as one of the standard quality models referred here advises (Lommel et al. 2015). Even if this definition stays relatively generic, it is significant because it stresses an industry-oriented approach, which is somehow different from those found in translation studies or in machine translation research, generally more theoretical and rigid (Doherty 2017). According to Doherty, TQA in industry “is typically dominated by human evaluation and semi-automated methods using deductive error-based typologies”, an activity in which “more advanced and tool-based TQA methods have become more widespread, especially in the localization and IT industries” (ibid., p.6).

As we have seen, the idea of the measurement in the translation industry comes usually with the concept of error-based typologies in the form of pre-defined metrics as an instrument for assigning marks and numeric values that can be then computed to get a quality score:

Translation quality assessment (TQA) is the process of evaluating the overall quality of a completed translation by using a model with predetermined values that can be assigned to a number of parameters used for scoring purposes.
In the end, the underlying idea is that TQA in the industry represents the process of measuring the degree of compliance of a translation according to a set of requirements formalised as a classification of issue types, severity levels and scores, all these elements being described within, or conceived as a framework (Doherty and Gaspari 2017). Indeed, quality frameworks in the translation industry are nothing new: the development of a number of them started already in the nineties: LISA QA, SAE J2450 and TAUS (Martínez Mateo 2014). However their usefulness, their implementation has been irregular, as reflected by Doherty et al. (2013), who also reveals that, after two decades of existence, the industry’s preference is still for internal and tool-specific models. One of the logical reasons argued for this irregular adoption is the “one-size-fits-all” approach of those first models (Uszkoreit and Lommel 2013).

The failure of those initiatives in connection with the variety of TQA purposes and the dynamic nature of the translation industry, brought to the scene two innovative approaches: the TAUS Dynamic Quality Framework and the QTLaunchPad Multidimensional Quality Metrics (García 2014, p.432). The first one, the TAUS Dynamic Quality Framework (DQF), tried to standardise the quality assessment in industry:

Translation quality evaluation is problematic. Last-century metrics like the LISA QA model and SAE J2450 still prescribe today’s quality processes which are static, costly and non-transparent. [...] In order to overcome these problems, TAUS supports buyers and providers of translation with the Dynamic Quality Framework (DQF): a comprehensive set of tools, best practices, metrics, reports and data to help the industry set benchmarking standards.

TAUS7

The second approach was incarnated and driven by the QTLaunchPad, a European Commission-funded collaborative research initiative, which developed

______________________________

7 https://www.taus.net/evaluate/dqf-background.
a new framework: the Multidimensional Quality Metrics. The lessons learnt in the past were also considered as in the TAUS DQF’s case, meaning “that evaluation metrics must be adaptable, with “dynamic” or “multidimensional” basically meaning that on size will not fill all cases; that they should be affordable, able to be shared across industries, suitable for benchmarking; and, most importantly, should be objective, in as much as it can” (García 2014, p.432).

In QT21’s words:

Translation quality assessment (QA) is an important task, but one that is often contentious and fraught with difficulty. Traditional methods were highly subjective and involved reviewers reading translated texts and marking “errors”, but reviewers often disagreed on their assessments. In response, many organizations developed formalized metrics for assigning errors to different types (e.g., terminology, spelling, mistranslations), counting them, and determining how serious they were. In the 1990s, these efforts led to the creation of a widely used specifications such as SAE J2450 and the LISA QA Model. Unfortunately, these models have not been updated over time and they are presented as “one-size-fits-all” models that do not reflect the needs of a rapidly diversifying translation industry.

QT21 (2014)

By 2015, TAUS and DFKI announced that they had completed the harmonisation of their respective translation quality metrics, DQF error typology and the Multidimensional Quality Metrics (MQM), creating a “an industry-wide quality standard”10, another step taken towards the set-up of a unique framework for quality assessment, an endeavour that had been qualified previously as “illusory” (Martínez Mateo 2014, p.75). As presented by Burchardt et al. (2016), DQF error typology is now a subset of MQM and can be implemented in MQM tools (translate5, XTM, etc.). As seen by Harris et al. (2016, p.52), “this methodology has


received positive feedback from a number of research and industry users”, whose DQF harmonisation can help “to promote industry-wide uptake and push consolidation in the area of quality evaluation”. The reaction displayed by translation software companies seem to be also in the same direction according to the claims made (cf. 1.1), tentatively integrating the new framework ideas into their translation environments.

2.3 TQA and CAT tools

Before quality metrics started to be integrated in CAT tools, authors like García (2014, p.434) already pointed out that such integration was going to be “inevitable” in the field of machine translation. In fact, this consideration seems to have become real in the last years for both machine and human translation. Indeed, both industry and scientific research have praised for the benefits of quality assessment integration in CAT tools (García 2014, p.435; Harris et al. 2016). A reason may be that this integration seems to provide actual tools that can be used where there is a necessity of finding clear representation of translation quality (Yang et al. 2017). In a clearer manner, Burchardt, Harris, et al. (2016), argues that “any serious attempt to improve translation quality must include feedback by human experts such as translators and linguists”, for which it is necessary to “make sure that the human analysis and annotations process is optimally supported by tools” (ibid., p.36). Although the latter argument is focused on machine translation quality, we could assume it can be extended, even with better justification, to human translation. In favour of this idea, we find authors supporting a similar position in other areas. For example, a recent research example is given by Yang et al. (2017, p.2), who explicitly promote “CAT environments for more efficient TQA” in the context of translator training. In the same direction it has been argued that computer-assisted translation tools do play “an important role” in the presentation of the texts subject to assessment (Doherty 2017, pp.11–12).

This growing popularity of TQA in CAT tools may find a correlation to the somehow prior built-in quality assurance (QA) functionalities, the automatic quality checks that many users of the industry have seen integrated in their usual workflows, mainly thanks to their capabilities in detecting certain type of errors, such as missing translations, inconsistencies, spelling, length limitations or
terminology issues (Makoushina 2007). The high penetration of QA checks has indeed been confirmed by different studies (Makoushina 2007; Doherty et al. 2013), although is not the solution to measuring quality in translation. As it has been already argued from the industry, these methods have “significant drawbacks” (Korkas 2017, p.58). Even the more advanced quality evaluation algorithms applied to machine translation such as TER or BLEU are not sufficient in this respect, concluding that “there are no other efficient ways to evaluate translation quality in detail than to do this manually” (Harris et al. 2016, p.52).

Becoming aware of the limitations of the automatic methods and the relevance of obtaining objective quality measures is probably behind the fact that “tool-based TQA methods have become more widespread” (Doherty 2017, p. 6). This is at least the impression conveyed by Zetzsche in his Computer Primer for Translators (2017), who also mentions the integration aspect in the translation workflow:

Various translation environment tools have started to integrate these QA models into their linguistic review processes. Fluency and XTM, for instance, support MQM, memoQ, Memsource and MateCat support DQF [Dynamic Quality Framework], and Trados Studio 2015+ supports both. (Zetzsche 2017, 294)

In this sense, apart from the more direct relationship with the review processes mentioned by Zetzsche, the relationship between CAT tools and TQA is also apparent in other contexts, such as translators’ training or translator screening. For instance, Gouadec (2010), García (2014) or Yang et al. (2017) defend the usefulness of metrics in professional environments as an adequate way to perform TQA to evaluate student translation skills. And, on a similar note, Doherty et al. (2013), Shuttleworth (Chan 2015, p.684) and Doherty (2017) endorse TQA’s adequacy for evaluating professional translators when selecting resources for a job using translation technologies.

Nevertheless, a part of the industry initially warned about the weaknesses of state of the art in providing technical mechanisms for TQA. Arle Lommel (DFKI), in collaboration with Yves Savourel (ENLASO) and Phil Ritchie (VistaTEC), in a presentation for FEISGILTT 2012, hold the idea that the state of the art in localisation quality markup was “fragmented, tied to specific tools,
inconsistent [and] not integrated” (Lommel et al. 2012, p. 2). Although this statement was done at a time where most CAT tools had not yet started to implement standard quality metrics at a consumer level, it is something that must be considered for their evaluation. Yet more recent papers show a better perception of CAT tools for assessment tasks, especially regarding their user interface, although there is still the consideration that such interfaces are “less powerful” (Burchardt, Harris, et al. 2016, p.40) than TQA-focused tools.

Finally, a link can be established between revision and TQA in CAT tools. Robert (2008) quotes several authors to argue that “the practice of translation revision will probably become progressively more frequent” due to the publication in 2006 of the new standard EN 15038 for translation services. If we follow the European Committee for Standardisation quoted by Robert (ibid., p. 2), revision implies “to examine translation for its suitability for the agreed purpose, compare the source and the target texts and recommend corrective measures” (2006, p. 6). This means that, at least for the “examination part”, TQA could be also applied. Although sources quoted by Robert (ibid. p.19) are rather against the computerisation of the revision tasks because of ergonomic problems or wrongly designed tools in 2006, the technology landscape seems to have improved considerably these aspects thanks to the implementation of means that contribute to the execution of both TQA and revision tasks, as described by Zetzsche (2017, p.305).

### 2.4 TQA and XLIFF

The relationship between TQA and XLIFF has been studied indirectly by the analysis of the XLIFF possibilities to carry information related to quality, either via XLIFF elements or via the ITS namespace (Porto et al. 2013):

...the design of CAT tools is becoming an increasingly important area for investigation, especially in relation to how it supports integration with automated language technology components such as machine translation, terminology extraction and quality assessment.

(Porto et al. 2013)
Such integration is materialised by the use of data categories defined in ITS 2.0\footnote{See https://www.w3.org/TR/its20/#lqissue-typevalues.} and supported in XLIFF 2.x, which are also mapped to XLIFF 1.2 (Savourel 2014a). However, when examining a use case, Porto et al. (2013) also concludes that “support for these new metadata features was found to be much easier in a green field implementation than when trying to refactor an established CAT tool”. In other words, it was not possible to implement these data categories in one specific CAT tool, although nothing is said in relation to other CAT tools. At that stage, the study confirms the room for improvement in this area:

If the integration of new technology such as machine translation, text analytics and quality assessment, is to be integrated into the localization workflow using metadata annotations as advocated by the W3C in ITS 2.0, then such filters must accommodate more flexible means for accommodating new metadata.

Nevertheless, this relationship between quality assessment and XLIFF through CAT tools highlights the relevance of XLIFF as one of the factors that may determine the possibilities of TQA implementations. As much as translation tools integrate quality assessment features, the limit of such assessment—from a technical point view—seems to be defined from what is supported or not in the underlying XLIFF schema used to store the annotations. For instance, in XLIFF 1.2, there is “no way to have overlapping \texttt{<mrk>}[tags], which results in an “important obstacle to implement any type of annotation” (Savourel 2014b). Even if this issue is presumably fixed in XLIFF 2.0 by the use of standalone marks (\texttt{<sm/}> and \texttt{<em/>}), the problems linked to parsing overlapping spans in XLIFF 2.1 are not yet totally resolved (Filip 2017, p. 69). The technicalities put aside, Filip (ibid.) also illustrates the challenge behind the markup of overlapping fragments of texts if the processors are not capable of interpreting the inline tags used to mark the start and the end of fragments.

On the other hand, the problem of the quality markup in XLIFF is not only linked to the specifications of XLIFF itself, but also to the very use implementers
make out of it. As long as the information is encoded according to proprietary implementations, it may be challenging, even virtually impossible, to integrate quality information back in the translation process (Lommel et al. 2012, p. 6). In other words, if a given quality model is used in a particular tool, there is no guarantee that another tool, even if supporting the same XLIFF version and the same quality model, will interpret or parse the marks made by the former tool properly.

2.5 CAT tools and XLIFF

When it comes to assessing quality in translation tools, we must look at the way the assessment is enabled as a software feature, but also at how issues and annotations are stored and exchanged electronically together with the assessed texts. From its inception, the most adequate file format to fulfil the requirement of storing localisation data and metadata seems to be XLIFF:

The purpose of this vocabulary is to store localizable data and carry it from one step of the localization process to the other, while allowing interoperability between tools.

OASIS (2008)12

The industry appears to have understood the convenience of the XLIFF vocabulary as a container for translation and localisation data for some years now (Bly 2010; Filip and Morado Vázquez 2012; Morado Vázquez and Filip 2012). In consequence, nothing should prevent tool makers to consume quality metadata through the XLIFF as an XML-compliant standard vocabulary for translation content. This is the reason why the degree of support given by the tool to XLIFF is relevant for anything related to marking or annotating texts.

Finally, when considering XLIFF as the standard localisation file format, an important distinction must be made between XLIFF 1.2 and XLIFF 2.x, since each version deals differently with markup and annotation (OASIS 2014; Savourel 2014b). Unfortunately, despite of the fact that XLIFF 2.x provides much

12 http://xml.coverpages.org/xliff.html#overview.
stronger support for annotation (Lommel et al. 2012; Schnabel et al. 2015) as well as relevant features for the assessment, such as the change tracking module that allows storing revision information (OASIS 2014, p.98), the standard de facto continues to be version 1.2 when it comes to translation tools (Morado Vázquez and Filip 2014). Since XLIFF 2.x is not backwards compatible with 1.2, tools adopting version 2 might be, in principle, hindering the reuse of legacy bilingual files based on 1.2. (OASIS 2014, p.11).

Based on Filip and Morado Vázquez 2012 and Morado Vázquez and Filip 2014, the evidence of this resistance can be illustrated by the reports OASIS have issued on the support given to the XLIFF standard, where there is virtually no translation or localisation tool supporting version 2.0 as an intermediate format, as shown in Table 1. If we associate the tools reviewed in the studies by Morado Vázquez and Filip with the corresponding system types identified in the same studies (third column), it seems evident that standard CAT tools fall behind compared to more specialised tools. Therefore, integrated translation environments may not be optimally prepared to support TQA in the best conditions, unless demonstrated otherwise in the present analysis. In conclusion, the prospects regarding the adoption of the latest XLIFF specifications according to the existing studies leave much to be desired, an aspect that is examined here in relation to the CAT tools and interoperability.

---

13 Note that there are tools which do process XLIFF 2.0 as a working format, but they cannot be considered as proper CAT tools or integrated translation environments.
<table>
<thead>
<tr>
<th>Tool name</th>
<th>XLIFF version</th>
<th>Tool type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alchemy Catalyst 10</td>
<td>1.2</td>
<td>Localisation tool</td>
</tr>
<tr>
<td>Araya XLIFF Editor</td>
<td>1.2</td>
<td>Editor</td>
</tr>
<tr>
<td>Microsoft LBA 5.3/MLP 5.3</td>
<td>1.2</td>
<td>Localisation platform</td>
</tr>
<tr>
<td>MemoQ 5.0.64</td>
<td>1.2</td>
<td>Translation tool</td>
</tr>
<tr>
<td>MultiTrans Prism</td>
<td>1.2</td>
<td>Translation tool</td>
</tr>
<tr>
<td>Okapi Framework M16</td>
<td>1.2</td>
<td>Localisation support</td>
</tr>
<tr>
<td>OmegaT 2.3.0</td>
<td>1.2</td>
<td>Translation tool</td>
</tr>
<tr>
<td>OpenTM2 V.0.9.5</td>
<td>1.1</td>
<td>Translation management</td>
</tr>
<tr>
<td>SDL Trados Studio 2011 SP1</td>
<td>1.2</td>
<td>Translation tool</td>
</tr>
<tr>
<td>Solas v1</td>
<td>1.2</td>
<td>Localisation platform</td>
</tr>
<tr>
<td>Swordfish Translation Editor</td>
<td>1.2</td>
<td>Translation tool</td>
</tr>
<tr>
<td>Translation Factory 4.5.14</td>
<td>1.2</td>
<td>Translation tool</td>
</tr>
<tr>
<td>Translation Workspace 1.10</td>
<td>1.2</td>
<td>Translation tool</td>
</tr>
<tr>
<td>XTM 6.2</td>
<td>1.2</td>
<td>Translation tool</td>
</tr>
<tr>
<td>SOLAS</td>
<td>2.0</td>
<td>Localisation platform</td>
</tr>
<tr>
<td>CMS-L10N</td>
<td>2.0</td>
<td>Content management system</td>
</tr>
<tr>
<td>ITS2.0 XLIFF/MT Round-tripping Web Service</td>
<td>2.0</td>
<td>Machine translation processor</td>
</tr>
<tr>
<td>Okapi Tools</td>
<td>2.0</td>
<td>Localisation support</td>
</tr>
<tr>
<td>XMarker FragID Decorator v.0.2</td>
<td>2.0</td>
<td>XLIFF validator/processor</td>
</tr>
<tr>
<td>xliFFRoundTrip4X2</td>
<td>2.0</td>
<td>XLIFF extractor/merger</td>
</tr>
<tr>
<td>DITA-XLIFF-RT-4X2</td>
<td>2.0</td>
<td>DITA/XLIFF processor</td>
</tr>
</tbody>
</table>

*Table 1. Overview of tools according to previous studies.*
Chapter 3: Methodology

3.1 Introduction

In order to demonstrate whether—and to what extent—CAT tools offer support to TQA tasks in terms of interoperability, and from a software implementation perspective, several approaches have been considered.

In the first place, a user survey could have been adequate to show the level of adoption that TQA features have in the market, as well as to obtain a picture of the current user satisfaction degree in relation to this function. However, this approach has been discarded given that TQA as a software feature is still a quite recent one and its penetration in the market seems to lack enough consolidation for getting a representative sample. At the same time, TQA involves technical aspects that would have been difficult to assess with an open questionnaire. In addition, a survey addressed to the tool makers in the form of self-assessment would have been possible.

Another method considered has been the compilation and analysis of the relevant documentation, marketing material and information provided by the different software companies on this area. This approach has the advantage of simplifying the analysis since it does not necessarily require executing and verifying what the product documentation describes. However, this seems also a disadvantage, as little proof could have been given on the actual implementation. Also, as we have seen in Chapter 2, TQA is still a complex matter, and not all software companies have the same approach, and not even the same vocabulary to refer to TQA, which would have made the comparison among tools unreliable.

Finally, the method chosen has been testing the CAT-TQA offerings by simulating a set of actions in a simple use case and comparing the actual results with a tool-agnostic reference metric inspired by a standard model. This method actually allows going into implementation details, verifying what is present, absent or can be improved. It also has the advantage of defining the steps to reproduce and replicate the results. On the other hand, testing software is sometimes complex and requires certain familiarity with the systems tested, which involves risks, such as not using the software as it should be used or introducing variables
that alter the results (like the operating system, the local settings, etc.). Overall, this method may give an overview on the actual state of TQA in CAT tools, with its strengths and weaknesses, which is a way to answer our initial question with accurate data in a realistic application scenario.

3.2 Setup

3.2.1 Quality metric

To perform the analysis, a basic quality scheme has been created based on the MQM framework (Lommel et al. 2015). The preference of MQM over other existing quality models relies on the following determining factors: it is comprehensive, customisable, standard-oriented and free to use (Mariana et al. 2015, p.137). These elements provide the flexibility and generality for defining schemes for a variety of scenarios, as opposed to other more specialised models such as SAE J2450 (automotive). In addition to this flexibility, one particular advantage is that MQM is a recent outcome of a Framework Programme of the European Commission in translation technologies (Uszkoreit and Lommel 2013). This very nature let us assume its validity in absolute terms, as well as a modern approach compared to other models like LISA QA (shut down on 2011) or TAUS DQF (harmonized with MQM).

When taking MQM as a reference, the intention is to define a simple, yet realistic quality metric that could fit within a generic translation-revision cycle. The underlying motivation is to come up with a model that proves the tool capabilities in generic and standard scenarios. With this objective in mind, it seems relevant to bear in mind MQM compliance as defined in the Multidimensional Quality Metrics definition (Lommel et al. 2015). Because of its general and comprehensive nature, such compliance should strengthen the conclusions on the applicability of a generic quality scheme in today’s translation tools.

----------------------------------------

15 More information about the LISA operation shut down on Common Sense Advisory.
To achieve the objective of creating an MQM-compliant metric, the practical guidelines for the use of MQM in scientific research on translation quality will be applied as much as possible (Burchardt and Lommel 2014). Therefore, following the MQM guidelines (Burchardt and Lommel 2014, p. 5), the metric presented here is meant to meet the following criteria:

1. Its granularity must be adapted to the research question (1.2). In this case, we want to prove the capacity of the CAT tools to support predefined quality metrics with a minimum number of categories and subcategories.

2. Quality categories must be relevant for assessing the quality of human translation with as little noise as possible. This means that issue types should be common and understandable, without any redundancy or overlapping among them.

3. The number of the issues tested must be easy to handle by a professional linguist. Since the model is meant to be applicable in generic workflows, we intend to keep the number of items below ten, including categories and subcategories.

Considering the above criteria, the quality metric used in the analysis is composed of two category levels (a top-level and a subtype level) and four severities associated to penalty values as described in MQM definition (Lommel et al. 2015). By applying a two-level hierarchy, the present study intends to test whether the tool in question is able to make use of a quality framework of a minimum of complexity compared to a mere list of issue types. Also, using severity levels with the associated penalty values should help depict the way the tool flags the quality issues and computes the overall quality score for the translation.

### 3.2.1.1 Quality categories and hierarchy

The idea behind the mentioned metric is to avoid any domain bias so that it can theoretically be applied to any purpose, such as standard translation workflows, translator’s training or machine translation evaluation. Therefore, it has been designed as balanced and plain as possible, taking the MQM Core framework as a multi-purpose reference along with the recommendations related to its
application (Burchardt and Lommel 2014; Lommel et al. 2015). By extending the cases where a user or a company role are in a position to consume such metric, this study intends to ensure the generalisability of its conclusions beyond specific use cases and scenarios.

As a result, based on the MQM framework (ibid.), the quality metric used for this analysis holds four main categories or dimensions (top-level branches) and five subcategories, as shown in Figure 2.

![Figure 2. Quality categories hierarchy.](image)

The same hierarchy can be represented as a mind map (Figure 3):

![Figure 3. Quality categories hierarchy (mind map).](image)

As it can be observed, the metric has empty categories and categories with subcategories, which should let us analyse the behaviour of the software products

---

17 See 'Annex – Metric definitions' for the MQM Core reference consulted here.
18 Generated with Docear 1.2.0.
regarding a given multilevel structure as well as their capabilities in terms of issue category handling.

3.2.1.2 Severity levels

As we have seen, TQA is meant to measure the quality, hence the importance of the definition of severity levels. Always with the idea of analysing the application of a generic metric in translation software products, the default severity levels have been defined and weighted as in the MQM definition version 1.0 because of the universality they have been built on:

- **none**: 0 (to be applied when issues must be flagged but should not impact the overall quality score).
- **minor**: 1 (to be applied in case of issues that do not affect usability or understandability of the text).
- **major**: 10 (to be applied in case of issues that impact usability or understandability but do not make the text unusable).
- **critical**: 100 (to be applied to issues that make the content unfit for its purpose).

Although the default severities are theoretically three (minor, major and critical), the neutral severity (none) can help test the flexibility of the software tools, which may presumably be designed with only the three default levels.

3.2.2 Sample corpus

Although the linguistic content is not particularly relevant for the purpose of this analysis, which is focused solely on TQA software implementations, it may help illustrate the different cases where an issue is flagged and computed. Therefore, the texts chosen as test data for the analysis are taken from an already annotated corpus that comprises the categories defined earlier: the QTLaunchPad MQM Annotated Corpora, version 2.0. More exactly, the set used was the ‘Spanish-to-English Annotations - Round 1’ (v. 2.0).

---


In order to use the data as a base for the TQA, as well as to ensure the replicability of the tests, the following procedure has been adopted:

1. Downloading the XML corpus, from Spanish to English.
2. Extracting the source and target segments from the existing annotated corpus (originally formatted as XML) by opening the XML file as a table in Microsoft Excel.
3. Removing all the metadata columns and leaving ‘source’, ‘target’ and ‘issue type’, which hold only clean content without annotations.
4. Removing the duplicates to avoid having the same issue flagged for the same source segment in several alternative translations.
5. Saving the contents as plain text and tab-separated values.
6. Converting the text file into a generic XLIFF 1.2 file (using Okapi Rainbow), so that the bilingual document can be opened in any translation tool supporting XLIFF as source format.\(^{21}\)
7. Verifying that the resulting XLIFF can be processed with the translation tools to be examined.

The outcome of this procedure is, therefore, a clean bilingual XLIFF used for marking issues and testing the tools, as well as a reference corpus with issues flagged and annotated, so that relevant issues can be taken from there and reproduced in the software. Since the objective of the study is to demonstrate the TQA capabilities of the tools rather than aspects such as the performance or the productivity, the total number of the issues appears as not relevant here. In this sense, it has been verified only that there were several examples of each type of issue, which should not imply that every issue type will be reproduced in the translation tools. In fact, a single sample of an issue must be enough to demonstrate whether the software tool meets the criteria in most cases.

\section*{3.2.3 Analysis items}

We intend to perform the tests based on a series of evaluation points that will help us define the degree and type of support given to the quality assessment

\(^{21}\) The whole process is described in this page of the Okapi Framework wiki.
effort by the tools. The goal is not the evaluation of the linguistic side of the assessment, nor its questioning in terms of usefulness or relevance, but the sheer analysis of the implementation and usability aspects of the quality assessment features: how the quality assessment is performed inside the tool and how the corresponding data is stored and potentially exchanged beyond the tool. In the end, we must be able to state what are the strengths and weaknesses of the tools when carrying out a quality assessment exercise.

Therefore, our purpose is to compile a list of items that will serve as the instrument to evaluate in a comprehensive way the capacity of translation tools to allow the execution of TQA tasks. Thus, we understand that for such purpose, these items should allow us, at least, to test whether the tool lets the user:

1. Define a quality metric including an issue hierarchy.
2. Perform annotations in the bilingual files.
3. Handle penalties, issue weighting and quality scores.
4. Calculate, report and exchange the results.

To facilitate the analysis and the discussion, two principal areas are differentiated here: the implementation of TQA inside the tool (i.e. the aspects of the tool oriented to the TQA process itself), and the interoperability characteristics of the implementation (i.e. the aspects oriented to the exchangeability of the results). These two areas shall cover in consequence the whole experience of the quality assessment; the success of the tests should demonstrate the adequacy or inadequacy of using translation tools for such purpose over alternative solutions such as Excel scorecards or specialised tools.

By making the difference between implementation and interoperability, the assumption is that tools can excel in the design of the TQA feature, so that an standard TQA process can be executed without limitations, but fail when it comes to exchanging or sharing the results; or, on the contrary, show a very poor implementation in technical terms, but allow other systems and tools to consume TQA results in a standard and straight-forward manner. This distinction may help, therefore, to emphasise the current limitations and the evolution and trends of the software solutions in question. Of course, nothing prevents a given tool from being well or poorly designed on both perspectives at the same time, which is also relevant for our analysis.
According to this approach, Table 2 shows the list of items that will be examined and that have been compiled based on the reference MQM definition (Lommel et al. 2015) and the assumptions mentioned above:

<table>
<thead>
<tr>
<th>Area</th>
<th>Analysis item</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation</td>
<td>Built-in quality models</td>
<td>TQA process is supported by including existing quality model templates (MQM, TAUS DQF, SAE J2450, LISA QA).</td>
</tr>
<tr>
<td></td>
<td>Customising a metric</td>
<td>TQA process is supported by allowing the edition or modification of a quality model for specific purposes.</td>
</tr>
<tr>
<td></td>
<td>Issue hierarchy</td>
<td>TQA process is supported by allowing the creation of an issue hierarchy of at least a top-level (dimension) and a bottom level (subtype).</td>
</tr>
<tr>
<td></td>
<td>Severity levels</td>
<td>TQA process is supported by allowing the definition of severity levels associated to penalty values per severity level, including 'zero' value.</td>
</tr>
<tr>
<td></td>
<td>Issue weighting</td>
<td>TQA process is supported by allowing the definition of weights per issue (decimal value multipliers), being the default 1.0.</td>
</tr>
<tr>
<td></td>
<td>Issue flagging</td>
<td>TQA process is supported by allowing the user to mark the existence of issues in the text.</td>
</tr>
<tr>
<td></td>
<td>Markup scope</td>
<td>TQA process is supported by allowing the markup of issue with a variable scope in length, from zero characters to several segments in the source and in the target.</td>
</tr>
<tr>
<td></td>
<td>Overlapping spans</td>
<td>TQA process is supported by allowing the user to mark up issues that overlap partially or entirely in the text.</td>
</tr>
<tr>
<td></td>
<td>Split issues</td>
<td>TQA process is supported by allowing the user to mark up issues that are physically discontinued so that a minimal markup is possible.</td>
</tr>
<tr>
<td></td>
<td>Report generation</td>
<td>TQA process is supported by generating a report with the final details and final score from the evaluation.</td>
</tr>
<tr>
<td></td>
<td>Score calculation</td>
<td>TQA process is supported by the automatic calculation of an overall quality score.</td>
</tr>
<tr>
<td>Interoperability</td>
<td>Intermediate format</td>
<td>TQA interoperability is supported by using a format that allows to process the results of the evaluation by other tools.</td>
</tr>
<tr>
<td></td>
<td>Markup</td>
<td>TQA interoperability is supported by using a markup mechanism that lets other tools process the results of the evaluation.</td>
</tr>
<tr>
<td></td>
<td>Report compatibility</td>
<td>TQA interoperability is supported by using a standard report format.</td>
</tr>
<tr>
<td></td>
<td>Import/export</td>
<td>TQA interoperability is supported by allowing the user to import and export metric using standardised description rules.</td>
</tr>
<tr>
<td></td>
<td>MQM compliance</td>
<td>TQA interoperability is supported by following the normative MQM aspect in the annotated files.</td>
</tr>
<tr>
<td></td>
<td>ITS 2.0 markup compliance</td>
<td>TQA interoperability is supported by applying ITS 2.0 vocabularies for localisation issues.</td>
</tr>
</tbody>
</table>

Table 2. Analysis items.

### 3.2.4 Translation tools

The main criteria used for selecting the CAT tools analysed here is that they include built-in TQA features by default. Apart from this condition, the selection of tools has not been restricted to any platform or technology. Therefore,
it has been considered a range of solutions: from desktop application leaders\textsuperscript{22}, to open-source solutions or cloud-based systems, as they may target different types of audiences, involve different technical challenges and represent different business models.

As a first step, a set of translation tool candidates has been considered for this evaluation based on tools appearing in specialised manuals and studies (Morado Vázquez and Filip 2014; Chan 2015; Zetzsche 2017). However, it must be underlined that a number of those tools had to be left outside the analysis since they do not meet the main criteria: to include a TQA functionality out-of-the-box and not as a plug-in or third-party extension. Although the reasons for—and the consequences of—the lack of TQA features in translation tools are beyond the scope of the present study, it seems nonetheless relevant to mention here the tools that were tentatively approached for suitability in this analysis and that failed in meeting such criteria. As of today, the solutions considered for evaluation that do not document or noticeably implement any TQA feature as defined in the previous sections are the following:

- Déjà Vu X3
- MateCAT (default version)
- OmegaT (default version)
- Swordfish Translation Editor
- Transit NXT Professional+
- Wordfast Classic
- WordFast Pro

Regarding the tools that do meet the condition of implementing TQA enabling features or any functionality allowing the execution of TQA following the criteria described here, and that are chosen for the analysis are those shown in Table 3.

\textsuperscript{22} According to an infographic by be.international made in August 2007, the most named desktop tools in U.S. job postings are: SDL Trados, Wordfast, SDLX and MemoQ: \url{http://be.international/portfolio/infographic-translation-tech-skills-requested-in-200-job-postings/}. 

- 38 -
<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
<th>Application type</th>
<th>Licence type</th>
<th>Source code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluency Now(^{23})</td>
<td>4.14</td>
<td>Desktop-based</td>
<td>Subscription</td>
<td>Proprietary</td>
</tr>
<tr>
<td>MemoQ Translator Pro(^{24})</td>
<td>8.1.7</td>
<td>Desktop-based</td>
<td>Perpetual</td>
<td>Proprietary</td>
</tr>
<tr>
<td>SDL Trados Studio 2017(^{25})</td>
<td>14.1</td>
<td>Desktop-based</td>
<td>Perpetual</td>
<td>Proprietary</td>
</tr>
<tr>
<td>translate5(^{26})</td>
<td>2.5.33</td>
<td>Server-based</td>
<td>Free</td>
<td>Open-source</td>
</tr>
<tr>
<td>XTM Cloud(^{27})</td>
<td>10.4</td>
<td>Server-based</td>
<td>Subscription</td>
<td>Proprietary</td>
</tr>
</tbody>
</table>

Table 3. Translation tools with TQA features to be examined.

As it can be seen, the selected tools represent different software approaches, including desktop and cloud solutions as well as different business models (open-source, proprietary, free, subscription-based or under perpetual licence agreements). Although these aspects are not related to the TQA implementations themselves, and they are not taken into consideration in the analysis, they can give some room for consideration regarding the evolution of the translation software industry, as mentioned in the discussion of the results.

In addition, a note must be made on SDL WorldServer and Smartling, which provide environments for the execution of quality assessment tasks, and that could enter in a TQA analysis. However, these systems have been dropped from the analysis for the following reasons:

1. **SDL WorldServer** is not a translation software as the ones examined here, but a translation management system that focuses on workflows and project management over translation/edition features. Also, its Online Editor is a simplified version of SDL Trados Studio with the main difference of being web-based.
2. **Smartling** is a client-side solution rather than a standalone CAT tool. Although it includes a web-based module for localisation, its approach is completely different to the CAT tools analysed here.

\(^{27}\) https://xtm-intl.com/.
3.2.5 Testing environment

Except for the cloud offering by XTM, which was tested in the company’s trial server due to the lack of support for a local installation, the trial versions of desktop software products (Fluency Now, MemoQ and SDL Trados Studio), as well as the server-based solution translate5, were installed in a local testing environment, specifically in a virtual machine running Windows 10. Using a virtual machine for the analysis offered several advantages for the purpose of this study, namely:

1. Performing the tests in a clean machine with the essential software installed helped reduce the risk of conflicts among tools and in connection to the operating system.
2. Having the possibility to save relevant machine states (snapshots) at any point allowed to reimage the machine to previous states when necessary.
3. Setting the environment in a virtual machine allows replicating the tests in any other host platform under the same conditions.

3.2.6 Analysis of test results

The analysis consists of the evaluation of the results of the tests. To allow an easier representation of the information extracted from the previous exercise, we provide a comparative table summarising the points evaluated. Also, we produce a graphical representation of each tool regarding quality assessment support areas (implementation, interoperability, user-friendliness). A radar graph may help to illustrate the areas where the tools are stronger or weaker.

---

28 Regarding translate5, this solution is a server-based system and not a desktop tool. However, it can be installed and tested in a client machine using the installer provided in the company’s website that installs an Apache HTTP server to run the application locally. To reproduce the installation in a virtual machine using Oracle VirtualBox two actions—that are not triggered by the installer—must be considered: 1) Java runtime must be installed previously, and 2) the virtual drive must be defined as a solid-state drive (under Storage > Attributes) in case the physical hard disk is of this type.

29 The virtual machine used ran under Oracle VM VirtualBox Manager available at: http://www.oracle.com/technetwork/server-storage/virtualbox/downloads/
3.3 Conclusion

As a summary, the setup for carrying out the analysis of the translation software has included the gathering and execution of the following items:

1. A tool-agnostic quality metric to be applied within the software, comprising a minimal issue hierarchy of two levels, severity penalties and a scoring calculation, based on the MQM reference model as described MQM definition (Lommel et al. 2015).

2. A sample corpus composed of bilingual texts in two forms: first, clean source and target segments as a base to perform the TQA, and second, a reference version including issues and annotations, so that relevant issues could tentatively be reproduced within the CAT software to illustrate how each tool behaves using real cases.

3. A list of software characteristics to be examined in the translation tools that covers the essential aspects of the TQA activity as described in the MQM definition (ibid.).

4. A set of translation software products, which should have as a prerequisite the out-of-the-box implementation of TQA features.

5. An adequate and replicable testing environment for the installation of the products mentioned above and the execution of the analysis itself.
Chapter 4: Findings

4.1 Introduction

The present chapter expounds the analysis of different translation software products from the perspective of the TQA effort. As already mentioned, the goal is to examine in detail the different TQA enabling implementations, focusing on what can be done and what cannot be done by state-of-the-art CAT tools in terms of human assessment using the default TQA features offered by each product; that is, without QA automation or using plug-ins, extensions, or any other third-party tool or method.

4.2 Results

This section presents the results of the analysis carried out on the different tools. First, the focus is put on the TQA implementation within the tools; then, the evaluation concentrates on the interoperability aspects that allow TQA outcomes to be consumed by other systems and tools. The analysis is followed by a summary of the results.

4.2.1 TQA implementation

4.2.1.1 Introduction

The following subsections introduce the translation software tools and the concept they have of TQA: as a workflow step, as a review module, as a global feature or simply integrated into a translation editor. It also shows the TQA models they are based on: some are built integrating multiple models, others are based on a unique model. After the introduction, each analysis item is explained and tested. Finally, a simple score is given to each tool; this score should represent the degree of fulfilment demonstrated by the software regarding the item in question.

Fluency Now

Western Standard’s translation software Fluency Now offers TQA capabilities in the unique form of MQM integration. At product design level, the TQA functionality is not labelled as translation or linguistic quality assessment, but
directly as a command called “Multidimensional Quality Metrics”. This command, to be found in the ‘Tools’ menu of the toolbar, displays a set of panes for the MQM-based assessment. Unlike other tools, Fluency Now functionalist approach seems to add an MQM-exclusive assessment rather than an open TQA module where other models could fit in.

![Figure 4. MQM integration in Fluency Now.](image)

**MemoQ**

Kilgray labels the human quality evaluation as ‘linguistic quality assurance’ (LQA).\(^\text{30}\) The company also conceives LQA as a component of a translation project. In this sense, a project must be manually configured to have an LQA model activated; otherwise, by default, LQA cannot be performed in any document. This approach allows to assign a different quality model per project, but, on the other hand, once the model is chosen, it cannot be changed (to another model) or modified (for instance, adding new categories) for the documents belonging to the project. If there is any change to be made at the LQA level, the documents must be reimported to take the changes into account (see Figure 6).

**SDL Trados Studio**

SDL has implemented TQA features since its version 2015, but it has not been improved or changed in version 2017. Conceptually, TQA is understood by SDL as a freely accessible feature rather than a step in a workflow or a project component. This means that TQA can be performed at any moment in any document opened in the translation editor; in other words, it does not require the linguist to be assigned a TQA job or the document to be in a particular workflow stage.

On the other hand, the report generation is designed as an automated task and, as such, it can be integrated in predefined task sequences inside projects. For example, a task sequence called ‘Finalize’ could, in a predefined way, generate the final documents and a TQA report based on the TQA performed in the document, which adds some project management characteristics to it.

**translate5**

Contrasting the rest of the commercial products analysed here, translate5 is a free, open-source, cloud-based solution developed by independent developers. Also, unlike other software, translate5 has been designed to integrate the MQM
framework from the beginning in collaboration with the QTLaunchPad project, which has funded major features of the tool, and, among others, those related to MQM annotation and reporting. In translate5, MQM is integrated by default in the editor. Therefore, it can be applied directly when editing a task (document) without additional configuration. Thus, rather than a workflow step or a project component, MQM is implemented as an available interface command for the user.

![MQM pane in translate5.](image)

**XTM Cloud**

In XTM, TQA is also called LQA and, by default, is an optional workflow step that must be defined inside a translation project. In this aspect, it is designed as a distinct task, unlike other tools where TQA is performed as a software feature/action at any point. That said, it is also possible to modify any workflow step to include LQA; for instance, it is possible to add LQA marking capabilities to the step “Translate”. Also, the metric scope in XTM is the tool (Settings > QA > LQA), which prevents the possibility of having custom metrics per project, as seen in other products.

![Global LQA settings in XTM Cloud.](image)

**4.2.1.2 Built-in quality models**

Feature overview

This item refers to built-in quality metrics offered by the tools out-of-the-box, in order to facilitate the organisation or the user the task of carrying out the
translation evaluation. By ‘built-in quality models’, it must be understood that the TQA metrics are installed with the product and, in consequence, the assessment can be performed directly on the translated document without the explicit transformation or creation of a quality metric from scratch.

**Evaluation criteria**

This feature is proven when the product includes existing and independent quality models in a translation project or task without creating them manually. A rating of 1 point means that the tool provides more than one existing quality model out-of-the-box. A rating of \( \frac{1}{2} \) point implies that there is only one built-in model proposed. Otherwise, a rating of 0 is given to the products that do not include any model or the model included is incomplete or unusable.

**Fluency Now**

The only model offered by the tool is MQM. The default definition of the MQM metric can be loaded, in principle, from the Fluency local directory:

![Load MQM Definition](image)

Figure 10. Loading default MQM definition in Fluency Now.

However, the default MQM definition loaded this way is incomplete. The definition file used by the tool as of today is a non-existent “version 2.0” (see Annex – Metric definitions). According to the most recent version of the MQM definition, the last version available would be 1.0 (Lommel *et al.* 2015, chap.6.1). If versions used by MQM are tracked back, it emerges that Western Standard has taken the example given for the MQM metrics description in the MQM version 0.1.14 (2014-06-06),\(^3\) which is an outdated sample for the metric description (and that coincidentally had a wrong version number). Therefore, the MQM

---

metric is not provided as such by default, but an old example of the metric under a misleading label.

**MemoQ**

Kilgray includes several quality models in MemoQ, namely J2450, LISA, TAUS, as well as its own model. Unlike other tools, it also offers per-language built-in models, so that linguists working in different languages can make use of the metrics in an easier way. It is to be noted as well that it does not include MQM, unlike other translations tools analysed here.

![Figure 11. Built-in quality models in MemoQ.](image)

**SDL Trados Studio**

The tool offers different built-in models for quality assessment: LISA QA, MQM Core (version provided is outdated), SAE J2450 and TAUS_DQF. These quality models are available in project templates installed together with the application. For existing projects, it is possible to import those models if previously exported from the templates. However, it must be noted that without using a template or importing the settings, TQA default settings remain empty and therefore unusable in an ordinary project. Although this is not a blocking issue—since quality metrics can be imported at any moment—, the tool stays slightly obscure in this sense, causing the misleading impression that metrics must be necessarily built up from zero (Yang et al. 2017, p.7).
translate5 comes with a built-in MQM metric. This metric is automatically loaded into the task. In the event an import package contains a valid custom model according to MQM description rules, the custom model overrides the default MQM metric.\textsuperscript{32}

**XTM Cloud**

According to the company’s website, “the TAUS DQF has been fully integrated with XTM Cloud v10.1. XTM now supports sending both productivity data and quality review results based on the harmonised MQM-DQF error typology.”\textsuperscript{33} On the other hand, the XTM documentation says that “the LQA feature in XTM is a translation quality scoring system that is based on the MQM model which has been designed as part of the QT Launchpad project.” (XTM 2017, p.50).

Beyond the apparent lack of consistency in the XTM reference material, in reality XTM offers by default an MQM-based quality model called ‘LQA’ on one hand, and the TAUS DQF on the other, which requires a TAUS account to be displayed, activated and used. Although TAUS DQF is integrated as a plug-in under a subscription scheme, the model considered in this study as a built-in model available out-of-the-box is the one called ‘LQA’, which is a collection of the MQM/DQF unified categories.

**Conclusion**

<table>
<thead>
<tr>
<th>Product</th>
<th>Score</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluency Now</td>
<td>0</td>
<td>Only one incomplete model can be loaded by default.</td>
</tr>
<tr>
<td>MemoQ</td>
<td>1</td>
<td>Several quality models are included in the product.</td>
</tr>
<tr>
<td>SDL Trados Studio</td>
<td>1</td>
<td>Several quality models are included in the product.</td>
</tr>
<tr>
<td>translate5</td>
<td>$\frac{1}{2}$</td>
<td>Only one model is included in the product.</td>
</tr>
<tr>
<td>XTM Cloud</td>
<td>$\frac{1}{2}$</td>
<td>Only one model is included in the product.</td>
</tr>
</tbody>
</table>

*Table 4. Built-in quality models summary*

\textsuperscript{32} http://confluence.translate5.net/display/BUS/ZIP+import+package+format

\textsuperscript{33} https://xtm-intl.com/blog/xtm-tausdqf-together/
4.2.1.3 Customising a metric

Feature overview

This item refers to the possibility of creating or adapting issue categories to meet the specific requirements of a given project or task. For instance, machine translation output, localisation projects or even translation training programmes may require specific issue types. Customising a model can be performed either by creating a new quality model or by modifying existing models. At the very least, customising a quality model should involve the possibility of creating, removing and changing issue categories. This item does not refer to the issue hierarchy, the severity levels or the individual issue weights.

Evaluation criteria

The capacity of customising a quality model is demonstrated when the issue types of the metric proposed in this study can be created from scratch or can be derived from an existing model, by using the own tool. A rating of 1 is given when the issue categories can be created or fully adapted inside the tool; a rating of ½ is given to the tools that allow only an approximation of the target metric, or the definition must be edited outside the tool; a rating of 0 is given to the tools that do not support a custom metric.

Fluency Now

Fluency Now does not provide any way to create a custom quality model. There is, in fact, no means to modify an existing metric within the tool. On the other hand, it is possible is to load an existing custom metric based on the XML vocabulary provided by the MQM definition (Lommel et al. 2015). In consequence, it must be assumed that the tool does not allow to create a custom quality model, but it does allow to use custom MQM-compliant quality metrics to the extent intended in this study.

Nevertheless, according to the mentioned limitations, it is possible to create a description file with the metric of this study (see Fluency Now custom metric in Annex – Metric definitions), which demonstrated the capacity of the tool of using a custom metric.
Kilgray’s MemoQ exhibits a powerful interface that allows creating quality models in different ways:

1. Defining a new model from scratch.
2. Cloning other models that, in turn, can be edited later on.
3. Directly modifying existing model issue categories.
4. Importing a model from a proprietary settings file (non-standard).

In the case of this study, the TAUS DQF model was cloned and edited to successfully achieve the metric wanted.
As in other tools, custom models can be exported in a proprietary format and imported by other MemoQ instances.

Although not affecting directly the capacity of customising quality models, it can be observed that MemoQ is the only tool that allows to map LQA issue types to QA automatic checks when customising a quality model.

**SDL Trados Studio**

SDL Trados Studio offers three ways to customise quality issues in any translation project:

1. Creating categories manually via the project settings.
2. Creating a project based on a project template that includes a quality model and modifying the issues afterwards.
3. Importing a model using a proprietary settings file (non-standard) previously generated by the user (i.e. no settings files are provided out-of-the-box).

In this case, the model to be applied was configured using the built-in MQM template and refining it. However, it can be remarked that the MQM
template included in Studio is not up to date. In both Studio 2015 SR2 (released in March 2016) and Studio 2017 SR1 (released in July 2017), the MQM built-in template is based on the MQM definition version 0.1.0 of 19 March 2014, while the most recent and definitive version is 1.0 from 30 December 2015.

*translate5*

Just as Fluency Now, translate5 does not offer any way to customise a quality model within the tool. The only way to use a custom metric is by packing the MQM metric description and importing it when a task is created. This operation overrides the default MQM schema. Also, the metric description must conform to a specific tool file name and extension (*QM_Subsegment_Issues.xml*) instead of using the MQM recommend file extension, which may hinder the first approach to using custom metrics in the tool.

*XTM Cloud*

XTM does not offer a mechanism to create a brand-new custom metric from scratch, as other tools do. However, it does provide one default built-in model that can be modified, which eventually enables the definition of a custom model inside the tool. This customisation is made under global settings, which means that it cannot be applied differently according to the project, but must be considered as a user account configuration. Apart from this limitation, the system allows to add, remove or rename categories until reaching the model described in this study.

Also, it deserves consideration the fact that the default quality model includes the MQM naming associated to existing issues; that is to say, each default issue category has a field that shows the MQM issue name. MQM names cannot be changed, but if the issue category is removed and a new issue category is created, the MQM is not displayed. In any case, MQM issue labels are used as hidden identifiers that can help the organisation when using languages other than English for the categories or the severities, but they do not seem to be critical for the TQA effort itself.
Conclusion

<table>
<thead>
<tr>
<th>Product</th>
<th>Score</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluency Now</td>
<td>½</td>
<td>The issue categories need to be edited in XML outside the tool.</td>
</tr>
<tr>
<td>MemoQ</td>
<td>1</td>
<td>The custom metric can be fully created inside the tool.</td>
</tr>
<tr>
<td>SDL Trados Studio</td>
<td>1</td>
<td>The custom metric can be fully created inside the tool.</td>
</tr>
<tr>
<td>translate5</td>
<td>½</td>
<td>The issue categories need to be edited in XML outside the tool.</td>
</tr>
<tr>
<td>XTM Cloud</td>
<td>1</td>
<td>The custom metric can be fully created inside the tool.</td>
</tr>
</tbody>
</table>

Table 5. Customising a metric summary

4.2.1.4 Issue hierarchy

Feature overview

This item refers to the capacity of establishing an issue hierarchy that organises issue types in a rational manner, including the possibility to define main categories and subcategories, with the purpose of enabling higher issue granularity and helping assessors apply the quality model in a more accurate way when necessary.

Evaluation criteria

This item is proven when the software supports issue hierarchies natively, that is, when the product provides a way to organise the issues within a layered structure with, at least, a top-level dimension and subtype branches. A rating of 1 is given to the products that allow reproducing the hierarchy of the tested metric fully (i.e. two-level structure) and offer potential for an extension into deeper levels. A rating of ½ is given to the products that only allow a hierarchy of two levels and/or do not offer the possibility of using top-level categories (resulting eventually in a list of issues rather than a multilevel schema). A rating of 0 is given to the products that do not allow a hierarchy, but provide only support for lists of issues.
**Fluency Now**

Since Fluency Now does not include any way to create or edit an error hierarchy, it is not possible to affirm what hierarchy depth is supported from the tool itself. On the other hand, since the application is able to interpret an MQM hierarchy that is described using MQM description rules, it is possible to confirm that several levels can be created below the top-level categories. Therefore, with respect to the model proposed in this study, the application is able to import the corresponding hierarchy including four top-level categories and five subtypes without any problem:

![Fluency Now hierarchy](image)

*Figure 15. Issue hierarchy in Fluency Now.*

**MemoQ**

MemoQ allows to create a simple hierarchy composed by categories and subcategories. Categories that do not have subcategories can be used as such. On the other hand, categories that do have subcategories cannot be used for flagging issues alone. For instance, if a given top-level category has two subcategories, that category *tout court* cannot be used as such to flag issues. Even if workarounds can be found to overcome this limitation, this implementation remains more rigid than other, and hinders some assessment methodologies based on decision trees (e.g. if subcategories do not match the issue type, the main category must be used).

If a third level was necessary (i.e. a subtype under an error type under a dimension), the only way to achieve such hierarchy would be to label the error type with the subtype altogether (e.g. “Mistranslation > Overly literal”). In reality, this technique allows as many levels as necessary by simply extending the names of the error types, but produces an unnecessarily unclear metric without actually using a true hierarchy easily exploitable at reporting level.
**SDL Trados Studio**

SDL Trados Studio supports the definition of a minimalistic hierarchy of two levels composed by categories and subcategories. As in MemoQ’s case, it is important to note that a main category including subcategories cannot be used as an error type alone. That is, for a main category (top-level) to be applied as an error type it is necessary to leave it empty. This leads to a situation whereby the organisation must decide to either create two levels with the same name (for instance, “Accuracy” as a subcategory under “Accuracy” as a main category), or to define enough error types so that all potential issues can be marked with the existing error types without the need of invoking a generic top-level category.

As in the case of MemoQ, if a third level was necessary, it could be achieved only by labelling the subcategory together with the subtype, as the built-in models actually illustrates. Similarly, this workaround would let as many levels as necessary by extending the names of the error types, but it would produce an unclear metric without actually using a true hierarchy.

![Figure 16. Issue hierarchy in SDL Trados Studio 2017.](image)

**translate5**

translate5 does not provide any mechanism to manage a quality metric within the tool, preventing the user to create or modify an issue hierarchy as a result. On the other hand, it allows to import a simple issue hierarchy using its own XML format (see Annex – Metric definitions). Although this process seems barely documented, the XML vocabulary used for describing the issue categories and the hierarchy is simple enough to manipulate it in order to accommodate
different kinds of issue hierarchies. If the format, as well as the file name and the extension for the XML containing the issues, are respected, the hierarchy is imported together with the bilingual document when creating a task.

**XTM Cloud**

In XTM 10.4, it is possible to create a hierarchy of several category levels, which makes its edition flexible enough to accommodate reasonably granular metrics. The two levels of the quality schema are applied as presented in Figure 17.

![Figure 17. Issue hierarchy in XTM 10.4.](image)

Unlike other tools, XTM allows using top-level categories as error types. That is to say, a quality issue can be indicated as a top-level issue (e.g. ‘Accuracy’) while having subcategories without preventing the user to apply the main category to flag any issue not covered by the subcategories. Therefore, the metric does not need to cover cases where no subtype for a given issue is adequate since the top-level category can be used instead.

**Conclusion**

<table>
<thead>
<tr>
<th>Product</th>
<th>Score</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluency Now</td>
<td>1</td>
<td>Issue hierarchy fully supported.</td>
</tr>
<tr>
<td>MemoQ</td>
<td>½</td>
<td>Issue hierarchy limited to two levels and top-level restricted.</td>
</tr>
<tr>
<td>SDL Trados Studio</td>
<td>½</td>
<td>Issue hierarchy limited to two levels and top-level restricted.</td>
</tr>
<tr>
<td>translate5</td>
<td>1</td>
<td>Issue hierarchy fully supported.</td>
</tr>
<tr>
<td>XTM Cloud</td>
<td>1</td>
<td>Issue hierarchy fully supported.</td>
</tr>
</tbody>
</table>

*Table 6. Issue hierarchy summary.*
4.2.1.5 Severity levels

Feature overview

This item refers to the possibility of accommodating different levels of relevance or impact for each issue type for reporting, statistical and scoring purposes. The three levels taken as a reference from the MQM definition are: minor, major and critical (Lommel et al. 2015). A fourth level could also be considered for neutral issues that do not impact quality and whose penalty may be set as null (none), but which still need to be flagged for the sake of completeness or consistency. For a maximum flexibility of the quality model, severity levels should be defined independently from the issue types, that is, issue types should not be forced to have a one-to-one relationship with the different severity levels (e.g. a terminology issue might need to be flagged as a critical, major or minor issue depending on the case or the requirements for a particular project).

Evaluation criteria

To consider this item as proved, the translation tools must offer the possibility of defining and using different severity levels. If the product allows the customisation of at least four custom severity levels together with penalty values from 0 to 100, a rating of 1 point is given. If the product offers built-in non-editable severity levels but still conforms to the quality metric of this study in its basic levels (minor, major and critical), a rating of ½ is given. If the product does not offer severity levels or those available do not conform to the three basic levels, a rating of 0 is given.

Fluency Now

The default severity levels are three: minor, major and critical. Even though the tool loads the metric description from an MQM-compliant XML file, which allows to define different severity levels, this possibility is something that cannot be achieved in practice in Fluency Now. The severity levels are hard-coded and appear in the interface as non-customisable items.
As a result, it is unworkable for a user to add a fourth severity for neutral issues with zero penalty value.

**MemoQ**

MemoQ supports the customisation of the severity levels, which can be added, deleted, and renamed, but it does not offer the possibility of assigning them a penalty value independent of the issue types. So, for instance, the severity critical cannot be associated with a global penalty value of 100, as it is the case in the MQM framework (Lommel et al. 2015, chap.8.1). Nevertheless, it associates each severity level with a penalty value per issue type. As it will be recalled later, what this approach provides is a mechanism to weight issue types without the need of another parameter (i.e. specific issue weights). In other words, what in models like MQM is codified as a global penalty value plus different per-issue multipliers, in MemoQ it is a penalty value that can be adjusted in a per-issue basis.

**SDL Trados Studio**

SDL Trados Studio offers the same approach as MemoQ regarding the definition of the severities. In Studio, any number of severity levels can be created, renamed or deleted inside a project or in a project template. As in MemoQ, penalties are not associated to global numeric values, but each issue type may have a different penalty value according to the severity, which removes the need of a weight or multiplier per issue type.
translate5 offers three default severities coming from the MQM scoring recommendation (Lommel et al. 2015, chap.8): minor, major and critical. They cannot be changed, deleted or renamed in any way inside the tool. So, for example, it is not possible to add a neutral severity with zero penalty. It is also unclear if there is a means to modify the values of the penalties, since the issue description does not allow to define severities nor issue weights. Such process seems undocumented, and the only customisation possible seems to refer exclusively to the issue types and the hierarchy. However, being translate5 an open-source solution, it is highly plausible that there exists a mechanism to configure these settings via extensions or additional parameters.

**XTM Cloud**

XTM offers a halfway solution between fully customisable tools like MemoQ and Trados Studio, and those with more rigid default configurations like Fluency or translate5. In XTM, default severities are predefined and cannot be changed: neutral, minor, major and critical. However, it is possible to change the values of the penalties. These values are global and not linked to issue types. Also, its scope is the user account, which means that they affect all projects created under the active user account.
## 4.2.1.6 Issue weighting

### Feature overview

This item refers to the possibility of modifying the magnitude of a particular issue so that the overall quality is determined accordingly. There are two possible ways to weigh single issues: 1) by assigning a numerical value applied as a multiplier of the severity penalty, being 1.0 the default weight that does not modify the penalty and 0 the weight for issues that must be flagged but not counted; and 2) by directly modifying the penalty points per issue type and severity, so that each issue type/severity pair must have a numeric value from 0 to any value.

### Evaluation criteria

This item is demonstrated if the software provides a way to modify the final penalty points per issue, either by using a multiplier of the severity penalty or by entering specific values per issue type and severity pair. A rating of 1 is granted to the tools providing a way to weigh categories so that they can have final penalty values ranging from 0 to any numeric value. A rating of $\frac{1}{2}$ reflects tools that compute weightings, but do not provide a way to modify them in a direct way. A rating of 0 indicates that the tool does not take into account issue weighting.

**Fluency Now**

Fluency Now computes issue weights as defined in the MQM metric definition loaded into the tool; that is by applying an issue multiplier, whose default value is 1.0. If the weight value is set to a higher or lower value than 1.0, then the overall quality score is affected accordingly. For instance, the two following examples produce different overall quality scores. In the first case, the tool...
computes the issue ‘mistranslation’ with the default severity penalty marked in the text (i.e. 100 for critical issues, 10 for major issues, etc.):

```xml
<issue type="accuracy" weight="1">
    <issue type="mistranslation" weight="1.0"/>
</issue>
```

In the second case, the metric comprises a non-default weight for ‘mistranslation’, which results in mistranslation issues counting half of the default severity penalties (i.e. 100*0.5 for critical issues, 10*0.5 for major issues, etc.):

```xml
<issue type="accuracy" weight="1">
    <issue type="mistranslation" weight="0.5"/>
</issue>
```

If no weight is set for an issue, the following warning message pops up:

![Figure 22. Flagging an issue with no weight set in Fluency Now.](image)

It must be noted that this warning is shown even though the selected issue is not a top-level issue. For instance, when the metric description is as follows and the selected issue is ‘spelling’ (i.e. a subtype):

```xml
<issue type="fluency" weight="1">
    <issue type="spelling"/>
    <issue type="unintelligible" weight="1"/>
</issue>
```

Also, if a top-level issue is added a weight, as in the previous sample for the ‘fluency’ case, then the issue can be flagged, regardless the fact that it is a non-empty category, unlike it happens in tools where top-level categories with subcategories cannot be used to flag issues.

**MemoQ**

The mechanism used in MemoQ to weigh issue types does not use multipliers per issue type, but lets the user set specific penalty values per issue/severity pair. Thus, after having created severity labels; each issue type must receive a penalty value per each severity created. This way, the total number of penalty values correspond to the number of severity levels multiplied by the number of issues (in the metric used here, this means 9 issue types × 4 severity levels = 36
penalty values). Also, to achieve the same effect of a multiplier of 0, the severity can be set to 0 on per-issue basis.

**SDL Trados Studio**

As in MemoQ, issue weights in SDL Trados Studio are implemented via specific penalty values assigned to existing severity levels on a per-issue basis. As we have seen, instead of having a single multiplier per issue type as a weighting mechanism, the tool allows to specify an absolute penalty per issue/severity pair. Compared to the multiplier mechanism where the severity penalties are decoupled from the issue weights, this approach embeds both concepts (i.e. severities and weights) into one single matrix, increasing the granularity at the expense of the simplicity:

![Figure 23. Issue weighting in SDL Trados Studio.](image)

**translate5**

translate5 implements the MQM mechanism of assigning weights per issue type, but does not allow to modify the weights inside the tool nor via the issue description imported with the package in the form of XML file (`QM_Subsegment_Issues.xml`).

**XTM Cloud**

In XTM, severity levels are configurable regarding the penalty values which are already predefined (neutral, minor, major and critical). Since these values are independent from the issue types, the weighting mechanism make use of single multipliers per issue type. For that purpose, each issue type —regardless it is an empty category, a category with subtypes or a subtype— has a weight associated.
Conclusion

<table>
<thead>
<tr>
<th>Product</th>
<th>Score</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluency Now</td>
<td>1</td>
<td>Issue weights are defined in the MQM metric description.</td>
</tr>
<tr>
<td>MemoQ</td>
<td>1</td>
<td>Issue weights are defined in an issue/severity matrix.</td>
</tr>
<tr>
<td>SDL Trados Studio</td>
<td>1</td>
<td>Issue weights are defined in an issue/severity matrix.</td>
</tr>
<tr>
<td>translate5</td>
<td>½</td>
<td>Issue weights are not directly customisable.</td>
</tr>
<tr>
<td>XTM Cloud</td>
<td>1</td>
<td>Issue weights are defined in the metric description.</td>
</tr>
</tbody>
</table>

Table 8. Issue weighting summary.

4.2.1.7 Issue flagging

Feature overview

This item refers to the possibility of marking issues in the source and the target texts, so that they can be easily identified.

Evaluation criteria

This item is demonstrated when the application allows to insert a mark in the text so that the issue is tagged or emphasized visually. A rating of 1 is given to the implementations that allow flagging the issues in the source and target texts, marking the start and the end at a sub-segment level. A rating of ½ is given to the solutions that allow flagging the start and the end of the issue in the target text only. A rating of 0 is given to the solutions that do not allow the user to mark the issue in the text.

Fluency Now

To flag an issue in Fluency Now, the user can either add an MQM issue using the context menu by right-clicking on the segment and choosing “Add MQM Issue”, or by selecting a part of the text and then clicking on any of the issue
types in the issue hierarchy. In the latter case, the part of the text flagged is highlighted with a colour that corresponds to the severity of the issue (yellow for minor, orange for major and red for critical).

A characteristic of flagging issues in Fluency is the fact that, once the issue is marked and highlighted in the text, it is not possible to know which highlight refers to which issue if more than one issue is marked in a segment. Only the “Notes” field could give a hint about the issue and its location in the text.

Unlike other tools, Fluency allows flagging the source text. On the other hand, an issue cannot be marked in both the source and the target text at the same time. Issues flagged in the source text are counted separately.

Figure 25. MQM pane in Fluency Now.

MemoQ

This tool implementation offers the possibility of marking the whole segment or a part of it. The way this operation is done is by selecting the text and rejecting the selection by clicking on a “Reject” button, which automatically flags the selection and opens a dialog to insert the details of the issue. The tool does not offer any visual hint on the severity of the issue; that is, all quality assessment issues are marked with a red highlight. However, it is possible to hover over the issue to see the type of issues marked as well as the originator of the flag.

- 64 -
Figure 26. Flagging an issue in MemoQ.

Figure 27 shows the dialogue box to categorise an issue in MemoQ.

Figure 27. Categorising an issue in MemoQ.

A limitation here is that it is only possible to flag either empty categories or subcategories, but not top-level categories when they contain subcategories. In other words, the user is always obliged to use the bottom level of the hierarchy. To circumvent this constraint, the tool allows two workarounds: either to create all main categories with and without subcategories; or create as many subcategories as necessary so there is always a subcategory that is adequate to a given quality issue (i.e. all potential issues are covered by subcategories).

SDL Trados Studio

The mechanism implemented for flagging issues is inspired in the “tracked changes” and “comments” features of Microsoft Word. To mark an issue, the user must either add or remove text, or add a comment to a text selection. When a deletion or addition is registered, two events are triggered: first, the formatting of the edited text changes automatically and it becomes underlined in case of an addition and crossed out in case of a deletion (besides a different fore and back colour); second, the tool offers the possibility to categorise the issue and annotate
it. In case of just a part or the entire segment is selected and a TQA comment is added, the text is simply highlighted and the dialogue box opens to categorise the issue.

Figure 28. Flagging an issue in SDL Trados 2017.

The dialogue box to categorise an issue can be seen in Figure 29.

Figure 29. Categorising an issue in SDL Trados Studio.

As in MemoQ, top-level categories cannot be marked if they include subcategories, in which case a subcategory must be chosen.

translate5

translate5 offers a unique approach to issue flagging in the sense that, instead of highlighting the issue, after selecting the corresponding fragment, it marks the start and the end of the issue with squared brackets and the code of the issue category.

Figure 30. Flagging an issue in translate5.
Unlike other tools, translate5 does not provide a dialog box to enter the details, but only a side pane where the issue category must be selected in order to mark the issue, as it can be seen in Figure 31.

![Figure 31. Categorising the issue in translate5.](image)

The analysis performed on translate5 did not allow to demonstrate the possibility of marking issues in the source text in a default editing view.

**XTM Cloud**

XTM 10.4 allows indicating the existence of one or more issues at the segment level, but the user cannot mark or annotate the issue in the text. As it can be seen in Figure 32, if the revision is performed at the same time of the assessment, an auto-comment is added at the bottom of the segment showing the change made in the segment. Thus a user can correlate the quality issue with the edition displayed in the comment.

![Figure 32. Flagging an issue in XTM 10.4.](image)

**Conclusion**

<table>
<thead>
<tr>
<th>Product</th>
<th>Score</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluency Now</td>
<td>1</td>
<td>Issues can be flagged precisely in the source and the target text.</td>
</tr>
<tr>
<td>MemoQ</td>
<td>½</td>
<td>Issues can be flagged precisely in the target text.</td>
</tr>
<tr>
<td>SDL Trados Studio</td>
<td>½</td>
<td>Issues can be flagged precisely in the target text.</td>
</tr>
<tr>
<td>translate5</td>
<td>½</td>
<td>Issues can be flagged precisely in the source and the target text.</td>
</tr>
<tr>
<td>XTM Cloud</td>
<td>0</td>
<td>Issues cannot be flagged in the text.</td>
</tr>
</tbody>
</table>

*Table 9. Issue flagging summary.*
4.2.1.8 Markup scope

Feature overview

This item refers to the capacity of establishing boundaries for the issue flags so that issues can be properly marked, including their exact extension.

Evaluation criteria

This item is demonstrated when issues can be marked with character precision and without any constraint. A rating of 1 is given to the implementations that allow marking from zero characters to several segments, including the entire text. A rating of \( \frac{1}{2} \) is given to the implementations that allow flagging from one or more character to the whole segment, but cannot flag beyond the segment limits. A rating of 0 is for the solutions that do not allow marking the exact extension of the issue.

Fluency Now

When marking the source text, there is no limit to the text selection and subsequent flagging. In practice, this means that all source text can be highlighted to indicate a potential global issue. Likewise, an issue mark can cover two or more source segments. On the opposite side, the minimal markup scope is zero characters: by right-clicking on any part of the source or target text, and MQM issue can be added (the issue is counted, but without any visual hint).

Regarding the target segments, since the tool displays only one target segment in the translation pane, the maximum markup scope is the segment itself, unless segments are merged (provided that the document type allows such merging action).

MemoQ

The scope for the issue markup in MemoQ’s is the segment. If the user tries to flag an issue beyond the segment boundaries, a warning is shown indicating that it is not possible to reject more than one segment (Figure 33). This limitation does not apply when two segments are merged, in which case the new segment can be marked without the previous limitation.
Unlike in Fluency Now, issue flags need a minimal text span to be marked. If the user does not select at least one character, the whole segment is marked by default.

**SDL Trados Studio**

By default, the largest annotation scope is the segment, which is originally a limitation of the XLIFF 1.2, as seen in 2.4. An annotation that would require marking a fragment starting in one segment and ending in the following segment would only be possible by merging the segments.\(^{34}\)

In the version tested, the smallest annotation scope is one character. Since TQA is implemented as a change tracking mechanism, it requires a minimal span. Therefore, if no text is selected, it is not possible to indicate an issue.

**translate5**

translate5’s interface does not provide an obvious way to mark an issue beyond the segment boundaries. Although the principle of using square brackets and issue codes (i.e. \([1...1]\)) should allow such possibility, it does not seem possible to place the opening mark and the ending mark in different segments. Also, it seems unworkable to merge segments so that issues can be marked beyond the standard segmentation.

The minimal mark for an issue flag is 0 characters.

---

\(^{34}\) Note that merging segments is restricted to certain formats and document structure items in the version tested (SDL Trados Studio 2015).
**XTM Cloud**

Strictly speaking, the extent of the issue mark in XTM is always the segment. This is due to the fact that the tool does not offer any means to mark or annotate text spans directly in the segment. The only way to specify which part of the segment presents the issue is by using a comment, which would involve describing the issue or quoting the corresponding fragment.

**Conclusion**

<table>
<thead>
<tr>
<th>Product</th>
<th>Score</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluency Now</td>
<td>1</td>
<td>Issues can be flagged from zero characters to several segments.</td>
</tr>
<tr>
<td>MemoQ</td>
<td>½</td>
<td>Issues can be flagged from one character to one segment.</td>
</tr>
<tr>
<td>SDL Trados Studio</td>
<td>½</td>
<td>Issues can be flagged from one character to one segment.</td>
</tr>
<tr>
<td>translate5</td>
<td>½</td>
<td>Issues can be flagged from zero characters to one segment.</td>
</tr>
<tr>
<td>XTM Cloud</td>
<td>0</td>
<td>Issues can only be flagged at segment level.</td>
</tr>
</tbody>
</table>

*Table 10. Markup scope summary.*

4.2.1.9 **Overlapping spans**

**Feature overview**

This item refers to the tool’s capacity of having two or more issues flagged within the same text span.

**Evaluation criteria**

This item is proven when the software supports marking more than one issue in the same text regardless of the start and end points of each issue. A rating of 1 is given to the implementations that allow the issues to fully overlap and that give an indication of each one’s start and end. A rating of ½ is given to the implementations that do support the overlapping at markup level, but do not offer a visual hint on the overlapping spans. A rating of 0 is given to the implementations that do not offer the possibility of overlapping issue marks both visually or at markup level.

**Fluency Now**

Visually, Fluency Now shows highlighted text in a flat and exclusive manner, which means that the user cannot know if two or more spans do overlap:
simply, the last highlight overwrites any underlying existing highlight, regardless the severity, the issue type or the location.

At markup level, though, Fluency Now uses a proprietary XML vocabulary (FTFX) that embeds RTF-based text. This allows the text spans to effectively overlap, as RTF marks are more permissive than XML rules in terms of structure. Therefore, since RTF marks do not need to be nested as XML tags, overlapping marked fragments of text are technically possible, as we can see in the following sample where the translation fragment “the elegance to the notoriety” is marked as a ‘style’ issue and “notoriety” as a ‘mistranslation’ issue:

```xml
<SourceText>Valentino prefiere la elegancia a la notoriedad</SourceText>
<TargetRTF>{\rtf1\s1\fs24\h\an\deff0\ftnbj{\f1\fsnil Arial;}{\f1\fsnil \fcharsize0 Arial;}{\colortbl;\red255\green255\blue255;\red0\green0\blue0;\red0\green0\blue0;}{\stylesheet{\fs24 Normal;}{\cs1 Default Paragraph Font;}}{\revtbl{Unknown;}\paperw12240\paperh15840\margl1800\margr1800\margt1440\margb1440\headery720\footery720\headhesive720\foothesive720\margshade\fet4\aendnotes\aftnnrlc\pgbdrhead\pgbdrfoot\sectd\pgbsx12240\pgbsx15840\guttersx0\marglxsx1800\margrsxsx1800\margtxsx1440\margbsx1440\headery720\footery720\sbkpage\pgncont\pgndec\plain\f1\fs24\lang1033\lch\f1\dch\f1\loch\f1\cf2\fs24 Valentino prefers the elegance to the notoriety\par }}}</TargetRTF>
```

The advantage for the user is, in principle, limited, as they cannot see such overlapping properly represented on screen. However, the spans and its potential overlapping are kept intact under the hood, which can be useful at a later stage, e.g. removing one of the overlapping issues, would automatically reveal the underlying one. In other words, there is no loss of information in case of overlapping spans.

**MemoQ**

MemoQ does offer the possibility of marking as many issues as necessary in the same fragment of translated text. However, since all issues are marked with the same highlight colour, it is not possible to differentiate the limits of the issues when they overlap. Nevertheless, the user can still hover over the marked issues to know what the issues marked are (see Figure 34). Also, the View pane shows the currently marked issues for the active segment with further details (see Figure 35).
Figure 34. Overlapping issues in MemoQ.

Figure 35. MemoQ’s View pane.

**SDL Trados Studio**

Flagging two issues that overlap is not possible in Studio. When flagging a text part that overlaps one existing mark, Studio throws a generic exception (“Object reference not set to an instance of an object”) in the version tested. This behaviour can be replicated, which denotes the software is not currently able to handle this use case. Presumably, this behaviour may be a consequence of using a “tracked changes” model for the TQA.

**translate5**

Thanks to its markup approach, translate5 offers a solid support for overlapping spans. By using single start and end marks only, and not extended highlights, this implementation allows the user to mark as many issues as necessary in the same text span. Moreover, it also offers a clear vision of what issues are marked and to which part of the text they concern (see Figure 35).

Figure 36. Overlapping spans in translate5.

**XTM Cloud**

XTM does not allow the marking of text spans; thus, it is not possible to produce overlapping spans.
Conclusion

<table>
<thead>
<tr>
<th>Product</th>
<th>Score</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluency Now</td>
<td>½</td>
<td>Issue overlapping is not shown but it is supported at markup level.</td>
</tr>
<tr>
<td>MemoQ</td>
<td>½</td>
<td>Issue overlapping is not shown but it is supported at markup level.</td>
</tr>
<tr>
<td>SDL Trados Studio</td>
<td>0</td>
<td>Issue overlapping is not supported.</td>
</tr>
<tr>
<td>translate5</td>
<td>1</td>
<td>Issue overlapping is clearly shown and supported at markup level.</td>
</tr>
<tr>
<td>XTM Cloud</td>
<td>0</td>
<td>No issue flagging possible.</td>
</tr>
</tbody>
</table>

Table 11. Overlapping spans summary.

4.2.1.10 Split issues

Feature overview

This item refers to the possibility of applying a minimal markup to the text whereby an issue that concerns two separated words in the text can be marked by flagging only the words in question and not the whole span of text (e.g. for agreement, consistency or style issues).

Evaluation criteria

This item is proven when the software gives the possibility of flagging an issue in two or more locations, yet being counted as one single issue. A rating of 1 is given to the solutions that support flagging one issue in two or more different locations and computing it as one single issue. A rating of ½ is given to the solutions that require the two or more locations to be covered by the same mark span, but provides alternative solutions such as marking one location with a penalising severity and the other location with a neutral severity, so that the second mark does not impact the overall quality score. A rating of 0 is for the implementations that do not support splitting the marks nor give the option of marking one of the locations with a neutral penalty.

Fluency Now

In Fluency Now, it is not possible to mark one issue discontinuously, since the markup is done by selecting the text and flagging the selection. If the selection of text could be done in different parts at the same time, the RTF-based format should allow such mark splitting, though. In consequence, as it is currently implemented, an issue marked in two locations counts as two issues.
MemoQ

Since MemoQ flagging mechanism is based on the text selection by the user, it is not possible to mark the same issue in separated locations. It is, nevertheless, possible to mark the first span as a single issue, and then mark the second and subsequent spans belonging to the same issue, with the same category but with a neutral severity. Also, as MemoQ is able to overlap spans, marking long spans is not as problematic as with tools that do not support more than one coincident span, where one long span would prevent other issues to be marked.

SDL Trados Studio

SDL Trados Studio does not support flagging an issue in separate locations of a text. While it is possible to flag words in two distinct locations, each marking is counted separately. Since Trados Studio does not support overlapping spans, issues that need to be marked in long spans may be problematic in the event other issues have to be marked in between. This problem is illustrated in Figure 37, where a fragment of the translation is flagged as unintelligible (“A link Sybase rid positive”), but there is also a mistranslation tagged (“rid”) inside:

![Figure 37. Marking an issue in distinct places in SDL Trados Studio.](image)

If the longest issue is marked twice, as in Figure 37, the issue is counted double:

![Figure 38. TQA pane showing a split issue in SDL Trados Studio.](image)

Figure 39 shows a style issue marked in two different locations so that minimal markup is preserved; to avoid the issues be counted as double, the reviser can mark the first occurrence with a penalising severity and the second occurrence with a neutral severity, so that the overall quality score is not altered.

![Figure 39. Split issues in SDL Trados Studio.](image)
translate5
does not offer the possibility to split issue spans. Although the markup mechanism should allow such possibility—as it does not rely on continuous highlighted spans—the interface does not provide any means to accomplish such markup. Also, as seen in 4.2.1.5 (Severity levels), the alternative of marking issues with a neutral severity is not offered by default.

*XTM Cloud*

Since XTM does not allow the marking of subsegment spans, and issues are counted manually only at segment level, the user can only flag one single issue regardless the fact the issue appears in various parts of the segment.

**Conclusion**

<table>
<thead>
<tr>
<th>Product</th>
<th>Score</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluency Now</td>
<td>0</td>
<td>Split issues not supported and neutral severity not applicable.</td>
</tr>
<tr>
<td>MemoQ</td>
<td>½</td>
<td>Split issues not supported but neutral severity applicable.</td>
</tr>
<tr>
<td>SDL Trados Studio</td>
<td>½</td>
<td>Split issues not supported but neutral severity applicable.</td>
</tr>
<tr>
<td>translate5</td>
<td>0</td>
<td>Split issues not supported and neutral severity not applicable.</td>
</tr>
<tr>
<td>XTM Cloud</td>
<td>0</td>
<td>Split issues not supported and neutral severity not applicable.</td>
</tr>
</tbody>
</table>

*Table 12. Split issues summary.*

**4.2.1.11 Report generation**

**Feature overview**

This item refers to the possibility of generating a quality report based on the assessment performed on the bilingual document.

**Evaluation criteria**

The item is demonstrated in the case the software provides a direct way to generate a quality report including—as a bare minimum—the issues marked and the overall quality score. A rating of 1 is given to the solutions that are able to generate full reports including all the details of the assessment, including marks, issue counts, and an overall quality score. A rating of ½ is given to the implementations that produce reports that do not include all the assessment details (e.g. showing only statistics, but not text marks). A rating of 0 is for the solutions that
do not provide a way to generate reports or for those that do generate reports, but they do not include quality scores.

**Fluency Now**

Under the MQM menu, the tool has a report generation command that allows saving the TQA statistics into a plain text file (Figure 40). This report shows the source/target location, the severity and the number of each issue flagged in the text. It also shows the overall quality score. Since the metric used in the analysis contains some customisation compared to the default model, a new dimension ‘Custom’ is added at the end of the report. On the other hand, the category ‘Verity’ is displayed in the report, although the metric loaded (see Annex – Metric definitions) did not have it.

![Figure 40. TQA report in Fluency Now.](image)

**MemoQ**

MemoQ features a complete report generation functionality that include a number of options. First, it allows to generate statistics and error lists separately. Also, the user can choose the report scope: project (Figure 41), active document (Figure 42), from cursor, selection, etc. Likewise, it is possible to group errors per
categories or severities as well. Finally, a choice is offered to select the word count algorithm to be used (either MemoQ or Trados 2007).\footnote{Trados 2007 is a product that is not any more in the market, nor SDL gives any more support to it since 2014. On the other hand, SDL has changed the word count algorithms along the years (see \url{https://multifarious.filkin.com/2012/11/13/wordcount/}), so it is unclear what “Trados 2007-like” option involves in MemoQ. It is however relevant in the sense that Kilgray acknowledges that differences in word counts impact the final quality score.}

The statistics part shows a table with categories (rows), severities (columns) and the penalty points (values). Totals include a breakdown of penalties per category and severity. The bottom row shows normalised scores. The error list (Figure 43) completes the report with a list of issues that include segment number, source and target segments, corrections, error types and comments.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig41.png}
\caption{Project-based reports in MemoQ.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig42.png}
\caption{Quality report in MemoQ.}
\end{figure}
SDL Trados Studio

SDL Trados Studio implements the possibility of generating TQA reports as part of the TQA process. The report generation is an optional action manually triggered by the user, or it can be included as a default step in a batch task sequence inside a translation project.

SDL Trados Studio reports are XML-based documents that gather all the information related to the TQA exercise inside a project. They include: a summary of the TQA with the project’s name, date, number of files and the “Project Pass Result”, which can be “Pass” or “Fail”; the metric used, including the list of issues shown with its hierarchy, the severities and the values associated to each issue/severity; the number of occurrences of each error type; a section with more data about the documents revised (file names, result per document and penalty/points result, general evaluation comments and document type).
Figure 45. TQA report in SDL Trados Studio.

Below, reports display a table with detailed annotations: source segment, original translation, revised translation, type of issue and severity (Figure 46).

Figure 46. Report’s annotations in SDL Trados Studio.

translate5

In translate5, MQM statistics can be displayed at any moment by clicking on the “MQM” button on the toolbar. The report shows only the issues type present in the document with the number of occurrences (Figure 47).
However, the interface does not allow generating more complete reports. If export options are used, XML reports can be generated but they contain the same information as the MQM statistics shown above.
**XTM Cloud**

XTM Cloud’s implementation offers the possibility of generating and downloading a statistical report in Microsoft Excel format. The report includes all issue types defined globally under the user account settings and their occurrences in the project, as illustrated in Figure 48. The report also shows the penalty points in two columns, one called “Raw” that shows the total points per error type, and another one called “Adj.” that shows the points after adjustments, that is, after applying the issues weights.

![Figure 48. XTM Cloud’s quality report.](image)

**Conclusion**

<table>
<thead>
<tr>
<th>Product</th>
<th>Score</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluency Now</td>
<td>½</td>
<td>The report includes statistics but not marks.</td>
</tr>
<tr>
<td>MemoQ</td>
<td>1</td>
<td>The report includes statistics and text marks.</td>
</tr>
<tr>
<td>SDL Trados Studio</td>
<td>1</td>
<td>The report includes statistics and text marks.</td>
</tr>
<tr>
<td>translate5</td>
<td>0</td>
<td>The report includes statistics, but not quality scores.</td>
</tr>
<tr>
<td>XTM Cloud</td>
<td>½</td>
<td>The report includes statistics but not marks.</td>
</tr>
</tbody>
</table>

*Table 13. Report generation summary.*

**4.2.1.12 Score calculation**

**Feature overview**

This item refers to the software’s capacity of producing an overall quality value based on the issues marked, their severity penalties and the issue weights
according to a given formula. The software may also include the possibility to define a pass mark or quality threshold, so that the organisation can enforce minimum quality targets for the translation to be acceptable in terms of quality.

Evaluation criteria

This item is demonstrated when the software automatically computes a quality score considering all issues in the document together with the severity penalties and the issue weights. A rating of 1 is given to the solutions that provide an overall quality score including the possibility of defining a pass/fail threshold. A rating of \( \frac{1}{2} \) is given to the solutions that compute an overall quality score, but do not provide a pass/fail threshold. A rating of 0 is given to the implementations not offering any quality score.

Fluency Now

This implementation follows MQM’s principle to calculate an overall quality score based on its formula \( TQ = 100 - TP + SP \), that is: translation quality equals 100 (maximum quality score) minus translation penalties plus source penalties. The overall quality score is shown in the MQM pane as well as in the quality report.

Fluency Now does not provide a pass/fail mark calculation.

MemoQ

MemoQ calculates an overall quality score and provides some additional options. To calculate the quality score, the tool considers the value 1 as the maximum quality achievable for the volume in words. Then, it subtracts to that maximum value the penalty points cumulated in the document, producing a proportional final value that adopts the format of \( 0.##### \) (for instance, 0.9826, as

---

36 MQM provides a scoring algorithm for computing the overall score \( TQ = 100 - TP + SP \), in which the \( TP \) (target penalties) and the \( SP \) (source penalties) calculation includes the volume in words. Because this formula considers source and target issues, issue weights and word counts, the comparison among tools in terms of numeric results is not contemplated here. Therefore, the only requirement searched is that the overall quality score reflects all issues, severities and weights; tools’ algorithms to obtain such results are not evaluated.
shown in the report section). For example, for a translation of 10,000 words with 1,000 cumulated penalty points, the result is 0.9500 \((1 - (500 / 10,000))\).

To define the pass/fail mark, MemoQ offers two options. First, the default setting is based on an absolute quality mark that may range from 0.00 to 0.99. Alternatively, the quality assessment fails if the number of penalty points is higher than a given number of words (for instance, more than 100 penalty points per 1000 words would result in a fail). Besides, MemoQ gives the option to the user to count multiple occurrences of the same error as one, which prevents certain types of errors (e.g. same wrong term) to impact excessively the final quality score.

![Figure 49. MemoQ's Pass/Fail criteria.](image)

**SDL Trados Studio**

Trados Studio TQA implementation computes the overall quality result as a cumulative value of penalty points. That is, the tool does not offer a normalised quality score (for instance, up to 100 or 1.0), but a simple sum of total penalty points. In this sense, the tool’s approach is oriented towards pass/fail scenarios where the relative measure of penalty points can be compared to a given number of words (threshold). The default settings show 50 penalty points per 1000 words as a quality threshold to determine a pass/fail result, but this can be customised per metric (Figure 50). To give a practical example of the calculation, a document of 10,000 words would have 500 penalty points allowed \(((10,000 / 1000) * 50 = 500)\); then, if total penalty points sum is higher than 500 points, the result is a fail.
By default, translate5 does not offer any quality score calculation in the interface nor in the MQM statistics that are generated.

**XTM Cloud**

This implementation provides the user with quality scores exclusively in the quality report, which shows subscores based on issue types and a total normalised quality score (up to 100%). The calculation is based on the total word count and penalty points after adjustments (issue weighting), being the total quality the percentage of the total penalty points. For example, for a translation of 10000 words and 500 penalty points, the overall quality would be 95% (100 - (500 / 10000) = 95). XTM Cloud does not produce a pass/fail result.

**Conclusion**

<table>
<thead>
<tr>
<th>Product</th>
<th>Score</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluency Now</td>
<td>½</td>
<td>An overall quality score is calculated without pass/fail result.</td>
</tr>
<tr>
<td>MemoQ</td>
<td>1</td>
<td>An overall quality score is calculated with pass/fail result.</td>
</tr>
<tr>
<td>SDL Trados Studio</td>
<td>1</td>
<td>An overall quality score is calculated with pass/fail result.</td>
</tr>
<tr>
<td>translate5</td>
<td>0</td>
<td>An overall quality score is not calculated.</td>
</tr>
<tr>
<td>XTM Cloud</td>
<td>½</td>
<td>An overall quality score is calculated without pass/fail result.</td>
</tr>
</tbody>
</table>

*Table 14. Score calculation summary.*
4.2.2 TQA interoperability

4.2.2.1 Intermediate format

Feature overview

This item refers to the capacity of the CAT tool to store, for translation, revision and assessment purposes, the source and target texts together with the TQA annotations in a format that is open, accessible and based on localisation standards, so that it can be shared, consumed and reused beyond the client local scope by other users or in different systems without the need of additional parsing or conversion. An intermediate format —also referred to as a ‘working format’ or ‘bilingual format’— that is based on localisation standards is XLIFF.

Evaluation criteria

This item is proved when the software solution implements an intermediate format that complies with a localisation standard such as XLIFF, so that files can be exchanged among different systems that support the same standard. A rating of 1 is given to the products that have a localisation standard as an intermediate format that relies exclusively on standard specifications and namespaces. A rating of ½ is given to the tools that use a customised localisation standard that complies only partially with the standard (for example, adding its own namespaces). A rating of 0 is given to the tools that do not use a localisation standard or the format used is not open to the user.

Fluency Now

As we have seen in 4.2.1.9, Fluency Now implements a proprietary XML vocabulary (FTFX) that embeds RTF text. Since RTF marks are more permissive than XML rules, markup is more flexible and accept, for example, overlapping spans or longer flagged fragments. On the other hand, although it is an open and standard-based format, it remains a highly complex and hybrid format that cannot be easily processed by other tools.

MemoQ

MemoQ’s intermediate format is a proprietary format with the extension .dat file. Although MemoQ keeps an XLIFF file with the original text in the
project folder, the translations are stored in the .dat file. A .dat file is a generic serialisation format but the data structure is determined by the programme that uses it, so that it cannot be consumed by any other tool without a previous conversion or deserialisation.

**SDL Trados Studio**

In SDL Trados Studio, the annotated files are saved as SDLXLIFF, a proprietary XLIFF flavour, which is a customised version of XLIFF 1.2 extended by SDL own namespace. Although SDLXLIFF remains an XML-compliant and standard-based format, it is not necessarily supported—or at least not fully—by the tools that otherwise support XLIFF 1.2.

**translate5**

translate5 does not use an intermediate format, but a MySQL database to store the source, the translation and the annotations. translate5 does not offer a way to access this data in a raw format from the interface.

**XTM Cloud**

Being XTM a proprietary cloud solution, the access to the intermediate file is not permitted. Also, as in the case of translate5, the interface does not provide any way to interact with the files and data stored on the server side.

**Conclusion**

<table>
<thead>
<tr>
<th>Product</th>
<th>Score</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flueny Now</td>
<td>0</td>
<td>The file format used is not a localisation standard.</td>
</tr>
<tr>
<td>MemoQ</td>
<td>0</td>
<td>The file format used is not a localisation standard.</td>
</tr>
<tr>
<td>SDL Trados Studio</td>
<td>½</td>
<td>The file format used is based on the standard XLIFF 1.2.</td>
</tr>
<tr>
<td>translate5</td>
<td>0</td>
<td>The raw data is not accessible.</td>
</tr>
<tr>
<td>XTM Cloud</td>
<td>0</td>
<td>The raw data is not accessible.</td>
</tr>
</tbody>
</table>

*Table 15. Intermediate format summary.*

### 4.2.2.2 Markup

**Feature overview**

This item refers to the capacity of the tool to mark the quality issues and store the annotations in the intermediate format according to the standards, such
as XLIFF 1.2\textsuperscript{37}, XLIFF 2.0\textsuperscript{38}, Internationalization Tag Set 2.0 (ITS)\textsuperscript{39} or MQM\textsuperscript{40}. Conformance to these specifications makes possible the interpretation of TQA annotations by tools and systems that also support such standards.

**Evaluation criteria**

This item is demonstrated when the tool complies with any of the standard specifications related to quality issues or assessment annotations markup. A rating of 1 is given to the tools that conform to any standard specification in their intermediate format. A rating of ½ is given to the tools that make use of standard elements but under some degree of customisation or extension (for example, using their own namespaces). A rating of 0 is given to the tools that do not comply with any standard, or to those whose intermediate format cannot be examined because it is not accessible or because it does not use an open format.

**Fluency Now**

TQA markup in Fluency Now is implemented through RTF-based formatting (see sample below). Although this format is an open format and the encoded data can be parsed by other tools, it does not follow the standards in localisation, such as XLIFF or ITS.

\begin{verbatim}
<SourceText>Valentino prefiere la elegancia a la notoriedad</SourceText>
<TargetRTF>{\rtf1\ansi\deflang3082\ftnbj\uc1\deff0
\{\fonttbl{\f0 \fn1 Arial;}{\f1 \fnnil \fs24 \fcbarh\ch\l0 Arial;}\}
{\color.tbl{\red255\green255\blue255 ;}\red0\green0\blue0 ;}\red0 \green0\blue0 ;}
{\stylesheet{\fs24 Normal;}{\cs1 Default Paragraph Font;}}{\revtbl{Unknown;}}
\paperw12240\parwh15840\marginl0\marginr0\marginb0\headery720\footery720
\footnoteskip0\footnotewidestrut
\margshade\fet4\aendnotes\aftnmc|\pgbrdr|\pgbrdrfoot\sectd|\pgwson12240|\pghsxn15840|\guttersxn0|\marglxsxn1800|\marginrxsxn1800|\margtxsn1440|\margbxsxn1440|\headery720|\footery720|\sbkpage|\pgndec\ps \fs24\pard\pl
\f1\fs24\par\pl\f1\fs24\bfn
Valentino prefers the elegance to the notoriety\par }\</TargetRTF>
\end{verbatim}

\textsuperscript{37} Although XLIFF 1.2 does not provide a specific element for quality issues or assessment annotations, it does provide a mechanism for annotations in general, including QA: http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html#mrk.

\textsuperscript{38} http://docs.oasis-open.org/xliff/xliff-core/v2.0/os/xliff-core-v2.0-os.html#annotations.

\textsuperscript{39} https://www.w3.org/TR/its20/#lqissue.

\textsuperscript{40} http://www.qt21.eu/mqm-definition/definition-2015-12-30.html#mqm_markup.
MemoQ

Markup in MemoQ is encoded in a proprietary format (extension .dat). It is therefore not possible to analyse its markup mechanism. It must be also concluded that, unless Kilgray offers an application programming interface (API) that is able to handle and process this file format for TQA and interoperability purposes, markup cannot be consumed by any other system or implementation out of the box.

SDL Trados Studio

The markup of quality issues is based on the <mrk> element defined in the XLIFF 1.2 specification, which comprises also the required attribute mtype.\(^{41}\) It includes a reference as well to a proprietary SDL namespace.\(^{42}\) Although other tools may parse correctly the <mrk> element, the assumption made here is that it is not guaranteed that they will interpret correctly SDL vocabulary, ignoring the TQA markup as a result. This can produce unwanted outcomes, as illustrated in the example below. In the following block of an SDLXLIFF file, a part of the sentence is deleted. However, in the event the <mrk> elements are ignored, the deleted parts may appear in the document as normal text.

```xml
<trans-unit id="fb0a8351-f977-4b61-ada8-2b3d423bc880">
  <source>Valentino prefiere la elegancia a la notoriedad</source>
  <seg-source>
    <mrk mtype="seg" mid="1">Valentino prefiere la elegancia a la notoriedad</mrk>
  </seg-source>
  <target>
    <mrk mtype="seg" mid="1">Valentino prefers <mrk mtype="x-sdl-feedback-deleted" sdl:revid="f8d35c29-ac41-42c2-a9fc-e44c17dbe268">the</mrk>elegance to <mrk mtype="x-sdl-feedback-deleted" sdl:revid="ba7be0f3-bf6b-4b07-b649-12576f889cca">the</mrk>fame</mrk>
  </target>
  <sdl:seg-defs>
    <sdl:seg id="1" conf="Draft" origin="interactive">
      <sdl:value key="SegmentIdentityHash">OxubqkUotplHnRKcMO91Huos8lk=</sdl:value>
      <sdl:value key="created_by"></sdl:value>
      <sdl:value key="created_on"></sdl:value>
      <sdl:value key="last_modified_by"></sdl:value>
      <sdl:value key="modified_on"></sdl:value>
    </sdl:seg>
  </sdl:seg-defs>
</trans-unit>
```

\(^{41}\) See [http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html#mrk](http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html#mrk).

\(^{42}\) See [http://sdl.com/FileTypes/SdlXliff/1.0](http://sdl.com/FileTypes/SdlXliff/1.0).
Since translate5 does not provide access to its intermediate format, it is not possible to evaluate the TQA markup applied to the bilingual document.

**XTM Cloud**

As in the case of translate5, it is not possible to evaluate the markup syntax used in the XTM’s intermediate format.

### Conclusion

<table>
<thead>
<tr>
<th>Product</th>
<th>Score</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluency Now</td>
<td>0</td>
<td>The markup used is not compliant with localisation standards.</td>
</tr>
<tr>
<td>MemoQ</td>
<td>0</td>
<td>The markup used cannot be analysed (format not open/accessible).</td>
</tr>
<tr>
<td>SDL Trados Studio</td>
<td>½</td>
<td>The markup used is partially compliant with XLIFF 1.2.</td>
</tr>
<tr>
<td>translate5</td>
<td>0</td>
<td>The markup used cannot be analysed (format not open/accessible).</td>
</tr>
<tr>
<td>XTM Cloud</td>
<td>0</td>
<td>The markup used cannot be analysed (format not open/accessible).</td>
</tr>
</tbody>
</table>

*Table 16. Markup summary.*

### 4.2.2.3 Report compatibility

**Feature overview**

This item refers to the capacity of generating reports in a format that allows the information from the assessment to be interpreted and displayed by other systems without the need of additional processors or the same tool that generated the report. Additionally, compatible reports should include a metric description following an existing standard (e.g. MQM markup\(^{43}\)) in a way that other systems can also consume and reuse the same metric without further configuration or processing.

**Evaluation criteria**

This item is considered proven when the tool can generate complete TQA reports in an open format. A rating of 1 is given to the solutions that generate

---

complete reports in an open format, including quality issues and the metric description. A rating of \( \frac{1}{2} \) is for the solutions that generate reports in an open format, but neglect or omit the exchange of TQA additional information such as the metric description, the list of issues or the quality scores. A rating of 0 is given to the tools that generate reports in a proprietary format and do not include full details in the TQA results.

**Fluency Now**

Reports generated by Fluency Now are simple plain text files with a basic layout, as shown in 4.2.1.11. Therefore, in terms of exchangeability, the reports can be shared with other stakeholders, although exploiting the report data in an automated manner seems far from trivial. The reports also lack details on the quality metric used or the issues flagged.

**MemoQ**

MemoQ allows exporting quality reports in Hypertext Markup Language (HTML) and in Comma-Separated Values (CSV), as shown in Figure 52. The reports do not include a full description of the metric used.

![Figure 52. Exporting reports in MemoQ.](image)

**SDL Trados Studio**

Reports are natively saved as XML inside the “Reports” folder of the project directory in the local file system. The XML vocabulary used for the reports is proprietary, which means that users with different systems may encounter
difficulties in processing these reports, although interoperability is effectively empowered thanks of the use of XML.\textsuperscript{44}

It must be mentioned as well that the markup used for the reported issues is not standard-based, as illustrated in the following sample. For interchange and interoperability purposes, normative MQM markup could have been used instead.

\begin{verbatim}
<revisedTranslation>
  <group>
    <item content="Valentino prefers " type="" />
  </group>
  <group author="XXX" dateTime="30/07/2017 13:46:20" comment=""
    category="Accuracy - Addition" severity="Minor">
    <item content="the " type="FeedbackDeleted" />
  </group>
  <group>
    <item content="elegance to " type="" />
  </group>
  <group author="XXX" dateTime="30/07/2017 13:46:28" comment=""
    category="Accuracy - Addition" severity="Minor">
    <item content="the " type="FeedbackDeleted" />
  </group>
  <group>
    <item content="fame" type="" />
  </group>
</revisedTranslation>
\end{verbatim}

SDL Trados Studio also offers the possibility of exporting the reports as HTML, or MIME Encapsulation of Aggregate HTML Documents (MHT), which may help other users interpreting the TQA results and other systems processing and archiving the data.

Finally, Trados Studio is able to produce complete reports, including a description of the quality metric used, as well as a list of reported issues.

\textit{translate5}

As we have seen in 4.2.1.11, translate5 does not produce complete TQA reports. However, the statistical reports exported from the system are saved as XML files. Therefore, translate5 reports can theoretically be exploited by other systems, although this would require additional processing.

\footnote{\textsuperscript{44} It must be noted that, even though raw XML data is not a friendly format to handle for the non-technical user, it represents an ideal way to enforce interoperability, automation and archiving needs. Regarding the display, Extensible Stylesheet Language Transformations (XSLT) can help customising a visual representation of the XML data better than rather volatile formats such as CSV or HTML.}
**XTM Cloud**

XTM Cloud gives the option to export the quality reports only as Microsoft Excel files, that is, a proprietary format. Although it is not impossible that other systems process or automate the TQA information, it is not the ideal situation in terms of data interchange. Also, some TQA details are missing, like the quality issues flagged in the text.

**Conclusion**

<table>
<thead>
<tr>
<th>Product</th>
<th>Score</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluency Now</td>
<td>$\frac{1}{2}$</td>
<td>Reports based on an open format, but lack TQA information.</td>
</tr>
<tr>
<td>MemoQ</td>
<td>$\frac{1}{2}$</td>
<td>Reports based on an open format, but lack TQA information.</td>
</tr>
<tr>
<td>SDL Trados Studio</td>
<td>1</td>
<td>Reports based on an open format, including complete TQA details.</td>
</tr>
<tr>
<td>translate5</td>
<td>$\frac{1}{2}$</td>
<td>Reports based on an open format, but lack TQA information.</td>
</tr>
<tr>
<td>XTM Cloud</td>
<td>0</td>
<td>Reports use a proprietary format and lack TQA information.</td>
</tr>
</tbody>
</table>

*Table 17. Report compatibility summary.*

**4.2.2.4 Import/export**

**Feature overview**

This item refers to the capacity of the tool to import and export data related to the TQA, including a metric definition and annotated files in a standard format. By including the option to export and import a metric definition in a standard way, other users and systems can apply the same quality model without additional processing. By giving the possibility to export bilingual files in a standard format, other users can further process the assessed documents.

**Evaluation criteria**

This item is demonstrated when the tool offers a mechanism to export the quality metric and the annotated files in a standard format. A rating of 1 is given to the tools that offer the possibility to export quality models and annotated files in a localisation standard format. A rating of $\frac{1}{2}$ is given to the tools that can export and import the metric definition and the annotated files, but do not totally follow any localisation standard. A rating of 0 is for the tools that do not offer the possibility of exporting/importing quality models or annotated files.
Fluency Now

Fluency Now is able to load metric description files following MQM normative markup, but it does not allow to export them. In fact, since the tool does not offer a way to modify the metric inside the tool, the export option would be useless. Therefore, the approach taken by Western Standard is that all customisation and handling of the metric must be done outside the tool, which is only designed to load and apply standard metrics. In that sense, Fluency Now is closer to the standard because the description of the metric can be exchanged with other systems that follow the MQM description rules, not imposing its own description model.

When it comes to exporting annotated documents, Fluency Now also fails to deliver a satisfactory functionality in terms of interoperability. Even if the tool has the option to export bilingual contents to XLiff, the XLiff is generated on the fly with source and target contents, but without annotations. The bilingual file generated is a clean XLiff 1.2 that shows the state of source and target segments at the moment of the export, but without any metadata.

MemoQ

MemoQ allows exporting the metric definition using its own proprietary XML format (.mqres), which the tool defines as an LQA-type resource file. This resource file can be then imported in other MemoQ projects and instances, which encourages the use of a single metric along the chain provided that the tool is compatible with such format. Nevertheless, although not an ideal situation, being .mqres an XML-based file, other systems could also parse such resources to adapt and import metrics defined in MemoQ.

Concerning the intermediate format, MemoQ provides the user with the ability of exporting the bilingual document as a MQXLiff, which is a custom XML vocabulary based on version XLiff 1.2. This format makes use of `<mrk>` elements to mark up the quality issues, which are consistent with the XLiff standard, although they are also extended with MemoQ's namespace properties, as illustrated in the following sample.

```
<mrk mtype="x-mq-range" mtype="start" mq:ownerid="5b5de469-2002-40f3-b41d-7cb8cf823d92" />
<mrk mtype="x-mq-range" mtype="end" mq:ownerid="5b5de469-2002-40f3-b41d-7cb8cf823d92" />
```
As in the case of SDL Trados Studio, `<mark>` elements can be processed or simply ignored by other tools, which may have consequences in the way the text is displayed or interpreted by these other tools.

**SDL Trados Studio**

SDL Trados Studio offers the possibility of exporting the definition of any custom metric as a proprietary, XML-based file format under the extension of `.sdltqasettings`. This facilitates importing the metric in other projects, or in the same project but in different workstations, provided that the agent remains SDL Trados Studio 2015 or higher.

Regarding the bilingual document where the annotations are made, although it is XLIFF-based, it remains a custom format (SDLXLIFF). No options are given to export it as a standard XLIFF.

**translate5**

`translate5`’s approach is similar to Fluency Now in that both tools seem to consider that metrics must be handled and modified outside the tool. However, `translate5` uses a custom description of the quality model, which hinders the exchange of the quality models with other systems.

Regarding the export of annotated files, the tool only offers the possibility of exporting the bilingual files in a clean version and in a differential version, but none of them include the quality annotations.

**XTM Cloud**

XTM Cloud does not offer any way to export the quality metric as configured in the tool. If the same metric must be used by another user, it is assumed that the metric must be reconfigured manually, even if using the same system or the same project.

With respect to the bilingual files, XTM Cloud allows to export the bilingual files as XLIFF, but without the TQA flags.
Conclusion

<table>
<thead>
<tr>
<th>Product</th>
<th>Score</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluency Now</td>
<td>½</td>
<td>Export/import according to standards partially supported.(^{45})</td>
</tr>
<tr>
<td>MemoQ</td>
<td>½</td>
<td>Export/import supported but not standard-compliant.</td>
</tr>
<tr>
<td>SDL Trados Studio</td>
<td>½</td>
<td>Export/import supported but not standard-compliant.</td>
</tr>
<tr>
<td>translate5</td>
<td>0</td>
<td>Export of metric description and annotated files not supported.</td>
</tr>
<tr>
<td>XTM Cloud</td>
<td>0</td>
<td>Export of metric description and annotated files not supported.</td>
</tr>
</tbody>
</table>

Table 18. Import/export summary.

4.2.2.5 MQM compliance

Feature overview

This item refers to the compliance of the tool regarding the normative aspects of the MQM framework, contributing to the exchange of TQA data among different systems and stakeholders along the translation cycle. The normative aspects of the MQM are: terms and definitions\(^{46}\), conformance\(^{47}\), issue types\(^{48}\), markup\(^{49}\), relationship to ITS 2.0\(^{50}\).

Evaluation criteria

This item is demonstrated when the tool follows and respects the normative aspects of MQM. A rating of 1 is given to the tools that comply with all normative aspects of MQM; a rating of ½ is given to the tools that comply only with certain aspects of MQM; a rating of 0 is for the tools that do not comply with any normative aspect of MQM definition.

\(^{45}\) Although Fluency Now does not support export/import of the metric, a ½ point is granted because of the advantage of being the only tool using MQM normative markup for metric description.


\(^{47}\) http://www.qt21.eu/mqm-definition/definition-2015-12-30.html#conformance. Also, with respect to “Conformance”, note that the only requirement is the use of the MQM vocabulary of issue types.


Fluency Now

Fluency Now complies with certain aspects of the MQM definition, namely: conformance, issue types and markup (for the description of the metric). Other aspects, such as terms and definitions or ITS 2.0 relationship are not explicitly followed or respected.

MemoQ

MemoQ does not comply with any normative aspect of the MQM, although it is possible to create an issue typology following the same vocabulary. However, as this model is not offered by the tool, we consider that the tool itself is not MQM-compliant.

SDL Trados Studio

SDL Trados Studio includes an MQM model that is based on the MQM typology. Therefore, at least in terms of conformance and issue types, the tool complies with the MQM. However, other MQM aspects are not respected.

translate5

translate5 complies with MQM aspects like conformance and issue types, but it is unclear—or cannot be stated—to what extent other aspects are respected, as seen in previous analysis items, mainly due to the fact that markup files are not accessible on the server.

XTM Cloud

This tool uses elements from the MQM definition, as the issue vocabulary—also explicitly present in the issue details form—, although it is not clear from the interface that the model used is MQM. Nevertheless, even if there seems to be certain confusion between TAUS DQF and MQM in the documentation, we can consider that at least conformance and issue vocabulary as respected.
Conclusion

<table>
<thead>
<tr>
<th>Product</th>
<th>Score</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluency Now</td>
<td>½</td>
<td>Conformance, issue vocabulary and markup (metric) are respected.</td>
</tr>
<tr>
<td>MemoQ</td>
<td>0</td>
<td>Explicit support to MQM is not included in the default installation.</td>
</tr>
<tr>
<td>SDL Trados Studio</td>
<td>½</td>
<td>Conformance and issue vocabulary are respected.</td>
</tr>
<tr>
<td>translate5</td>
<td>½</td>
<td>Conformance and issue vocabulary are respected.</td>
</tr>
<tr>
<td>XTM Cloud</td>
<td>½</td>
<td>Conformance and issue vocabulary are respected.</td>
</tr>
</tbody>
</table>

Table 19. MQM compliance summary.

4.3 Summary

In the following table, all products analysed here are shown with the corresponding evaluation described in the previous chapters. The values must be interpreted as indicated in the respective sections designed as ‘Evaluation criteria’. Nevertheless, for the sake of simplicity or in case of doubt, the marks should be read as follows:

1. Full support (1 point): the feature or aspect analysed is implemented, compliant or existent, or it can be performed as defined.
2. Partial support (½ point): the feature or aspect analysed is only implemented partially or it is not fully compliant with the model.
3. Not supported (0 points): the feature or aspect analysed is not implemented, it is not compliant or it cannot be performed.
Table 20. TQA in CAT tools summary.

<table>
<thead>
<tr>
<th>Function</th>
<th>Fluency Now</th>
<th>MemoQ</th>
<th>SDL Trados Studio</th>
<th>translate5</th>
<th>XTM Cloud</th>
<th>Subtotals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Built-in quality models</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>½</td>
<td>½</td>
<td>3.0</td>
</tr>
<tr>
<td>Customising a metric</td>
<td>½</td>
<td>1</td>
<td>½</td>
<td>½</td>
<td>1</td>
<td>4.0</td>
</tr>
<tr>
<td>Import/export</td>
<td>½</td>
<td>½</td>
<td>½</td>
<td>0</td>
<td>0</td>
<td>1.5</td>
</tr>
<tr>
<td>Intermediate format</td>
<td>0</td>
<td>0</td>
<td>½</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>Issue flagging</td>
<td>1</td>
<td>½</td>
<td>½</td>
<td>½</td>
<td>0</td>
<td>2.5</td>
</tr>
<tr>
<td>Issue hierarchy</td>
<td>1</td>
<td>½</td>
<td>½</td>
<td>1</td>
<td>1</td>
<td>4.0</td>
</tr>
<tr>
<td>Issue weighting</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>½</td>
<td>1</td>
<td>4.5</td>
</tr>
<tr>
<td>Markup</td>
<td>0</td>
<td>0</td>
<td>½</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>Markup scope</td>
<td>1</td>
<td>½</td>
<td>½</td>
<td>½</td>
<td>0</td>
<td>2.5</td>
</tr>
<tr>
<td>MQM compliance</td>
<td>½</td>
<td>0</td>
<td>½</td>
<td>½</td>
<td>½</td>
<td>2.0</td>
</tr>
<tr>
<td>Overlapping spans</td>
<td>½</td>
<td>½</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2.0</td>
</tr>
<tr>
<td>Report generation</td>
<td>½</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>½</td>
<td>3.0</td>
</tr>
<tr>
<td>Report compatibility</td>
<td>½</td>
<td>½</td>
<td>1</td>
<td>½</td>
<td>0</td>
<td>2.5</td>
</tr>
<tr>
<td>Score calculation</td>
<td>½</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>½</td>
<td>3.0</td>
</tr>
<tr>
<td>Severity levels</td>
<td>½</td>
<td>1</td>
<td>1</td>
<td>½</td>
<td>½</td>
<td>3.5</td>
</tr>
<tr>
<td>Split issues</td>
<td>0</td>
<td>½</td>
<td>½</td>
<td>0</td>
<td>0</td>
<td>1.0</td>
</tr>
<tr>
<td>Totalsa</td>
<td>8.0 (50%)</td>
<td>9.5 (59%)</td>
<td>11.0 (69%)</td>
<td>6.0 (38%)</td>
<td>5.5 (34%)</td>
<td>40.00 (50%)</td>
</tr>
</tbody>
</table>

Note: grey background indicates interoperability items.
Table 21. TQA implementation and interoperability in CAT tools.

<table>
<thead>
<tr>
<th></th>
<th>Fluency Now</th>
<th>MemoQ</th>
<th>SDL Trados Studio</th>
<th>translate5</th>
<th>XTM Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation $^b$</td>
<td>6.50</td>
<td>8.50</td>
<td>8.00</td>
<td>5.00</td>
<td>5.00</td>
</tr>
<tr>
<td>Interoperability $^c$</td>
<td>1.50</td>
<td>1.00</td>
<td>3.00</td>
<td>1.00</td>
<td>0.50</td>
</tr>
<tr>
<td>Totals $^d$</td>
<td>8.00</td>
<td>9.50</td>
<td>11.00</td>
<td>6.00</td>
<td>5.50</td>
</tr>
</tbody>
</table>

$^a$ Out of 16 total points.

$^b$ Out of 11 total points.

$^c$ Out of 5 total points.

$^d$ Out of 16 total points.
Table 22. TQA in CAT tools according to application type.

<table>
<thead>
<tr>
<th></th>
<th>Desktop(^e)</th>
<th>Cloud(^f)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation(^g)</td>
<td>7.67</td>
<td>5.00</td>
</tr>
<tr>
<td>Interoperability(^h)</td>
<td>1.83</td>
<td>0.75</td>
</tr>
<tr>
<td>Totals(^i)</td>
<td>9.50</td>
<td>5.75</td>
</tr>
</tbody>
</table>

\(^e\) Average points (Fluency Now, MemoQ and SDL Trados Studio).

\(^f\) Average points (translate5, XTM Cloud).

\(^g\) Out of 11 total points.

\(^h\) Out of 5 total points.

\(^i\) Out of 16 total points.
Table 23. TQA in CAT tools according to business model.

<table>
<thead>
<tr>
<th></th>
<th>Perpetual</th>
<th>Subscription</th>
<th>Free (Open-source)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation</td>
<td>8.25</td>
<td>5.75</td>
<td>5.00</td>
</tr>
<tr>
<td>Interoperability</td>
<td>2.00</td>
<td>1.25</td>
<td>1.00</td>
</tr>
<tr>
<td>Total</td>
<td>10.25</td>
<td>7.00</td>
<td>6.00</td>
</tr>
</tbody>
</table>

\(^j\) Average points (MemoQ, SDL Trados Studio).

\(^k\) Average points (Fluency Now, XTM Cloud).

\(^l\) Total points (translate5).

\(^m\) Out of 11 total points.

\(^n\) Out of 5 total points.

\(^o\) Out of 16 total points.
Chapter 5: Discussion

5.1 Introduction

The interest in normalised assessment approaches seems to have led to an equal interest for the translation technologies sector to implement solutions that support evaluation tasks according to TQA principles (see 1.1). Considering TQA key features, several questions were raised about its practical implementation in CAT tools. First, are tool makers providing adequate instruments for companies and end-users in their CAT tools to carry out TQA activities regardless their quality framework and purpose (training, quality assurance, quality feedback, machine translation evaluation)? And second, do these tools provide mechanisms to allow translation stakeholders working with different systems to exchange TQA data without the need of further conversion and processing? Based on the reference framework chosen, MQM (Lommel et al. 2015), the assumption made was that software companies that claim to offer TQA features in their products should have integrated in them a solid support to TQA characteristics such as custom issue hierarchies, issue flagging mechanisms, quality scores and report generation, among others. Regarding the second question about the exchangeability of the assessment-related data, we considered that TQA implementations should be designed so that they correct or mitigate what Harris et al. (2016) have defined as a highly fragmented landscape of quality evaluation solutions. Our expectations were, therefore, that TQA implementations could offer a way to handle assessment data beyond the tool itself so that all stakeholders in the translation cycle (clients, vendors, linguists or students) can benefit from the TQA effort.

5.2 TQA in CAT tools or a TQA for each CAT tool

The first idea that comes up from the analysis data is that there is not a unique answer to how CAT tools implement quality assessment nowadays. Leaving aside the fact that still many tools do not offer any TQA support out-of-the-box, those that do so tend to show significant differences among them. This is noticeable already in the manner they refer to TQA: from “Linguistic Quality Assurance” (Kilgray) to “Translation Quality assessment” (SDL) all the way up to the absence of a generic label (Western Standard). However, the various
interpretations made by tool makers and crystallised in their tools also concern other more transcendental aspects of TQA.

As we have seen, the functions that enable the definition of issue hierarchies are implemented very differently from one tool to another. For example, XTM Cloud, translate5 or Fluency Now tend to offer higher freedom when it comes to creating or loading complex typologies: they allow multiple levels and more flexible category definitions. On the other hand, they partially fail in their application —XTM in particular— due to a simplistic issue marking mechanism that focuses mainly on the marks or the score, falling short in the annotation details, and therefore neglecting an important part of the actual TQA benefits, that is, contributing “to improve translation output, i.e., prevent those issues from recurring in the future” (Burchardt, Harris, et al. 2016; Harris et al. 2016, p.50). Other tools such as SDL Trados Studio and MemoQ, on the other hand, show a more rigid approach to the definition of issue typologies with fewer category levels and lower granularity, but they offer in exchange richer annotations, so that not only statistics but also complete linguistic feedback can be exploited by users in both bilingual files and quality reports, so that translation output can be improved in the future.

Different approaches have also been observed with respect to penalty mechanisms and the quality score calculation. Although all tools —except translate5— can generate an overall quality score out-of-the-box, each tool has a unique way to get the final quality results. Differences have been identified at multiple levels: from the way issues are weighted to the algorithms used to compute the final marks. Also, while there are tools —Fluency Now, MemoQ— that generate normalised marks (1.000, or 100% as the maximum quality achievable), other tools are more oriented to relative pass/fail thresholds, in particular, SDL Trados Studio, whose overall score cannot be understood in absolute terms, but only in relation to a predefined and relative threshold.

In general, although there is nothing intrinsically wrong with the different scoring methodologies applied (all produce an overall quality score, indeed), it is especially striking the lack of uniformity and standardisation. To give an example, depending on the tool, the same TQA marks would produce a 0.8736 (MemoQ),
an 87.36% (Fluency Now, XTM Cloud) or a 244/1074\(^{66}\) (SDL Trados Studio). Regrettably, even if all of them would use the same normalised maximum value, each one would still generate a different result, due to the different weighting mechanisms and also because of the different word count algorithms. Yet, if tools conformed to the same format (for instance, a 0 to 100% score, as other evaluation methods do), even with slight variations among them, users would have a better sense of the quality achieved, and the correlation among tools would also be facilitated.

The mixed picture that TQA implementations represent should not necessarily be detrimental to the achievement of TQA goals in each one of the solutions. In other words, different TQA design approaches do not imply an advantage or disadvantage per se. As we have seen, marking and measuring issues can be carried out in various manners, provided that they allow characterising as objectively as possible whether a translation “complies with quality specifications” (Lommel et al. 2015). On the other hand, we cannot deny that certain designs fulfil TQA requirements better than others, as reflected in the different marks given in this study, in particular based on the idea that TQA should provide complete and open quality feedback, thus contributing to the translation improvement process (Burchardt, Harris, et al. 2016; Harris et al. 2016). This brings us to a refinement of our first question: that is, whether existing TQA software offerings show a satisfactory implementation regardless the different approaches.

5.3 TQA implementations: still room for improvement

As it has been set up, a successful and complete TQA implementation should require the fulfilment of predefined criteria, failing at which TQA cannot be performed, or not in an ideal way. In actual terms, the results of the examination against such criteria show that the most complete implementation (SDL Trados Studio) obtains only 69% of the total possible points, being 50% the average for all tools. Following to the distribution of marks, we can observe that, while

\(^{66}\) To be read as 244 identified penalty points out of 1074 penalty points allowed; a barely intuitive score format for the ordinary CAT tool user.
very basic items are generally well or sufficiently implemented —customising a metric, creating issue hierarchies and defining issue weights— other items are simply not implemented or just poorly designed. Those neglected points are nevertheless significant, such as using standard formats, standard markup, overlapping spans or the capacity of applying non-continuous marks.

Technical problems such as overlapping spans (i.e. marking the same text span with more than one issue) are not entirely solved even at the XLIFF standard specification level (Filip 2017). For such items to be successfully implemented, tool makers are forced to use the XLIFF in ways different to those devised by the original specification. That is, they need to create customisations or extensions via their own namespaces, which usually has consequences in terms of interoperability. Moreover, tools that base their bilingual format in XLIFF 1.2 do not have a real standard mechanism to properly insert non-nested marks (Savourel 2014b), which is necessary to overlap marked spans. In fact, the tools that support overlapping spans do not use a standard localisation bilingual format (MemoQ, Fluency Now, translate5). On the other hand, SDL Trados Studio, which is based in XLIFF 1.2, does not accept overlapping marks.

Another technical problem that is not solved by existing tools is the capacity of applying non-continuous marks for long-distance dependencies. As recommended by Burchardt and Lommel (2014), “it is vital in creating error markup that errors be marked up with the shortest possible spans. Markup must identify only that area needed to specify the problem. In some cases, this requirement means that two separate spans must be identified”. Although there are workarounds (e.g. using neutral severities to mark more than one location without impacting the quality score), these approaches cannot be considered as real solutions.

What should be said, nevertheless, is that these complex markup requirements would be less of a challenge if tools had adopted a standard like XLIFF 2.0 (OASIS XML Localisation Interchange File Format (XLIFF) TC 2014), which is meant to provide a bigger range of solutions to this kind of problems (Savourel 2014a), such as standalone marks (\texttt{<sm>}, \texttt{<em>}), modules and extensions (Schnabel \textit{et al.} 2015, p.131), or improved ITS 2.0 integration (Porto \textit{et al.} 2013),
which would help in filling the gaps while keeping an standard format for the side-by-side annotation.

5.4 TQA data interoperability: a major issue in CAT tools

Contrary to what one could expect from software company claims, CAT tools do not currently promote interoperability for TQA-related data, which has been summarised in Table 21. As referred in the introduction of this study (see 1.1), the background hopes originated by tool makers in relation to TQA were all about giving feedback (Kilgray), encouraging communicative purposes (Western Standard), involving everyone in the translation supply chain (SDL), or enabling project managers and vendors to view and discuss quality trends (XTM). Although these claims make perfect sense considering that human TQA must provide elements for the translation quality improvement process (Burchardt, Lommel, et al. 2016; Harris et al. 2016), it is necessary that tools provide the technical means for this feedback to be carried out in the best conditions and regardless the systems used by every stakeholder. And, as we have seen, standards are at hand to make this real: XLIFF 2.0 including ITS 2.0 integration or MQM normative aspects regarding markup and metric description, to mention only the most noteworthy aspects.

The degree of compliance with standard formats and standard markup (cf. Intermediate format and Markup items) turned out to be discouraging, in that only one tool was found to perform slightly over 0 points. This aspect highlights once again the endemic interoperability problems of CAT tools that unfortunately are still in force, as already reported by studies in the area of TQA (Harris et al. 2016, p.52). Adopting and complying with localisation standards should not be rocket science, especially in relation to XLIFF 2.0 that is precisely meant “to offer better support to the localisation and translation industry” (Schnabel et al. 2015, p.11). As Torres del Rey and Morado Vázquez (2015, p.605) indicate, “the future of XLIFF and its new version depends to a large extent on its adoption by tool developers and content producers”, adding that “there have already been several good signs” (ibid.). Unfortunately, according to the present findings, these signs must be questioned. Translation software companies even today demonstrate very little interest in making simple features interoperable, ignoring how much
industry and research could benefit from them, as researchers have already pointed out (Harris et al. 2016).

5.5 TQA support according to the software business model

The CAT tools examined here represent different business models: from desktop to web-based applications, and from perpetual licence schemas to subscription-based or open-source solutions. Although the samples analysed here are not enough to draw definitive conclusions, especially considering that many developers still need to integrate TQA in their tools, this perspective may open up opportunities for further research.

As summarized in Table 22, desktop solutions seem to offer a more complete set of TQA features in terms of implementation and interoperability than cloud-based solutions. Reasons for this disparity may be the maturity of the products (MemoQ was released in 200667, SDL Trados Studio in 200968 and Fluency Now in 201069; compared to XTM Cloud, released in 2011 to the general public70 and translate5 in 201371). Another aspect to consider is that TQA is not a core feature in CAT tools by definition, yet it is still demanding in terms of development and design, especially for inline annotations, hence the lighter implementations in web-based platforms where clean and efficient interfaces may still be prioritised. Finally, with respect to the business models (see Table 23), a similar reflection can be made on the difference in TQA coverage between tools with perpetual licence schemas over subscription-based and open-source models. Subscription, open-source and cloud-based solutions are more recent, compared to standard licensed products (MemoQ and SDL Trados Studio), which, besides being desktop applications, have a longer history and a larger number of product

68 http://www.sdltrados.com/about/history.html.
69 Zetsche (2010).
70 https://www.commonsenseadvisory.com/Default.aspx?ContentType=ArticleDetAD&tabID=63&Aid=1369&moduleId=390
71 Uszkoreit and Lommel (2013)
releases, only justified by an ever-growing set of features, including—but not limited to—TQA.
Chapter 6: Conclusion

In the last years, some translation software companies have started to include in their translation tools features that support the execution of translation quality assessment tasks. In this analysis, we have questioned whether these tools offer a complete and satisfactory support to the quality assessment effort and whether the assessment-related data generated by the tools respect basic interoperability aspects. To answer these questions, we have defined what we understand by translation quality assessment in the industry and we have reviewed different areas related to quality assessment and translation technologies, including computer-assisted solutions and localisation standards. Then, we have deconstructed the quality assessment as a global software feature into a set of examination items that could be analysed separately in relation to each tool. The evaluation of these elements has been formalised in a way that scores could be given for each evaluated item. The resulting scores have provided an overview of the degree of support the tools analysed here offer to translation quality assessment tasks as of today.

According to the results of the analysis, we can conclude that CAT tools offering TQA features do support basic aspects of normalised quality assessment tasks. On the bright side, we have confirmed that the five tools examined allow the creation or the import of custom or standard quality metrics, including basic error typologies; four out of five also include a scoring system based on penalties or weights that generate an overall quality score. Regarding quality annotations, four out of five allow flagging issues at a sub-segment level in the target text. On the other hand, we have also determined that more advanced technical aspects such as split or overlapping marks are generally poorly supported or not supported. Finally, the analysis has also shed some light on interoperability aspects, which is the area where CAT tools still need to improve the most, according to the evaluation performed in this study.

The analysis has provided a comprehensive vision of what TQA represents in CAT tools nowadays. However, it must be emphasised that TQA in CAT tools is still in its infancy, being only the most recent versions of a small number of tools that include TQA assessment features nowadays. On the other hand, the consideration made about the variety of tools and the business models they
represent let us think that TQA is already present in different translation technology sectors: from desktop applications to cloud solutions, and from traditional license schemas to open-source and subscription-based models. Our last conclusion is, therefore, that TQA is here to stay and that the future will probably witness an evolution of TQA in different areas such as machine translation evaluation, translator training, client feedback or freelance screening.

6.1 Further research and recommendations

This research has focused on a limited set of tools and a modest number of evaluation items. There is no doubt that same or other CAT tools may already offer more advanced solutions for quality assessment, either out-of-the-box or via extensions, customisations or third-party plug-ins using available APIs. Therefore, the picture given in this study should be understood only as an introductory overview of the state of the question, which is meant to evolve rapidly in the years to come. Notwithstanding, some of the tools analysed here are market leaders and cover a significant part of the translation and localisation market, which let us think that the approaches reflected here represent already a trend and that any further analysis should take them into account.

Finally, the present study has shown some weaknesses in the implementation of TQA features in CAT tools. As we have seen, the most concerning aspects that need urgent improvement are those related to interoperability. Even assuming that commercial interests play a role in the limited openness and compliance shown by CAT solutions, it is nevertheless necessary to recall that TQA benefits are precisely being an invaluable instrument to improve translation quality throughout the translation cycle. By enforcing standard and open means to store and transmit translation quality information —including but not limited to standard metric descriptions, XLIFF 2.x-based annotated files, normalised quality scores and complete quality reports—, translation software companies can pave the way to improve overall quality levels in many significant ways, from machine translation output to richer and more accessible quality feedback mechanisms for language vendors, linguists and students.
References


Stejskal, J. (2009) ‘Quality Assessment in Translation’, *MRA’s Alert! Magazine*, (7), - 113 -


Annex – Metric definitions

MQM Core version 1.0 (2015-12-30)

Fluency Now default

```xml
<?xml version="1.0" encoding="UTF-8"?>
<mqm version="2.0">
  <head>
    <name>Small metric</name>
    <description>A small metric intended for human consumption</description>
    <version>1.5</version>
    <source>http://www.example.com/example.mqm</source>
  </head>
  <issues>
    <issue type="accuracy" display="no"/>
    <issue type="omission" weight="7.0"/>
    <issue type="addition" weight="2.5"/>
    <issue type="mistranslation" weight="8.5"/>
  </issues>
  <issues>
    <issue type="fluency" display="no"/>
    <issue type="spelling" weight="7.5"/>
    <issue type="grammar" weight="6.3"/>
    <issue type="punctuation" weight="4.0"/>
    <issue type="mechanical" weight="2.5"/>
  </issues>
  <issues>
    <issue type="style" weight="5.0"/>
    <issue type="locale" weight="5.0"/>
  </issues>
  <displayNames>
    <displayNameSet lang="en">
      <displayName typeRef="accuracy">Accuracy</displayName>
      <displayName typeRef="omission">Omission</displayName>
      <displayName typeRef="addition">Addition</displayName>
      <displayName typeRef="mistranslation">Mistranslation</displayName>
      <displayName typeRef="fluency">Fluency</displayName>
    </displayNameSet>
  </displayNames>
</mqm>
```

Figure 53. MQM Core schema.
Fluency Now custom metric (MQM-compliant)

<?xml version="1.0" encoding="UTF-8"?>
<mqm version="0.9">
<head>
  <name>Metric for MSc Dissertation on TQA and translation tools</name>
  <desc>A metric intended for analysis purposes</desc>
  <version>1.0</version>
</head>
<issues>
  <issue type="accuracy" weight="1"/>
  <issue type="mistranslation" weight="1"/>
  <issue type="omission" weight="1"/>
  <issue type="addition" weight="1"/>
  <issue type="fluency" weight="1"/>
  <issue type="spelling" weight="1"/>
  <issue type="unintelligible" weight="1"/>
  <issue type="terminology" weight="1"/>
  <issue type="style" weight="1"/>
</issues>
<displayNames>
  <displayNameSet lang="en">
    <displayName type="accuracy">Adequacy</displayName>
    <displayName type="terminology">Terminology</displayName>
    <displayName type="omission">Omission</displayName>
    <displayName type="addition">Addition</displayName>
    <displayName type="fluency">Fluency</displayName>
    <displayName type="style">Style</displayName>
    <displayName type="spelling">Spelling</displayName>
    <displayName type="unintelligible">Unintelligible</displayName>
  </displayNameSet>
</displayNames>
<severities>
  <severity id="none" multiplier="0"/>
  <severity id="minor" multiplier="1"/>
  <severity id="major" multiplier="5"/>
  <severity id="critical" multiplier="10"/>
</severities>
</mqm>

translate5
SDL Trados Studio

<?xml version="1.0" encoding="utf-8"?>
<TQAProfile xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" Version="1.0"
Created="2017-08-03T17:49:53.2279566+02:00">
  <Threshold LenghtValue="1000" ScoreValue="50" UnitKey="Words" />
  <CategoryItems>
    <CategoryItem Id="a53bf5a0-bcf9-4abf-afe4-bc8a04028aa9" Name="Accuracy">
      <Abbreviation>ACC</Abbreviation>
      <CommentHint />
      <CommentRequired>true</CommentRequired>
      <Description>Accuracy category for the MQM Core translation quality assessment framework.</Description>
      <SubCategories>
        <SubCategoryItem Id="d171d276-0eb2-4b2b-9707-bcc15fa5d6d2" Name="Mistranslation" ParentId="a53bf5a0-bcf9-4abf-afe4-bc8a04028aa9">
          <CommentHint>Example: A source text states that a medicine should not be administered in doses greater than 200 mg, but the translation states that it should be administered in doses greater than 200 mg (i.e., negation has been omitted).</CommentHint>
          <CommentRequired>true</CommentRequired>
          <Description>The target content does not accurately represent the source content.</Description>
        </SubCategoryItem>
        <SubCategoryItem Id="c637601f-1569-401f-b029-94f04f678fb1" Name="Omission" ParentId="a53bf5a0-bcf9-4abf-afe4-bc8a04028aa9">
          <CommentHint>Example: A paragraph present in the source is missing in the translation.</CommentHint>
          <CommentRequired>true</CommentRequired>
          <Description>Content is missing from the translation that is present in the source.</Description>
        </SubCategoryItem>
        <SubCategoryItem Id="2289a9fb-00b0-45a9-b5e2-3b1a783882e9" Name="Addition" ParentId="a53bf5a0-bcf9-4abf-afe4-bc8a04028aa9">
          <CommentHint>A translation includes portions of another translation that were inadvertently pasted into the document.</CommentHint>
          <CommentRequired>true</CommentRequired>
          <Description>The target text includes text not present in the source.</Description>
        </SubCategoryItem>
      </SubCategories>
    </CategoryItem>
    <CategoryItem Id="59ca105a-91d0-48a0-802f-23f22e350238" Name="Fluency">
      <Abbreviation>FLU</Abbreviation>
      <CommentHint />
      <CommentRequired>true</CommentRequired>
      <Description>Fluency category for the MQM Core translation quality assessment framework.</Description>
      <SubCategories>
      </SubCategories>
    </CategoryItem>
  </CategoryItems>
</TQAProfile>

- 117 -
<SubCategoryItem Id="6bba08c8-3722-4f1f-a9f5-032c8bd6e359" Name="Spelling" ParentId="59ca105a-91d0-48a0-802f-23f22e350238">
  <Abbreviation />
  <CommentHint>Example: The German word Zustellung is spelled Zustetlugn.</CommentHint>
  <CommentRequired>true</CommentRequired>
  <Description>Issues related to spelling of words.</Description>
</SubCategoryItem>

<SubCategoryItem Id="4f8ab5ad-86ff-4d9a-ba5e-0d5707c1e55" Name="Unintelligible" ParentId="59ca105a-91d0-48a0-802f-23f22e350238">
  <Abbreviation />
  <CommentHint>Example: The following text appears in an English translation of a German automotive manual: “The brake from whe this part numbr,..”</CommentHint>
  <CommentRequired>true</CommentRequired>
  <Description>The exact nature of the error cannot be determined. Indicates a major break down in fluency.</Description>
</SubCategoryItem>

<CategoryItem Id="fdc1156a-d384-42d5-93dd-02435ae9526f" Name="Terminology">
  <Abbreviation>TER</Abbreviation>
  <CommentHint>A French text translates English e-mail as e-mail but terminology guidelines mandated that courriel be used. The English musicological term dog is translated (literally) into German as Hund instead of as Schnarre, as specified in a terminology database.</CommentHint>
  <CommentRequired>true</CommentRequired>
  <Description>All issues specifically related to use of domain- or organization-specific terminology are included in this issue.</Description>
</CategoryItem>

<CategoryItem Id="68ecada5-a046-493d-be48-3a0d470c1be2" Name="Style">
  <Abbreviation>STY</Abbreviation>
  <CommentHint>STY</CommentHint>
  <CommentRequired>true</CommentRequired>
  <Description>The translation of a light-hearted and humorous advertising campaign is in a serious and “heavy” style even though specifications said it should match the style of the source text.</Description>
</CategoryItem>

<SeverityItem Id="5416ebd6-3f67-4e02-8626-bf4e771e539a" Name="None" OrderInList="1" Score="0" />
<SeverityItem Id="6d95848a-9991-431c-834a-5e04ead85743" Name="Minor" OrderInList="2" Score="0" />
<SeverityItem Id="05273968-595c-4973-a2ab-27e9fa38f629" Name="Major" OrderInList="3" Score="0" />
<SeverityItem Id="d6b4d8b6-d8bf-47d9-bb44-e4418b7eaccd" Name="Critical" OrderInList="4" Score="0" />
</SeverityItems>
</TQAProfile>
<MemoQResource ResourceType="LQA" Version="1.0">
  <Resource>
    <Guid>6e131957-6c8b-4ce3-9d38-0e80eaff5c7</Guid>
    <FileName>TQA.mqres</FileName>
    <Name>TQA</Name>
    <Description />
  </Resource>
</MemoQResource>
<?xml version="1.0" encoding="utf-8"?>
<XMLSerializedLQAModel xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Version>1</Version>
  <Name>TQA</Name>
  <RevisionID>93e0799b-4643-4b85-83ae-cf935012c171</RevisionID>
  <UseSeverityLevels>true</UseSeverityLevels>
  <UsePenaltyPoints>true</UsePenaltyPoints>
  <RequiresReviewerCommenting>false</RequiresReviewerCommenting>
  <SeverityLevelNames>
    <string>Critical</string>
    <string>Major</string>
    <string>Minor</string>
    <string>Neutral</string>
  </SeverityLevelNames>
  <ModelEntries>
    <ModelEntry>
      <CategoryName>Accuracy</CategoryName>
      <SubcategoryName>Mistranslation</SubcategoryName>
      <CatSubcatDescriptor>
        <ErrorCategoryComment />
        <SeveritiesAndPenalties>
          <SeverityWithPenalty>
            <SeverityIdx>0</SeverityIdx>
            <PenaltyPoint>100</PenaltyPoint>
          </SeverityWithPenalty>
          <SeverityWithPenalty>
            <SeverityIdx>1</SeverityIdx>
            <PenaltyPoint>10</PenaltyPoint>
          </SeverityWithPenalty>
          <SeverityWithPenalty>
            <SeverityIdx>2</SeverityIdx>
            <PenaltyPoint>1</PenaltyPoint>
          </SeverityWithPenalty>
          <SeverityWithPenalty>
            <SeverityIdx>3</SeverityIdx>
            <PenaltyPoint>0</PenaltyPoint>
          </SeverityWithPenalty>
        </SeveritiesAndPenalties>
      </CatSubcatDescriptor>
    </ModelEntry>
    <ModelEntry>
      <CategoryName>Accuracy</CategoryName>
      <SubcategoryName>Omission</SubcategoryName>
      <CatSubcatDescriptor>
        <ErrorCategoryComment />
        <SeveritiesAndPenalties>
          <SeverityWithPenalty>
            <SeverityIdx>0</SeverityIdx>
            <PenaltyPoint>100</PenaltyPoint>
          </SeverityWithPenalty>
          <SeverityWithPenalty>
            <SeverityIdx>1</SeverityIdx>
            <PenaltyPoint>10</PenaltyPoint>
          </SeverityWithPenalty>
        </SeveritiesAndPenalties>
      </CatSubcatDescriptor>
    </ModelEntry>
  </ModelEntries>
</XMLSerializedLQAModel>
<PassFailCriterion>
  <CritDefinitionRules />
  <QALQAMapping />
  <Timestamp>2017-09-10T09:11:45.6092494Z</Timestamp>
</XMLSerializedLQAModel>