[*Translating and the Computer* 29, November 2007]

Translation Quality Assurance Tools: Current State and Future Approaches

Julia Makoushina

Palex Languages and Software

Tomsk, Russia

julia@palex.ru

Contents

Abstract

Compared to translation memory tools, the market currently offers a rather limited number of automated translation quality assurance tools, and those available differ greatly in approaches, functions and prices.

The purpose of this paper is to review translation quality assurance tools, to define where they currently are, their advantages and disadvantages and to visualise their future capabilities and role in translation process. The review is done from the viewpoint of a practitioner with software development and testing background and is highly focused on what real features need to be implemented in such tools in the nearest future.

In order to make the review more valuable we carried out a survey among translation professionals to identify the most popular quality assurance tools, the overall acceptance and common usage of such tools, and their desired features and capabilities.

During this presentation we will briefly examine how translation quality assurance tools developed, consider what they have in common and point out their unique features, benchmark their performance and discuss how the translation community accepts and uses them, what translators expect of them and what kind of future awaits them.

This paper focuses only on those translation quality assurance tasks that may be formalised. Linguistic quality assurance as well as software localisation quality assurance is beyond its scope.

Introduction

While translation volumes are rapidly growing and turnaround time is shrinking, the translation workload is increasing exponentially. Adding more and more languages builds up pressure. What remains unchanged is the necessity to maintain the required quality level. In such a situation automating quality assurance tasks may become a viable strategy for language service providers.

Translation quality assurance tasks can be easily grouped into two categories. The most conventional definition of translation quality is that the translated text should be grammatically correct, have correct spelling and punctuation and sound as if it was originally written by a native speaker of the target language. We will refer to all quality assurance tasks performed to ensure this type of quality as *linguistic.* Obviously most of these tasks require human intervention and are hard to be automated.

However, there is another aspect of quality which we will call *formatting.* Ensuring such type of quality means not only making certain that the original text formatting was not damaged, but also detecting unnecessary double spaces, double full stops at the end of a sentence, verifying that the same sentences were not accidentally translated differently and that the project glossary was followed, and numerous other tasks which do not require working knowledge of the target language. Usage of translation memories gives rise to the need to ensure that the translator did not apply fuzzy matches as perfect ones, made necessary corrections in the fuzzy matches as well as to make certain that the perfect matches were correct and did not get into the translation memory by mistake. Apparently these tasks require a lot of attention, even thoroughness, and are therefore highly error-prone. It is not easy to find a human being who could produce fast and consistent results 8 hours a day, 5 days a week.

Formatting tasks are monotonous and boring, but fortunately easy to formalise, which makes them ideal candidates for automation. Ceasing to rely on how sharp are the eyes of human quality assurance specialists would significantly increase both the error detection level and the throughput.

Key concepts

To ensure better understanding we'll briefly define some terms used herein.

TM - translation memory, a database of pairs of segments (usually sentences) where one segment in the pair is the translation of another.

TM tools - software applications that support translator's work with TM, selecting necessary segments, creating new pairs, correcting existing pairs etc.

100% match (perfect match) - a record in a TM where source segment is equal to a segment being translated.

Fuzzy match - a record in a TM where source segment is similar (to some definite extent) to a segment being translated.

QA - quality assurance, a variable set of measures and procedures performed to ensure that translated text contains no errors and/or to detect and correct existing errors. Within this paper, we'll use the term "QA" only with regard to translation.

QA tools, QA automation tools - software applications that help translators and/or QA managers perform QA procedures.

Linguistic checks - a set of procedures performed to make sure translated text is grammatically and stylistically correct and uses applicable terminology and correct spelling.

Formatting checks - **a** set of procedures performed to make sure translated text uses correct separators for digits and quotation/punctuation marks applicable to the target language, has no unnecessary spaces and has all necessary spaces in place etc.

Empty translation - an empty segment in the target language.

Forgotten translation - a segment in the target language that is identical to the segment in the source language.

Skipped translation - a segment that was never opened (in TM systems that "open" and "close" segments for translation such as Trados and Wordfast).

Partial translation - a segment in the target language that contains some sequential words found in the source language.

Incomplete translation - a segment in the target language that is significantly shorter than the segment in the source language.

Corrupt characters - characters that are in no case used in the target language.

Inconsistent sentence count - a segment in the target language that consists of more or less sentences than the segment in the source language.

Source inconsistency (inconsistency in source segments) - different source segments are translated equally.

Target inconsistency (inconsistency in target segments) - identical source segments are translated differently.

Punctuation at the end of segment - a check that ensures source and target segment have the same punctuation mark at the end.

Spaces before punctuation - a check that ensures all necessary spaces before punctuation marks are in place and no unnecessary spaces before punctuation marks are left in the target segment.

Double spacing - a check that ensures the target segment does not contain consequent spaces.

Double dots (double full stops) - a check that ensures the target segment does not contain consequent dots.

Double punctuation - a check that ensures the target segment does not contain consequent punctuation marks.

Number formatting - **a** check that ensures numbers in target segments use correct decimal and thousand separators.

Blacklist - a check that ensures the target segment contains no blacklisted words. Blacklists may contain unwanted words or misspellings that a spell-checker cannot catch.

Functionality - the ability of an application to detect and report errors. Within this paper, we'll only use this term in relation to QA tools.

Efficiency in speed - the response time of an application. Within this paper, we'll only use this term in relation to QA tools.

Efficiency in error detection - the ratio of true and false (or undetected) errors detected. Within this paper, we'll only use this term in relation to QA tools.

Reliability - a characteristic that refers to how rare the application crashes or hangs up. Within this paper, we'll only use this term in relation to QA tools.

Usability - a characteristic that refers to ease of operation and use of an application. Within this paper, we'll only use this term in relation to QA tools.

Learnability - a characteristic that refers to the ease of performing basic tasks for first-time users of an application. Within this paper, we'll only use this term in relation to QA tools.

Value for money - the benefit obtained for a given amount of money. Within this paper, we'll only use this term in relation to QA tools and only as subjective estimation.

Customer support - the characteristic that refers to how fast and accurate customer support responses were in case of problems with an application. Within this paper, we'll only use this term in relation to QA tools and only as subjective estimation.

Adaptability - the characteristic that refers to the ease of an application customisation according to specific needs. Within this paper, we'll only use this term in relation to QA tools.

The actual state of translation quality assurance tools

QA tools: retrospection and state of the art

Whereas translation memory tools came into the market approximately in 1985, translation quality assurance tools are rather young. The oldest quality check utilities were probably incorporated into Star Transit. Back in 1998, it already offered formatting, terminology and spelling checks. This means there is a 10-15 years gap in TM and QA tools development.

With constant TM technology development and its increasing penetration into translation community, the translation market is becoming more and more demanding not only to turnaround time, but also to terminology and consistency.

On the other hand, wide application of TM tools resulted in new types of errors, such as implementing a fuzzy match without necessary corrections which are also more predictable. Such corrections are often related to figures, tags and slight text changes that should or should not be reflected in translation.

These aspects of TM penetration became reasons for the QA automation tools to appear. Unfortunately such tools appeared much later than they should, and are currently developing not as rapidly as users would like them to.

As it was already mentioned, Star Transit has been employing some kind of checks for almost 10 years. However, those checks don't seem to have improved and/or extended with later versions. Transit still has the most limited QA functionality. On the other hand, as we will see from the benchmark, it proves to be the most stable checking tool from language to language as well as one of the closest to the functionality claimed.

SDLX included terminology QA check feature in 2003, but SDL extended its scope from terminology to other issues only in 2005. Its capabilities are still rather limited, but allow for some extension due to regular expressions. Users can formalise many error types they encounter often enough. In practice, it still poses some difficulties as people responsible for QA are often unable to create regular expressions correctly. Normally they require additional help from technically skilled people like software developers.

Trados (particularly, TagEditor) got the QA functionality only from version 7 which was released in 2006 after SDL/Trados merger. In addition to its default checks enabled by checkboxes it also provides capabilities for extension through regular expressions.

Probably all TM tools currently include some QA features, at least the most popular of them (SDLX, Trados, Déjà Vu and WordFast) do. The latter two allow users to include their own quality check macros and SQL[1] queries respectively, which definitely extends the number of error types the tools are able to catch, but again, as with SDLX and Trados, creating macros and SQL queries requires technical skills that people responsible for QA do not normally possess.

As an alternative to QA plug-ins and QA features of TM tools, a series of standalone QA tools are also available on the market. The most popular of them are Yamagata's QA Distiller, ApSIC's XBench and DOG's ErrorSpy.

QA Distiller is the oldest one among them. It was developed in Yamagata Europe to automate the detection of measurable errors in translation, and version 3.0.7 was eventually presented to the public during the LISA 2004 conference in Saint-Petersburg[2]. The latest version of QA Distiller to date is 6.0, and it is now probably the most comprehensive and usable (although one of the most expensive) QA tool on the market.

---

[1] Structured Query Language, a standardized computer language for defining and manipulating data in a relational database.[1] Structured Query Language (SQL) (HTML). International Business Machines (October 27, 2006).

[2] http://www.lisa.org/archive/forums/2004spb/presentations.html

The latest version of ErrorSpy to date is version 4.0 which has significantly changed compared to version 3.0, at least with regard to functionality and efficiency. Version 3.0 was initially tested for this paper, and then, when version 4.0 was released, we had to re-test the application. The number of detected errors has noticeably increased while the amount of false positives has decreased.

XBench is the youngest and the only free tool on the market. The most current version is 2.7, which according to ApSIC's Web site is still under beta testing. Additionally, it differs from other QA tools in many aspects. Firstly, it is not a pure QA tool, but a multi-dictionary tool. It allows to import files of numerous different formats simultaneously and use them as glossaries as well as for concordance search. It also supports online terminology search. To employ its QA capabilities, the user has to assign one set of files as ongoing translation. Some other may be assigned as project glossaries that should be adhered to. Secondly, it currently supports the widest variety of input file formats, and thirdly, no technical knowledge is required to create additional checks and checklists.

QA tools classification

Existing QA tools may be classified according to several criteria shown in Table 1 below.

Criterion	Implementation	
Architecture	Standalone tool that accepts many different file formats and possibly offers additional functions not directly related to QA	TM tool plug-in/integral part
Usability	To customise the tool, users have to possess technical skills (customisation is done via SQL queries, regular expressions, macros etc.)	To customise the tool, users do not need technical skills, customisation involves easy Boolean logic like AND, OR etc.
Learnability	Minimum set of checks is enabled by default; additional learning is required to enable other checks. Running a program without any customisation results in detecting few types of detectable errors	Maximum set of checks is enabled by default, running a program without any customisation results in detecting most of detectable errors

Table 1. QA tools classification criteria

The first and most obvious criterion was already mentioned above: according to their architecture tools are divided into standalone applications and plug-ins. Plug-in tools are usually designed to QA translations performed in the respective TM environment and are normally not useable for other file formats because this would require file conversion which often results in corruption of internal tags and TM tool-specific properties. Standalone tools, on the other hand, often support many different input file formats and don't directly make any changes in the bilingual texts, but normally allow opening them in their native TM tool and making all necessary changes there.

Another criterion for classification may be the approach to extension, refining and customisation of checks. While some tools like Déjà Vu or Trados QA Checker require some developer knowledge to extend functionality others such as XBench allow less technical users to easily create their own rules to check. According to this criterion, one tool (Star Transit) stands by itself. With rather limited amount of preset checks, it does not allow extending the check set and allows only minimum customisation according to target language.

Existing QA tools may also be classified according to the default check set. Some of them are almost "out of the box" solutions where the user may only change some checkbox configurations and perform all necessary checks. QA Distiller, Star Transit and SDLX QA Check belong to this group of tools. The default configuration of other tools allows only to perform rather limited check set, while other checks are also potentially supported via additional more complicated configurations and/or regular expressions. These tools include Déjà Vu, Wordfast and XBench. In this respect ErrorSpy is a unique tool because although its customisation is easy enough, at first it requires some actions outside of the program's interface.

The table below represents summary classification of existing QA tools; however, it doesn't claim to be comprehensive.

And yet another criterion to classify QA tools is their ability to handle different encodings and scripts.

Those characteristics of each individual QA tool will be considered in detail in the benchmark section of the paper.

Tool	Architecture		Usability		Learnability	
	Standalone	Plug-in	Technical skills required	Technical skills not required	Minimum default check set	Maximum default check set
Déjà Vu		✓	✓		✓	
ErrorSpy	✓		✓			✓
QA Distiller	✓		✓			✓
SDLX QA Check		✓	✓		✓	
Star transit		✓		✓		✓
Trados QA Checker		✓	✓		✓	
Wordfast		✓	✓		✓	
XBench	✓			✓		✓

Table 2. QA tools classification

QA checks classification

For the purpose of this research, we have divided all check types into several groups:

Segment-level checks include checks for untranslated, skipped, incomplete, partially translated, forgotten segments as well as for segments containing corrupt characters and segments that consist of different amount of sentences in source and target language.

Inconsistency checks include detecting equal segments that are translated differently, different segments translated equally as well as checking word-level consistency.

Punctuation checks involve comparison of punctuation at the end of segment, spaces before punctuation (normally the tools should check that there are no spaces with some exceptions such as for French), double spacing, double punctuation marks (including, but not limited to double full stops[3]), brackets and parentheses as well as apostrophes and quotation marks.

Number checks mean comparison of number values and formatting (decimal and thousand separators), digit-to-text conversion, measurement unit conversion etc.

Terminology checks confirm project glossaries adherence and equivalence of untranslatables in the source and the target language. They also include checks against black list with possible correction of misspelled or unwanted words.

Tag checks ensure equivalence and correct order of tags in source and target texts in tagged formats like HTML[4], XML[5], MIF[6] etc.

[3] It must be noted that tools should also distinguish double dots from triple dots as ellipsis character is not always applicable.

[4] Hypertext markup language, a markup language for Web pages.

[5] Extensible markup language, a general-purpose markup language.

[6] Maker interchange format, a markup language for Adobe FrameMaker.

Each QA tool implements those kinds of checks to some degree, and we can easily classify them by this degree as well as by the kind of false positives they generate for each type of checks.

Survey description

To evaluate QA automation tools acceptance and usage by the translation community, we conducted an on-line survey. This is, to our knowledge, the first survey of that kind in the industry. The survey was aimed at translation/localisation service providers and buyers rather than at freelance translators; therefore we expected to get a rather limited amount of responses, in particular, from 150 to 200.

In order to refine survey questions and answers and make them as non-ambiguous as possible, test interviews were conducted with QA and project managers at Palex (8 participants in total). The survey questions were amended and changed according to the feedback received, which resulted in a 30-question survey that still was rather complicated. To avoid complex survey logic, we made it flat and presumed the questions that are not relevant to a respondent's experience and competence will not be answered.

The survey was opened online on August 1st, 2007 and closed on September 27th, 2007. During this period, it was actively promoted in the translation and localisation community[7].

The main goals of the survey were:

- to evaluate awareness of existing QA automation tools in the industry and QA technology penetration;
- to distinguish approaches to quality assurance according to organisation size and type;
- to reveal the environment QA managers work in;
- to discover strong and weak sides of existing QA tools and QA automation in general;
- to reveal types of QA checks performed regularly as well as those which are desirable to perform and automate;
- to find out the readiness to automate QA checks and the reasons for not checking translation quality and for avoiding automation;
- to reveal content types and formats currently supported by QA automation tools and those not yet supported;
- to identify languages and scripts that pose difficulties for QA automation;
- to identify the possibilities for expanding QA tools functionality and application.

Survey results and analysis

181 professionals responded to the survey during two month. Not all of them answered all questions; however, it was presumed that specialists in different areas may reply to different questions. 169 responses were valid for analysis, so the drop-off rate was below 7%.

QA tools user profiles

As expected, most of the respondents (141 or 86.5%) represented translation/localisation service provider companies while a few (more specifically, 11 people) were from service buyer side and 2 were software developer representatives. 3.07% of other organisations were consulting and academic institutions, and one respondent reported his organisation to be multilingual quality assurance service provider. We didn't classify this organisation as a translation/localisation service provider because QA is the only service it provides which makes this company rather unique.

[7] It must be noted that the survey was performed during the vacation time which also limited the number of respondents. After the survey was closed, many people showed and are still showing their interest in participation.

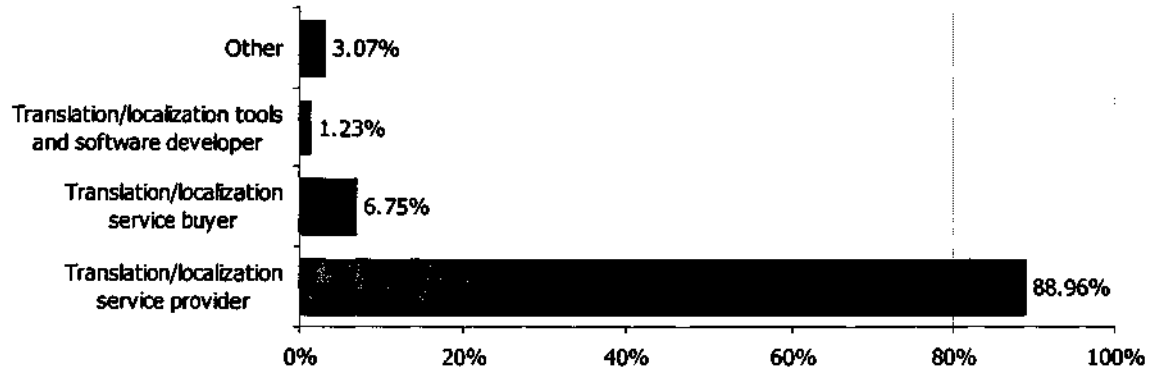

Figure 1. Types of organisations participated in the survey

Vast majority of the respondents (121 people or 76.58%) represented companies with no more than 50 employees. Half of this number represented small companies consisting of 1-5 people. 8 respondents (around 5%) represented very large companies (500+ employees).

For comparative analysis, we divided companies into three groups: small (1-5 employees), medium (5-100 employees) and large (more than 100 employees).

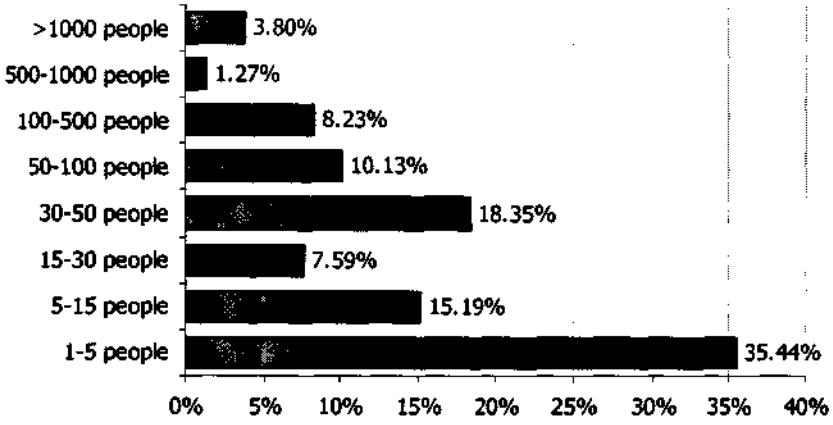

Figure 2. Organisation size (number of employees)

By working status, almost equal number of company owners and company employees participated in the survey, with the number of freelance translators about twice less. The number of company owners vs. company employees was almost in inverse proportion to company size as illustrated in Figure 4.

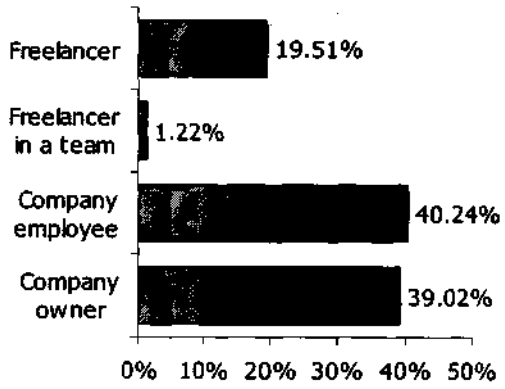

Figure 3. Working status of the respondents

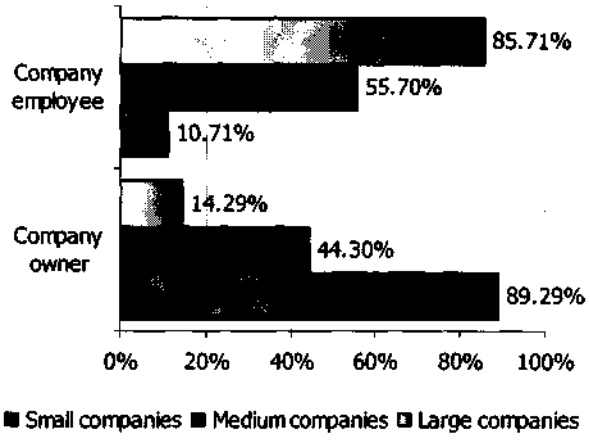

Figure 4. Distribution of company employees and owners according to company size

While most of the respondents are translators and/or executives, the amount of project and QA managers who gave their feedback is also considerable (33 and 10 respectively). Over 7% of other occupations include a significant number of localisation professionals as well as a few representatives of marketing services and application developers.

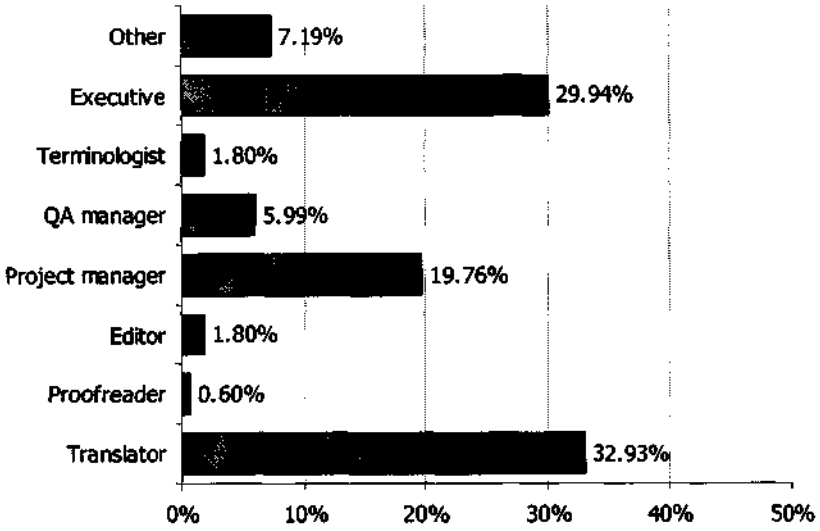

Figure 5. Types of professionals by occupation

Working Environment and Translatable Content

Exactly 1/4 of all respondents indicated that their companies translate (either directly or via a translation service provider) more than 10,000,000 source words, with small companies never exceeding the limit of 5 million source words per year.

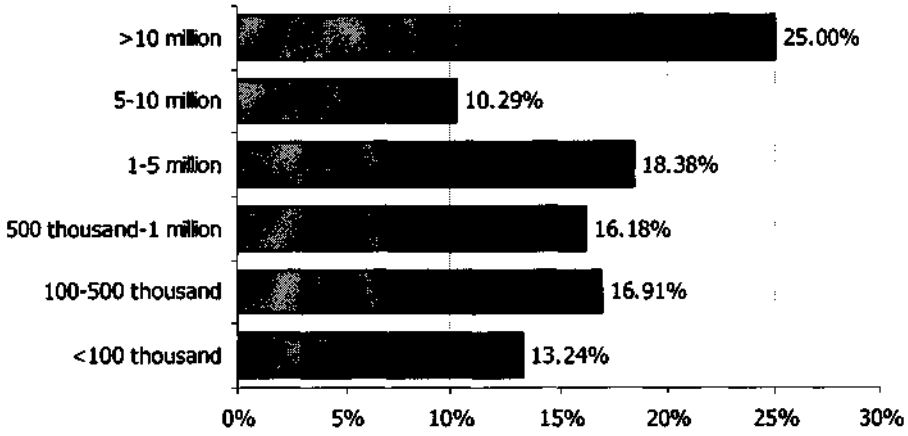

Figure 6. Yearly translation volumes (in source words)

Almost 1/3 of the respondent companies translate their content into more than 30 languages. In general, monolingual companies or freelancers comprised only 7.33% of the respondents. It was not surprising that companies translating into more than 30 languages are mainly large and medium-sized. Less than 14% of small companies indicated they translate into more than 5 languages.

It is also not surprising that companies translating into only one language translate no more than 1 million source words per year while companies which translate over 10,000,000 source words normally handle more than 30 languages.

Another 1/3 of the respondents (vast majority of them represent small companies) indicated their companies to handle 2-5 languages.

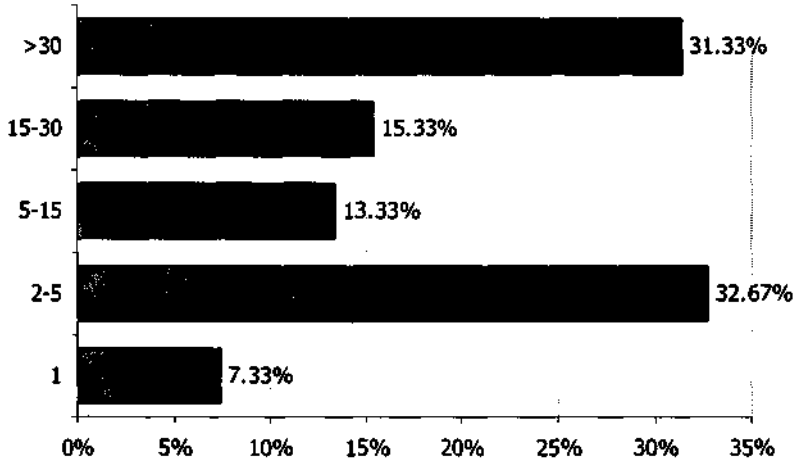

Figure 7. Number of target languages

Almost all localisable content types fall into the same percentage range. The most popular ones, however, are technical materials (almost 23%) and software (almost 18%). Among other content types (3.4%), respondents indicated general subject, tourism and leisure, games, subtitling, academic and historic materials as well as various internal documentation and communications. The distribution of content types by translation volumes is almost equal.

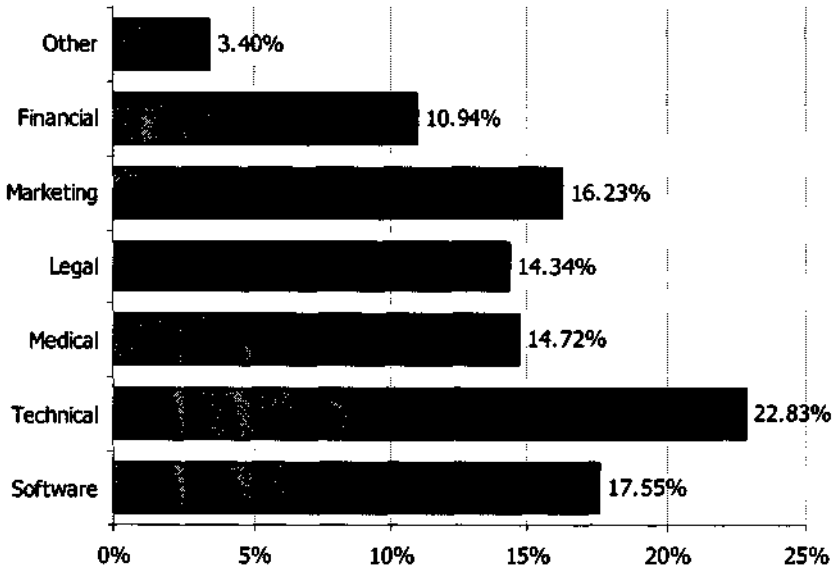

Figure 8. Types of translatable content

Microsoft Word is the leading format of translatable documents (13.55%) with all Office formats taking up 31.77%. Desktop publishing application formats comprise approx. 19% of all translatable documents. Many respondents indicated they often got PDF[8] documents for translation (9.35%). Among other formats, most respondents indicated ready TM files such as .itd for SDLX and .ttx for Trados TagEditor, content management system's formats and numerous proprietary formats from translation/localisation service buyers.

[8] Portable Document Format, a file format for document exchange (created by Adobe Systems).

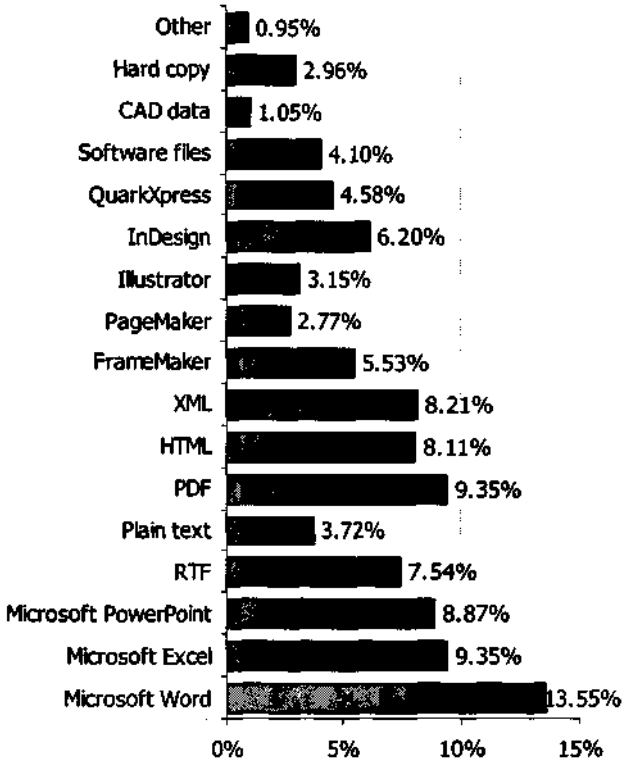

Figure 9. Formats of translatable content

It is interesting to note that the most popular formats in large companies are Microsoft Word and XML (both account for 11.03% of total formats) while apparently it is small companies that translate most of hard copy documents. On the contrary, small companies are not inclined to handle "complex" formats like DTP[9] applications, CAD[10] tools or HTML/XML preferring MS Office and PDF.

The most popular operating system is Microsoft Windows, and 62.57% of respondents confirm their companies work only in MS Windows with no other OS[11]'s. Users of both Windows and MacOS who follow Windows users comprise only 19.35%. Users of three OS's (Windows, MacOS and Unix/Linux) account for approx. 9%, and those who work under Windows and Linux comprise 7.1% of all respondents. 0.65% (1 respondent per each category) work only in MacOS, Unix/Linux and other (medical hardware) OS.

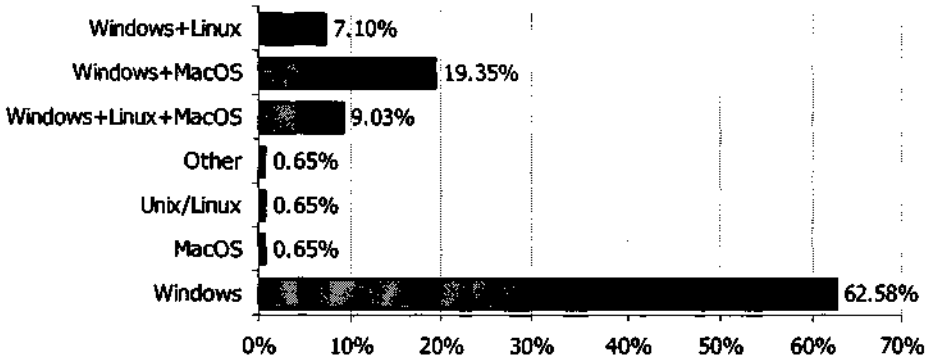

Figure 10. Operating systems

[9] Desktop publishing. DTP tools assist to create publication documents on the computer. Most popular DTP applications include QuarkXPress, Adobe InDesign, Microsoft Publisher and others.

[10] Computer-aided design. CAD tools assist engineers, architects and other design professions in their work.

[11] Operating system.

Usage of Translation Memory Tools and Quality Assurance Practices

After SDL/Trados merger, SDL translation memory tools are indeed prevailing. Almost 60% of respondents use Trados and/or SDLX as their translation memory solution. Star Transit (11.11%) is the third popular TM according to the feedback, and Wordfast and Déjà Vu account for 9.8% and 7.84% respectively. Other tools mentioned were across, Idiom, Logoport, MemoQ, Lingotek, Heartsome, MultiTrans, OmegaT, WordFischer and proprietary tools. Many respondents also named Passolo, Catalyst, RC-WinTrans, Helium, LocStudio and other localisation tools which, however, are beyond the scope of the paper. 4.9% of the respondents stated they don't use any translation memory tool at all.

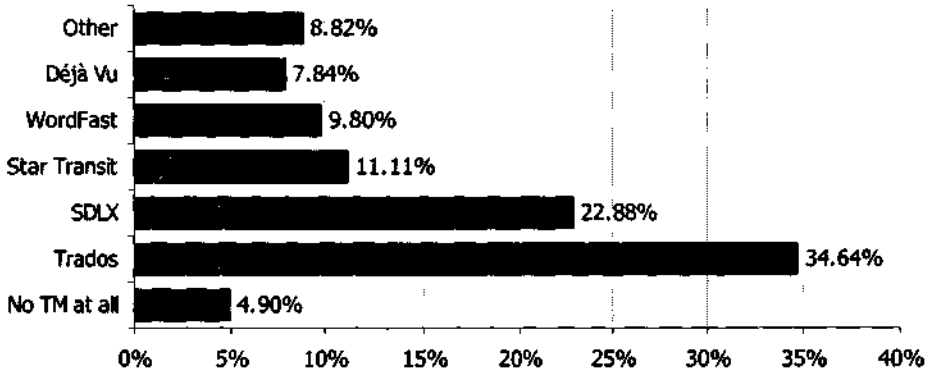

Figure 11. Translation memory tools used

Trados and SDLX are still prevailing both in companies of all sizes. However, WordFast occupies the third position in the group of small companies, where the fourth most popular answer was "No TM at all" - 9.18% of small companies don't use any TM tool. Companies of medium size probably tend to be flexible and are therefore actively using almost all TM tools (they account for the lowest number of "No TM" responses) while large companies are the most active users of other TM tools (usually Idiom, across, proprietary and localisation tools).

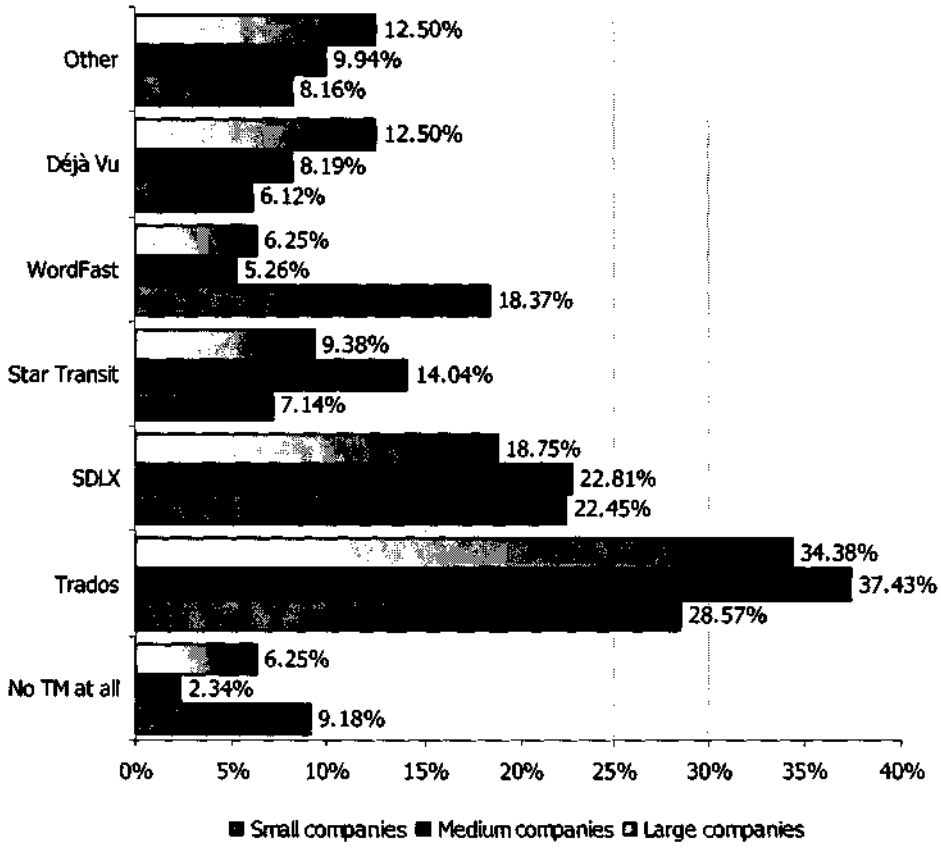

Figure 12. Translation memory tools according to company size

29.51% of the respondents using TM tools employ only one of them. Most companies employ 2 or 3 solutions, and some even tend to use all TM tools possible. Large companies tend to be slightly less flexible: none of 14 large company representatives indicated they would use more than 5 TM tools. In fact, only one representative of a small company and three of medium-sized companies reported they use 5 and more TM tools.

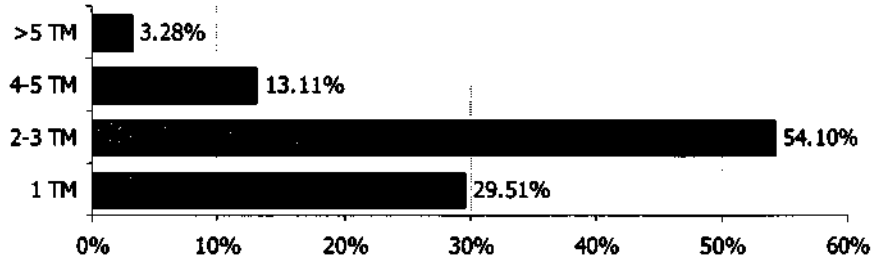

Figure 13. Number of translation memory tools used

With regard to quality assurance, almost 40% of the companies (with 54.41% of medium-size companies) have 2-5 people responsible for them. Almost 27.5% have only one employee responsible for QA (2/3 of this number are small companies). Over 26% of the respondents replied they have more than 5 people in their QA team, and 2/3 of them are large companies. It is obvious that small companies may not fall into this segment at all. On the other hand, 6.67% of the companies, most of which are small ones, have no QA staff at all.

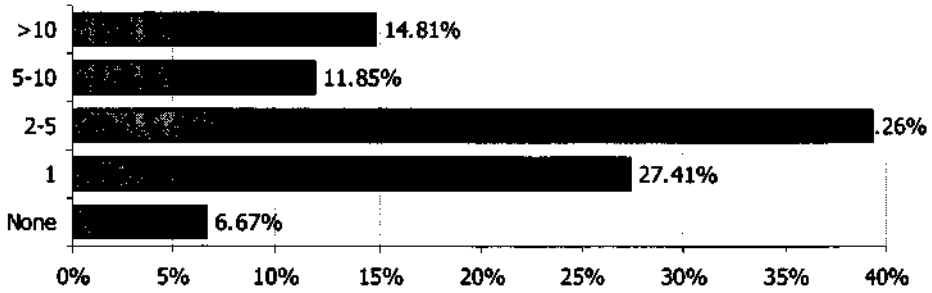

Figure 14. Amount of people responsible for quality assurance

Although there is no direct relation between the amount of people responsible for QA and the translation volumes, the overall trend is that the more words the organisation translates the more people it needs to perform QA tasks. While organisations that translate less than half a million words mostly have no more than 1 QA specialist and never have more than 5 people in this capacity, companies that translate over 10 million words normally have larger QA departments.

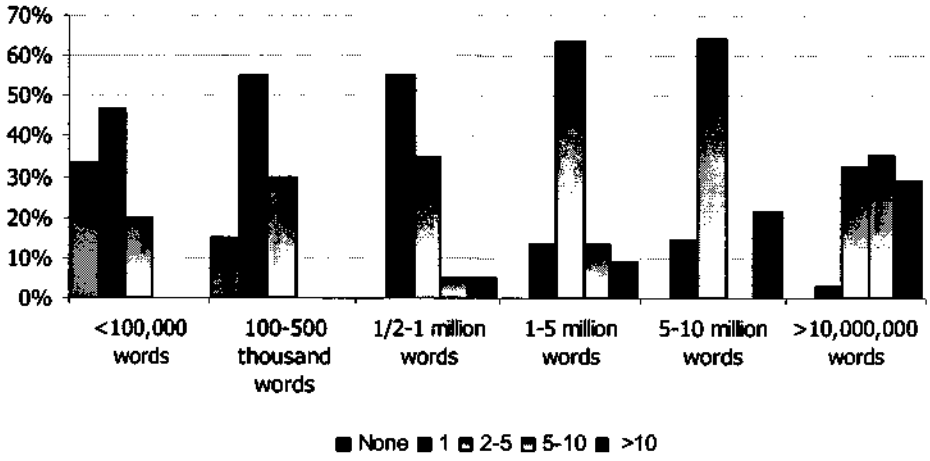

Figure 15. Amount of people responsible for quality assurance according to translation volumes

Almost 1/3 of the respondents reported they apply quality assurance procedures at the end of each translation stage (translation, editing, proofreading), and companies which adhere to this approach are mostly large and medium-sized. Among small companies, the most popular approach is to apply QA only at the final stage before delivering the files to the client. This is the second popular approach for all companies, selected also by 1/4 of medium-sized companies and 1/8 of large ones. 30% of respondents reported they apply QA procedures to source files as well as to final ones which often helps to avoid repeated translation errors, especially when there is a significant number of target languages. Over 5% of the respondent companies, mostly large ones, don't apply any QA procedures in-house and outsource them either to their localisation vendors or to third-party companies. Other QA methods (selected by 4.62% of the respondents) include spot-check of final files and terminology check, while the most popular response in this category was "it depends on a project".

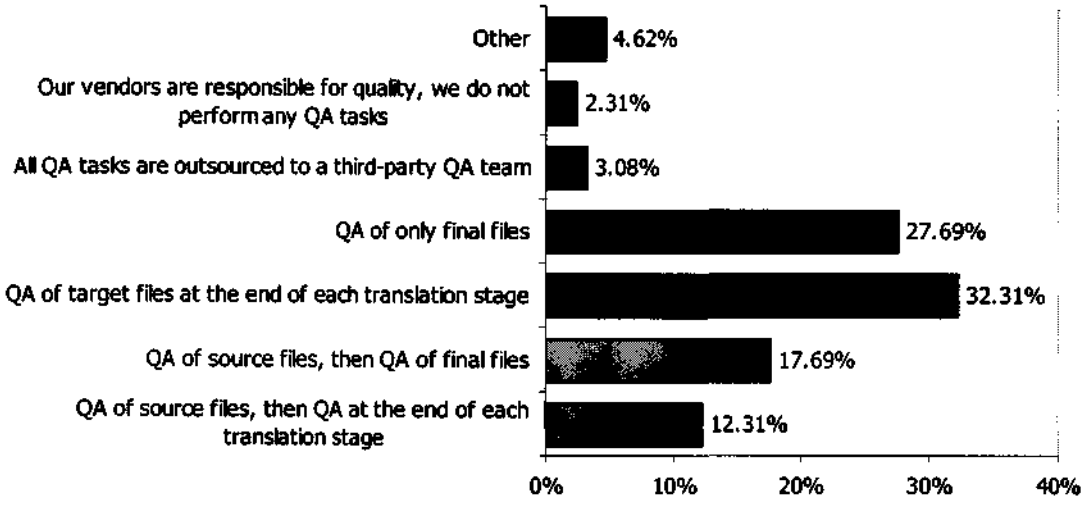

Figure 16. Approaches to quality assurance

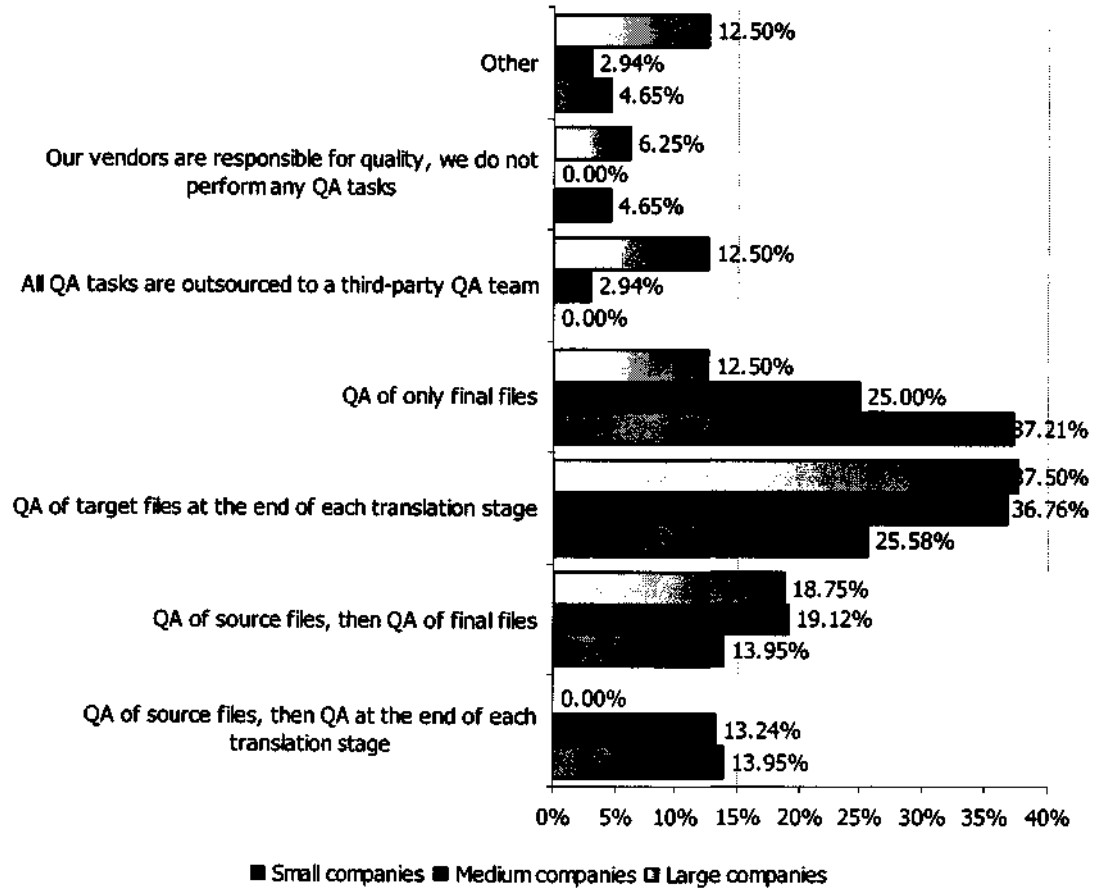

Figure 17. Approaches to quality assurance according to company size

Main reasons to avoid internal quality assurance are time constraints (for large companies) and absence of QA requirements from the customers (for small ones). The third main reason to exclude QA from the project lifecycle is that companies employ translators and translation teams with reliably high quality of their work. This approach is most popular among medium-sized companies. In general, project budgets seem to be the least important constraint for QA procedures. Only 6.52% of the respondents selected this option, more than a half of which represent large companies. Vast majority (10.87%) of respondents who selected other reasons in fact indicated that they always perform QA despite them.

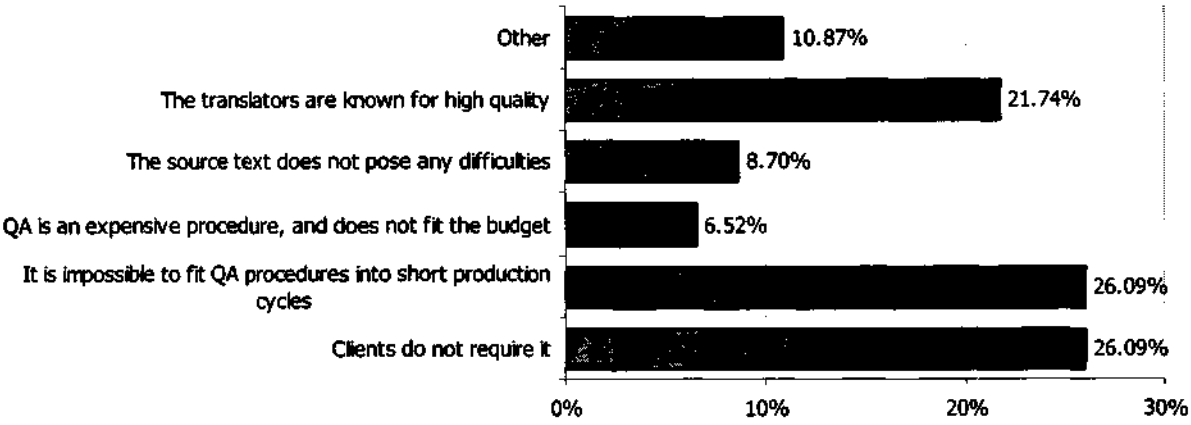

Figure 18. Reasons to avoid quality assurance stage

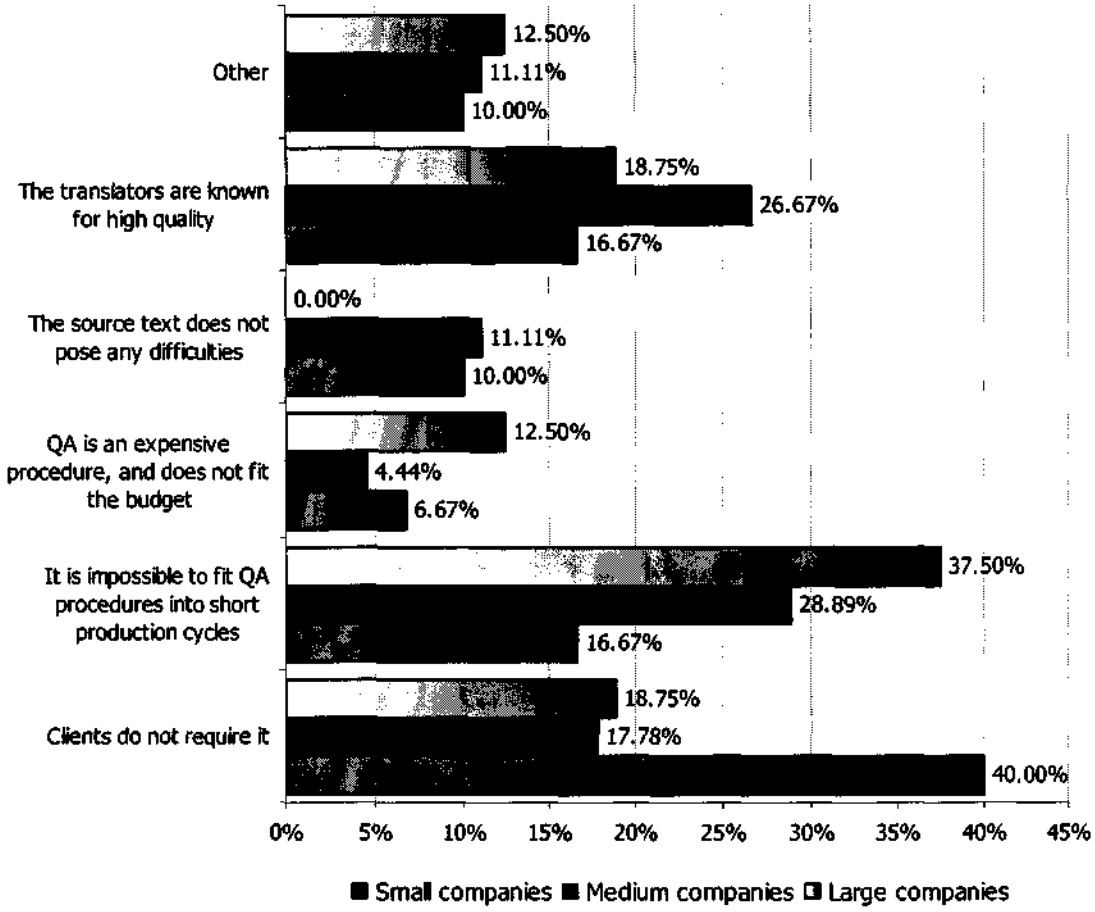

Figure 19. Reasons to avoid quality assurance stage according to company size

All types of quality checks are almost equally appreciated by companies of all sizes, and in general all companies tend to perform as thorough checks as possible. The least popular check is word-level consistency check which often is one of the most important, but on the other hand is very hard and time-consuming to implement to perform. Other types of checks which comprise only 1.57% included checks

according to SAE J2450 Translation Quality metric standard as well as spelling and grammar checks and proofreading. The latter, however, is out of scope of this paper because it cannot be automated.

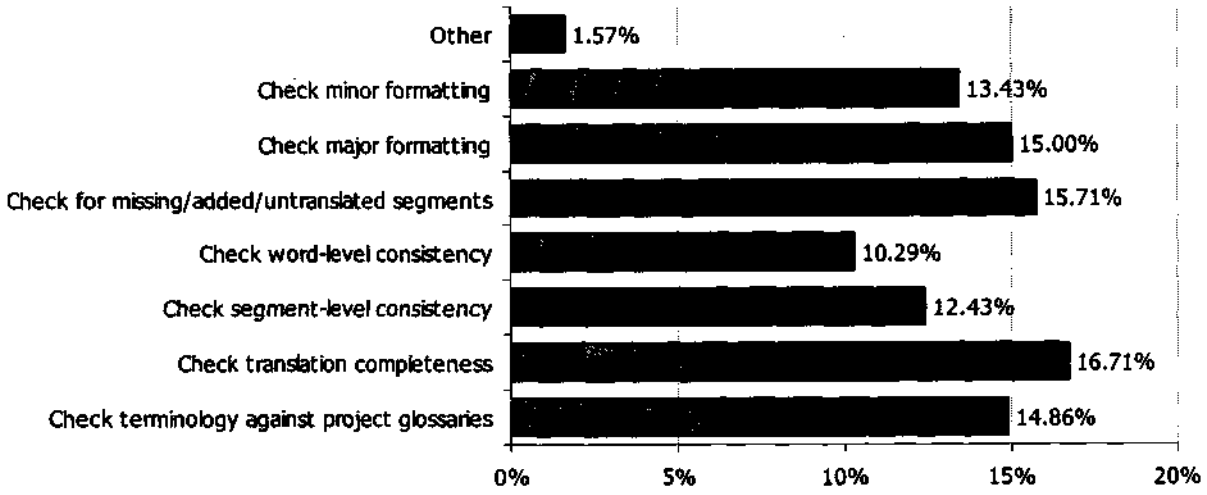

Figure 20. QA tasks performed

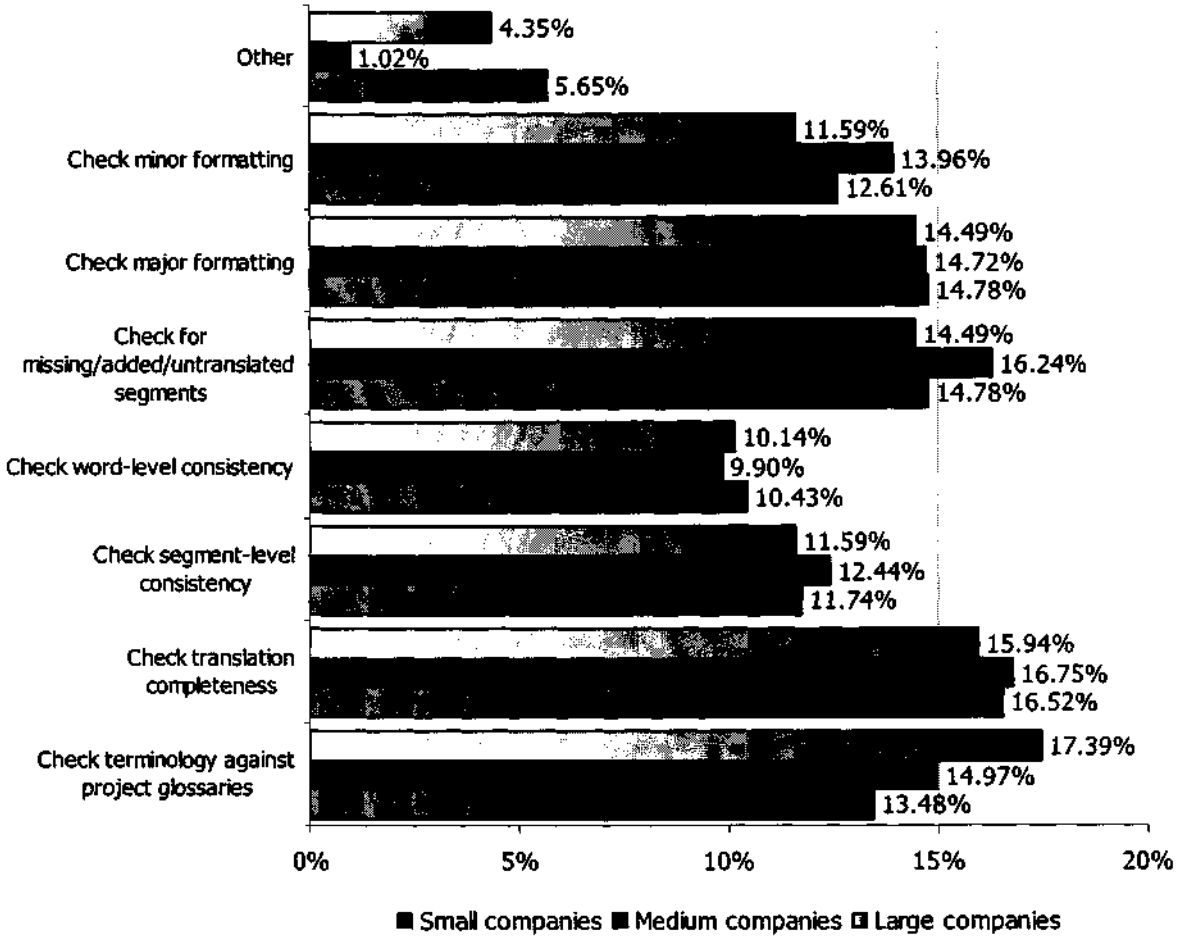

Figure 21. QA tasks performed according to company size

The most popular QA automation tools are those built into the most popular TM tools - Trados and SDLX. Those two are particular favourites of medium-sized companies. Using no QA automation tools is the third popular choice (the most popular one among small companies). For large companies, the second popular option is using proprietary tools. Almost 17% of large companies indicated they use their own QA automation tools.

Other tools specified by respondents included Ando tools, Microsoft Word spell-checker and SDL's ToolProof and HTML QA. Also SAE J2450 standard and LISA[12] QA model were mentioned which are not in fact QA automation tools, but metrics.

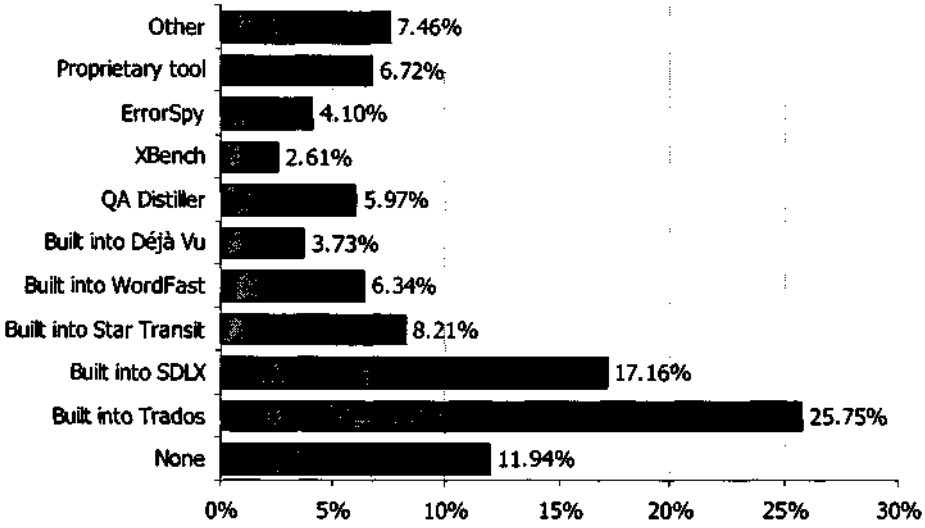

Figure 22. QA automation tools used

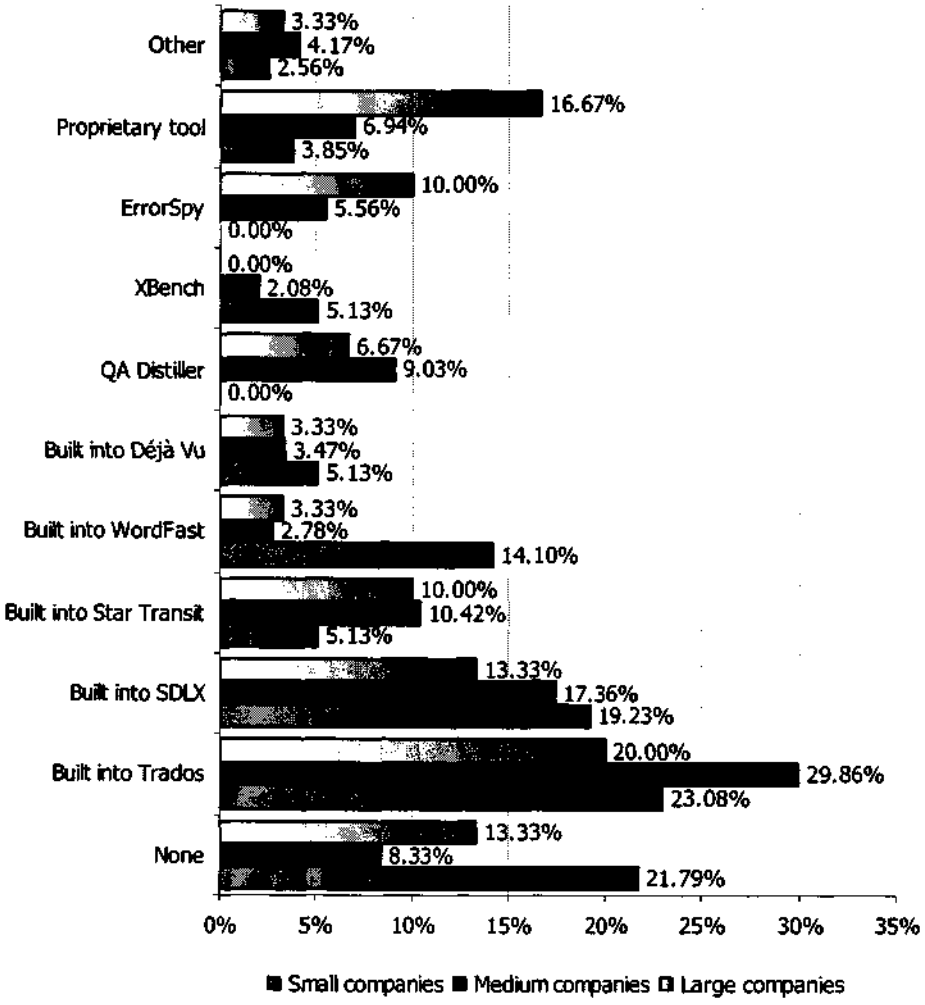

Figure 23, QA automation tools used according to company size

[12] The Localisation Industry Standards Association (http://www.lisa.org)

Standalone QA tools are much less popular. Whereas the most famous one, QA Distiller, accounts for around 6% of all responses, ErrorSpy got only 4.1% of the votes, and XBench, probably the least known tool to date, - 2.61%. This balance is different, however, in companies of different size. While large and medium-sized companies can afford rather expensive QA Distiller and ErrorSpy, small companies definitely prefer free XBench with its wide range of functions.

In general, only 17.17% of respondents indicated they use standalone QA tools while the rest of companies employ QA functionality built into TM tools. Additionally, large companies clearly tend to use standalone tools more, though 75% of them use built-ins.

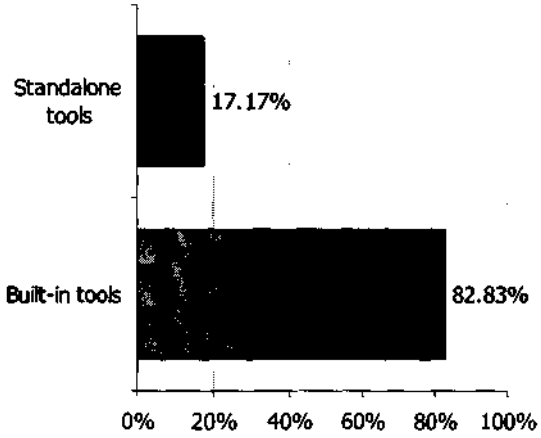

Figure 24. QA tools architecture

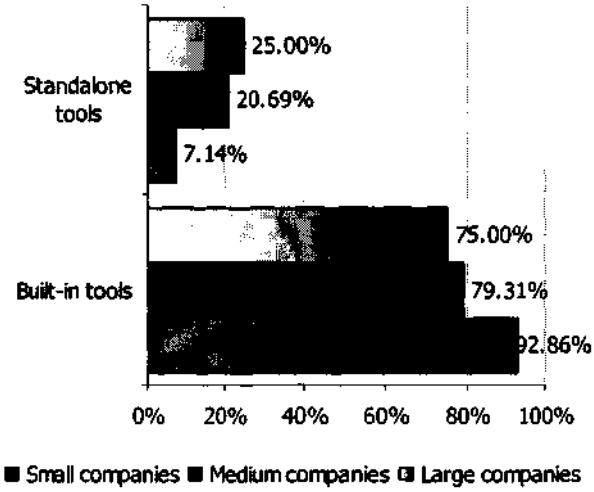

Figure 25. QA tools architecture according to company size

The most popular solution is using 2 or 3 QA tools which perfectly correlates with TM tools usage. Using only one QA tool is the most popular solution for small companies while those of medium size again tend to be as flexible as possible and employ as many QA tools as they can. Almost 5.5% of medium companies use more than 5 QA tools whereas no small or large company exceeds the limit of 5 tools.

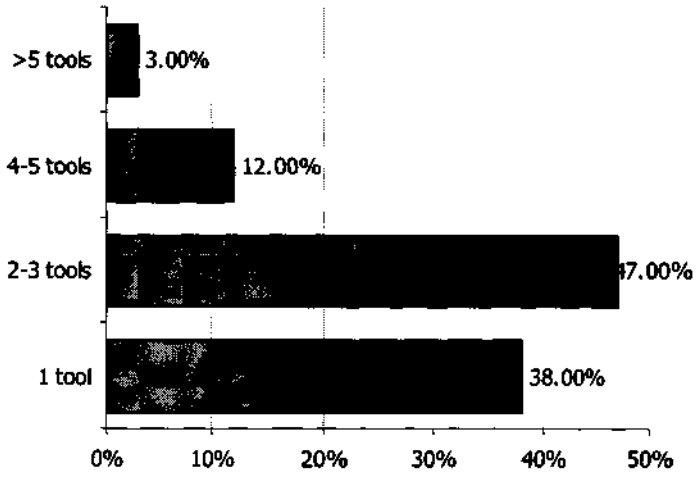

Figure 26. Number of QA automation tools used

The majority of the respondents prove to be rather experienced QA tools users. The number of companies using such tools for more than 2 years exceeds 72% with almost 70% of large companies using these tools for over 5 years. Apparently small companies are rather new to such tools, with 25% of those only starting to employ some kind of QA automation.

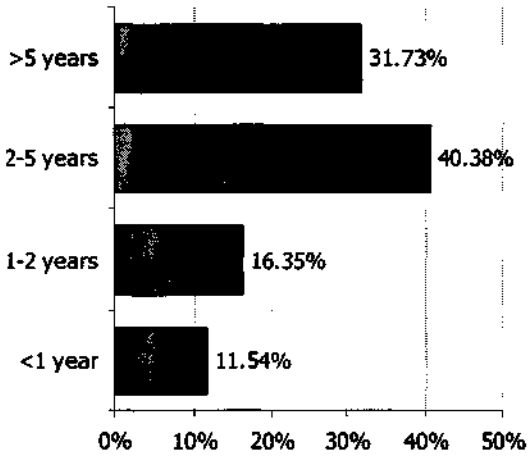

Figure 27. Duration of QA tools usage

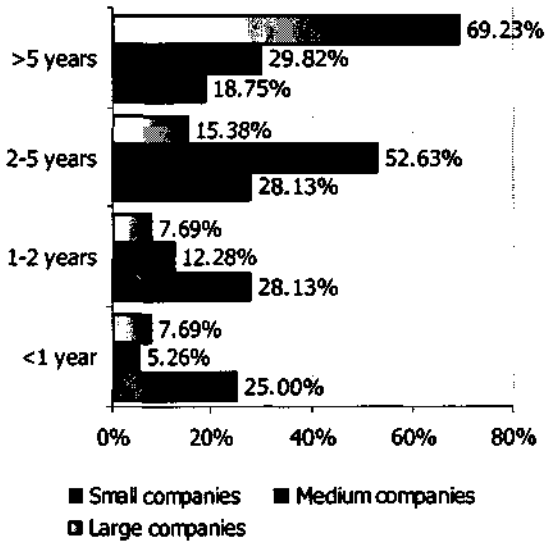

Figure 28. Duration of QA tools usage according to company size

Most of the respondents confirmed they are experienced enough in using their QA automation tools. Only 14.13% of all the respondents indicated they use default tool configuration and never tried to change it. Almost 20% of the respondents at least tried different configurations of their tools whereas more than 65% of them have a good knowledge of their tools capabilities and configuration options. This percentage is a little lower for small companies which may be easily explained by the fact that small companies have less experience in using such tools. It is also worth to note that almost 2/3 of the respondents use their tool's default configuration, and half of those are usually satisfied with the default configuration of QA tools.

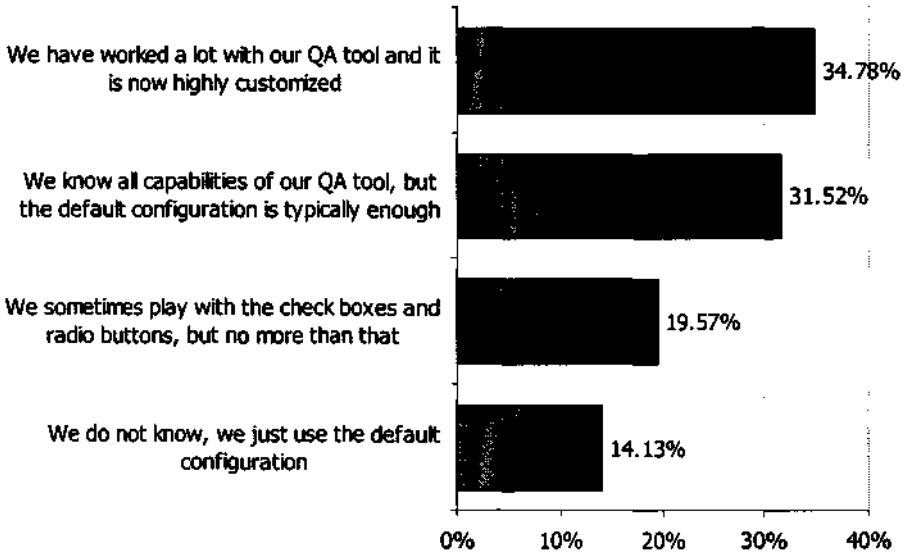

Figure 29. QA tool familiarity

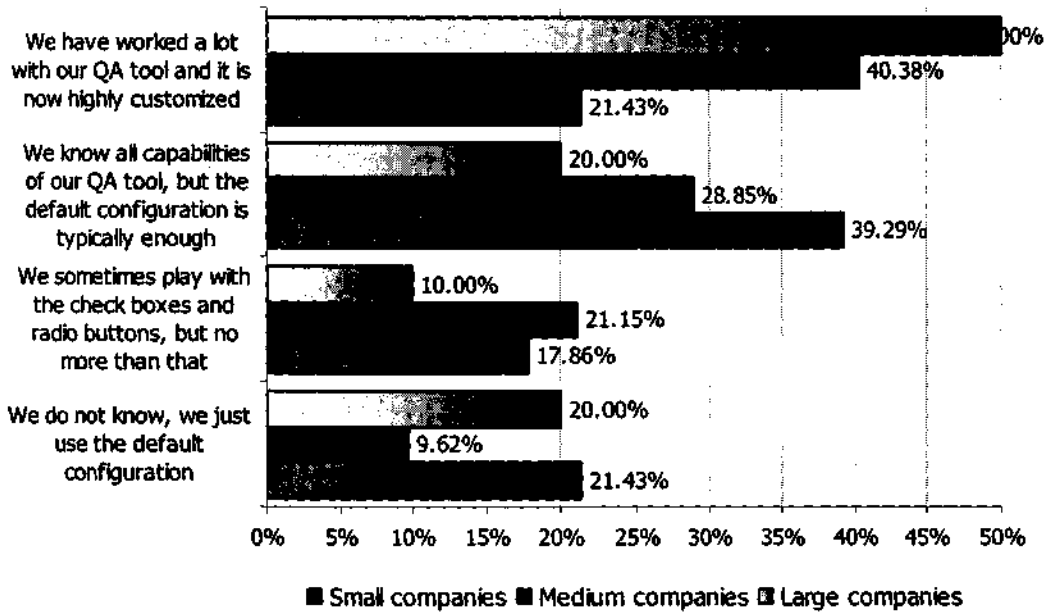

Figure 30. QA tool familiarity according to company size

Among 32 companies that don't use any QA tools, 20 were just not aware of the fact that such tools existed on the market. It is the most popular reason for non-using such tools selected by 25% of the respondents in total. The percentage of companies that are not aware of QA automation tools reaches almost 38% among small companies, 20% for medium-sized companies and is the lowest (however, still high enough with almost 12%) among large companies. The second popular reason is using other QA methods which are most often non-automated (such as proofreading) and/or proprietary methods/utilities. Lack of time/resources is the third reason for not using QA automation tools. Almost 23% of medium-sized companies and 17.65% of large ones (with no small companies at all) have indicated it to be one of the most important reasons for them. One of other most popular reasons for small companies is the price of the existing tools. Over 10% of small companies indicated that they just could not afford purchasing the tool they would like to use. For large companies, the situation is a bit different. Apparently they often use numerous proprietary file formats, so almost 12% of large companies specified that the existing QA tools were not suitable for the file formats they worked with. Other reasons for not using QA automation tools included planning to buy some utility or the fact that no tools were capable to proofread translations.

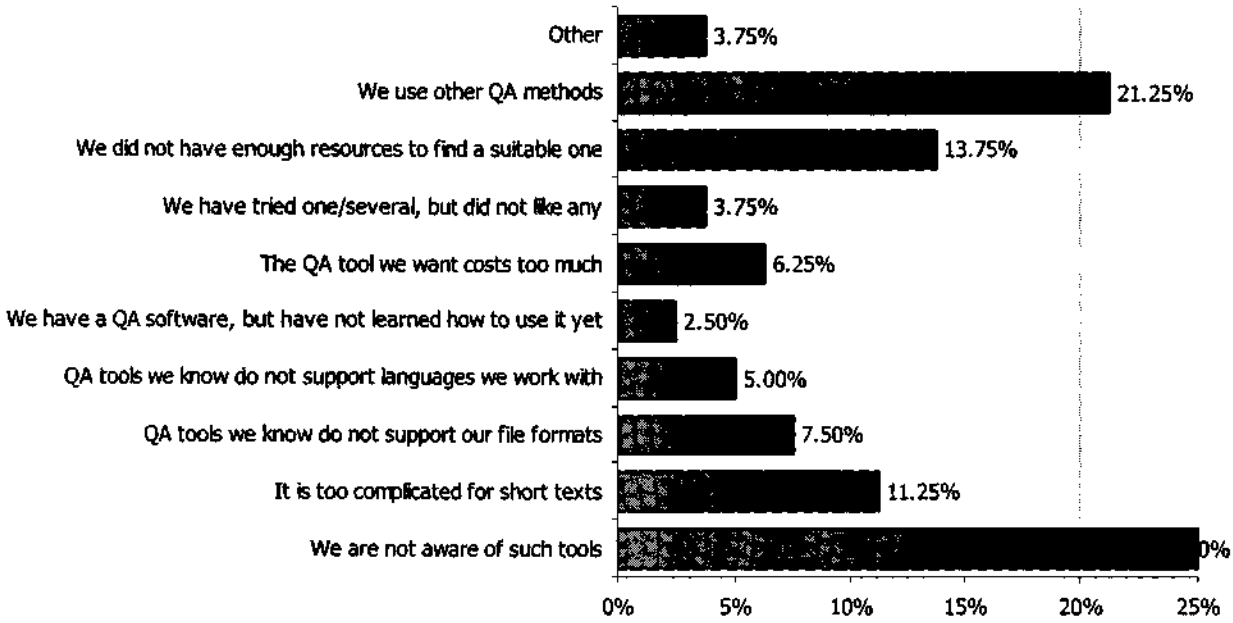

Figure 31. Reasons for not using QA tools

In general, whereas the vast majority of all companies have no reasons to avoid using QA automation tools, some of them (mainly large companies), still have several. E.g. the percentage of small and medium-sized companies which selected more than 2 reasons is under 4% while among large companies this figure reaches almost 15%.

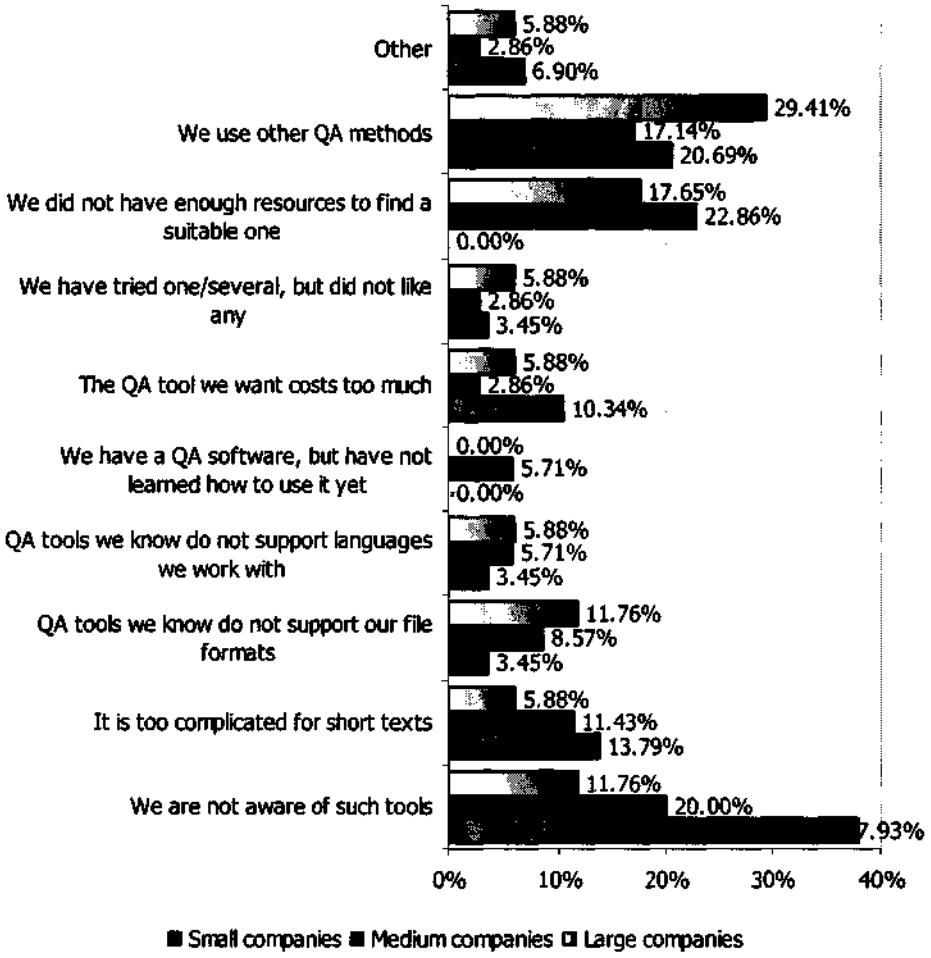

Figure 32. Reasons for not using QA tools according to company size

It was especially interesting to reveal what languages, if any, are most hard to check using automated QA tools. Almost 35% of the respondents specified they did not apply QA automation tools to CJK[13] languages while 24.29% of respondents omit checking languages with Cyrillic script, and only 17.14% of QA tool users indicated they do not apply such tools to right-to-left languages. This in fact hardly correlates with our benchmark results which will be discussed below.

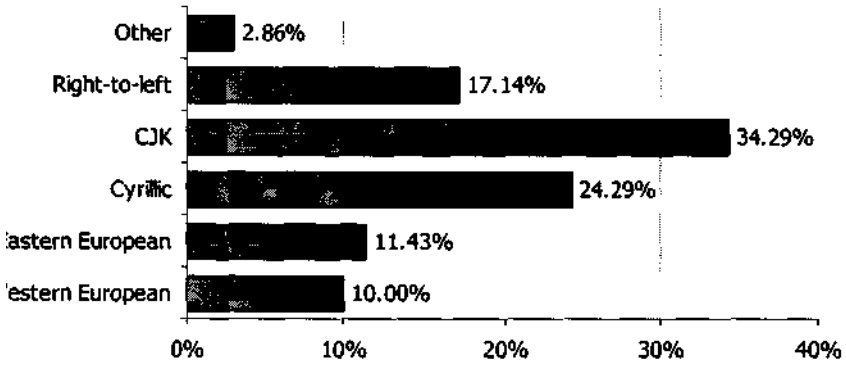

Figure 33. Languages not checked using automation tools

[13] Chinese, Japanese, Korean

2.86% of other languages which respondents find hard to check automatically include Indian and African languages which were in fact beyond our benchmark. However, taking into account that even CJK and right-to-left languages that are very popular target languages on today's translation market are not supported well by QA automation tools, we can assume that Indian and African languages support is even weaker.

QA Automation Tools Evaluation and Expectations

The survey also attempted to reveal the reasons for some reluctance to use QA automation tools. Many respondents indicated at least one difficulty one can encounter integrating a QA automation tool in the established work processes, and more than 30% of the respondents encountered several difficulties. Necessity to learn new software and to change people thinking accounts for more than a half of all difficulties, whereas lack of support for some particular languages and file formats comprised more than 40%. This percentage does not vary significantly depending on company size.

6.33% of other difficulties mentioned include time constraints, too many false positives generated by QA automation tools and numerous instructions for different projects that are hard to follow using the tools.

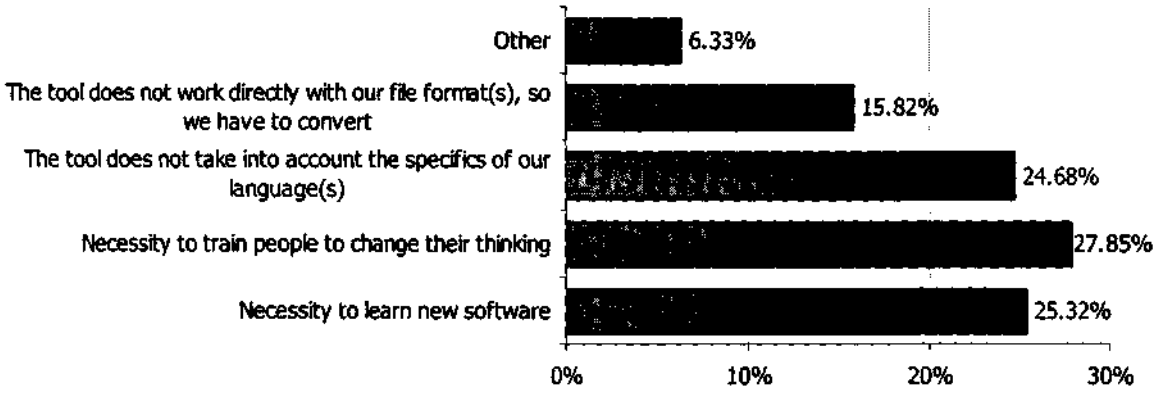

Figure 34. Difficulties encountered using QA automation tools

Almost 40% of the respondents confirm that QA automation tools increase error detection level, and almost 35% are sure they increase productivity as well. Almost 1/4 of QA tool users indicate they help to avoid monotonous work. Other reasons include responses like "It helps to stay in this business" and "Clients react positively to this offering". On the other hand, almost the same number of QA automation tools users complain about their weak points. Numerous false positives are the main complaint selected by more than 35% of the respondents. Over 28% of them are not satisfied with limited capabilities of such tools (for large companies, this percentage exceeds 45%), and more than 32% of users are sure that human eyes are still much more reliable than software programs.

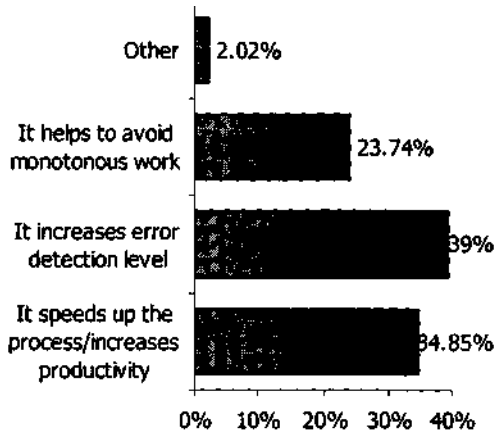

Figure 35. Positives about QA automation

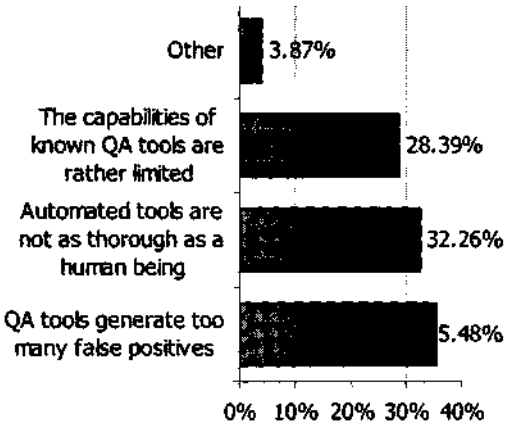

Figure 36. Negative comments about QA automation

Most of the respondents are however quite satisfied with the QA tools they use. The highest level of satisfaction refers to efficiency in speed which confirms that QA automation tools help save time and speed up the processes. The survey also confirms that the existing tools are not usually hard to learn and use as well as that in general they do not hang up often. Features that the users are still not satisfied with are efficiency in error detection (as we said before, users indicated that generating too many false positives is the main drawback of such tools), customer support, adaptability and reportability. If we assume value for money to be the average (although subjective) satisfaction factor, we may say that in general 20% of respondents are not satisfied with the utilities they use.

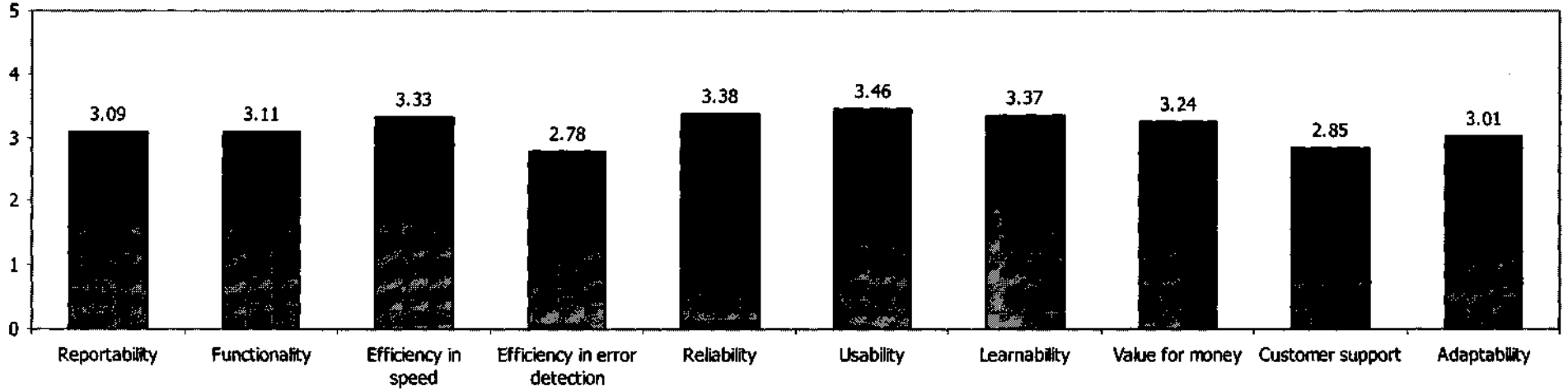

Figure 37. Overall QA tools evaluation[14]

[14] Criteria for this evaluation were selected based on the quality metrics proposed in the EAGLES framework.[2] King, Margaret (1997) "Evaluation Design: The EAGLES framework" Konvens 97.

The graph below represents weighted evaluation of QA tool features according to users experience in years of using the tools. We can observe that user satisfaction with all features generally increases with time. While 1st year users' marks start with 2 and hardly ever reach 3, only two marks of more experienced users did not reach the limit of 3, and more experience users have, the higher the marks are. This generally means that users get adapted to the tools they use and are inclined to be more loyal to them with time.

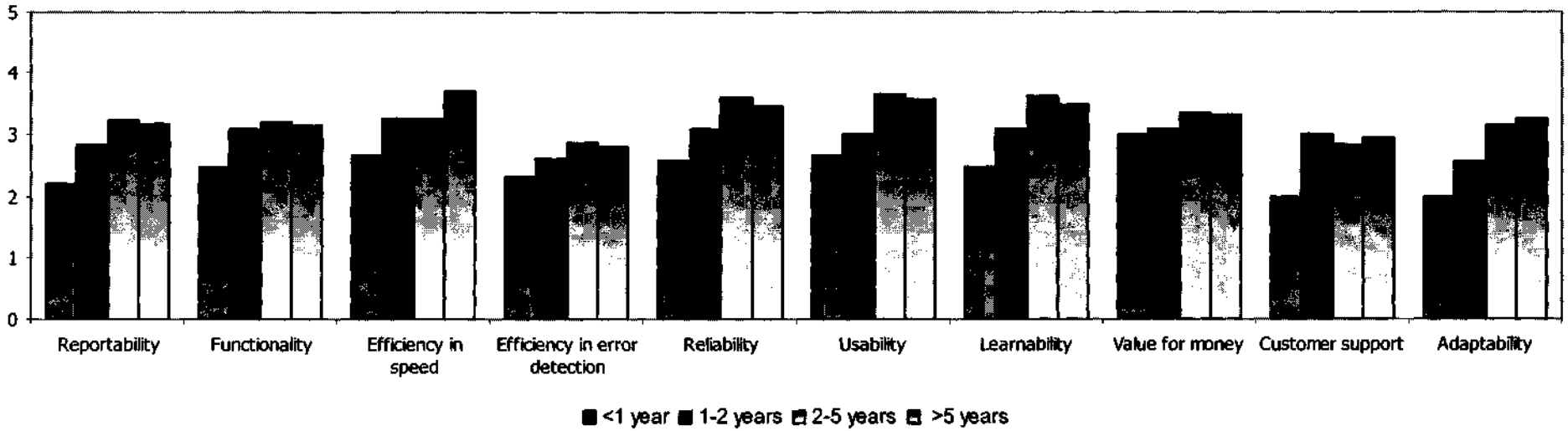

Figure 38. QA tools evaluation by companies according to their experience in QA tools

According to individual QA tools evaluation, WordFast, XBench and Déjà Vu, with all marks (except for customer support) between 3 and 4 seem to be the most effective tools. However, it is necessary to highlight that only two respondents evaluated XBench and Déjà Vu and 5 respondents evaluated WordFast, so those evaluations are not reliable enough. Star Transit's evaluation with all 3 marks equal to "Quite satisfied" is also unreliable.

Unfortunately as the amount of assessments we received for some tools (namely Star Transit, XBench and Déjà Vu) is rather low, the evaluation of those tools cannot be considered as statistically reliable.

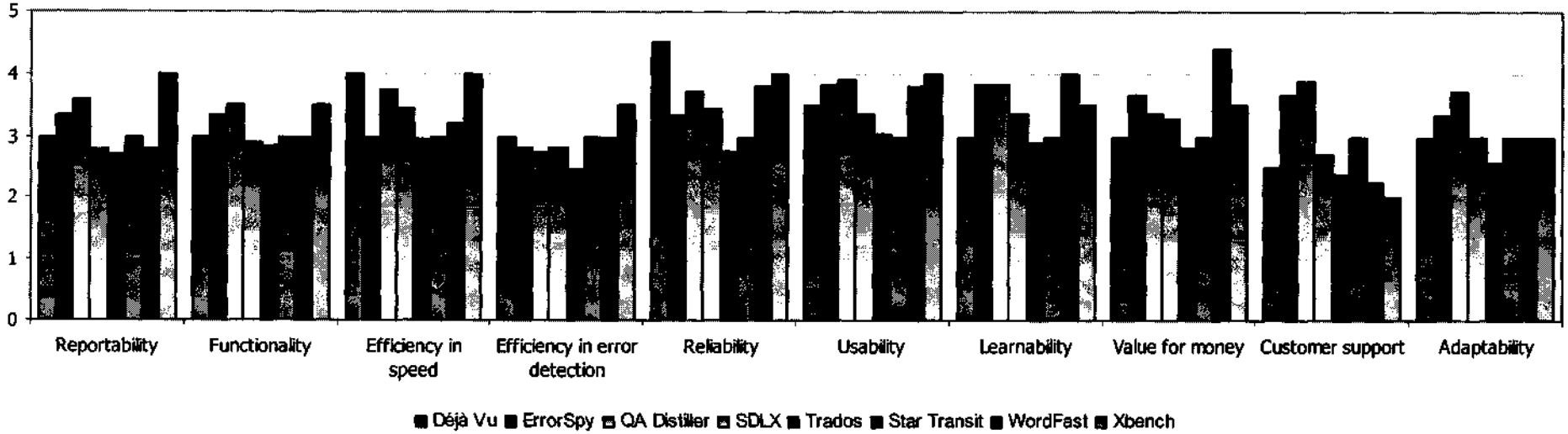

Figure 39. Comparative QA tools evaluation

The following graph provides the idea of what tools were in fact evaluated by the respondents. Again, two most popular tools were Trados and SDLX, the third most popular one was QA Distiller. On the other hand, only a few number of respondents evaluated Star Transit, Déjà Vu and XBench.

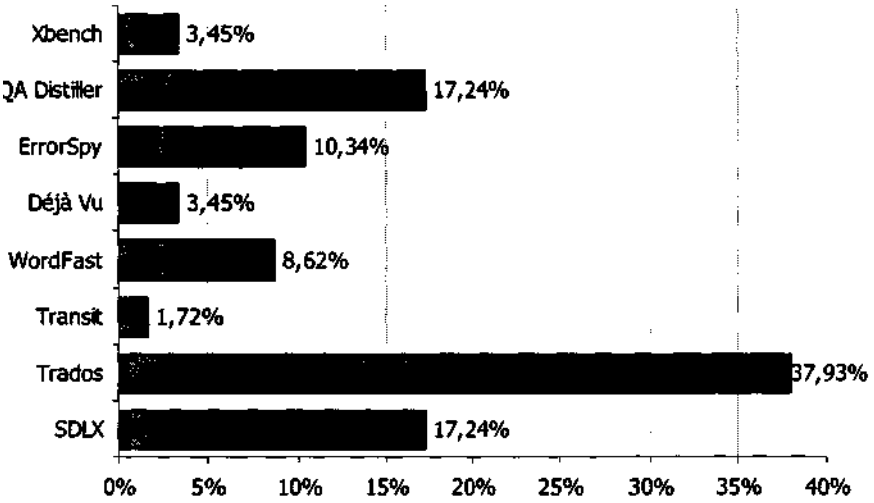

Figure 40, QA tools evaluated

Last, but not least, the survey's goal was also to identify what users need and expect from the QA tools.

According to the figure below, more than 60% of respondents would like to see a fully functional standalone application that supports a lot of input file formats whereas over 26% of users prefer to see QA tool as a part of their TM tool. 1/3 of users would like QA tools to have extended functionality not only limited by QA features (something like XBench already offers). One respondent expressed a desire to have both independent and plug-in tool, and another one indicated that the architecture of the tool was not important as long as the tool was affordable.

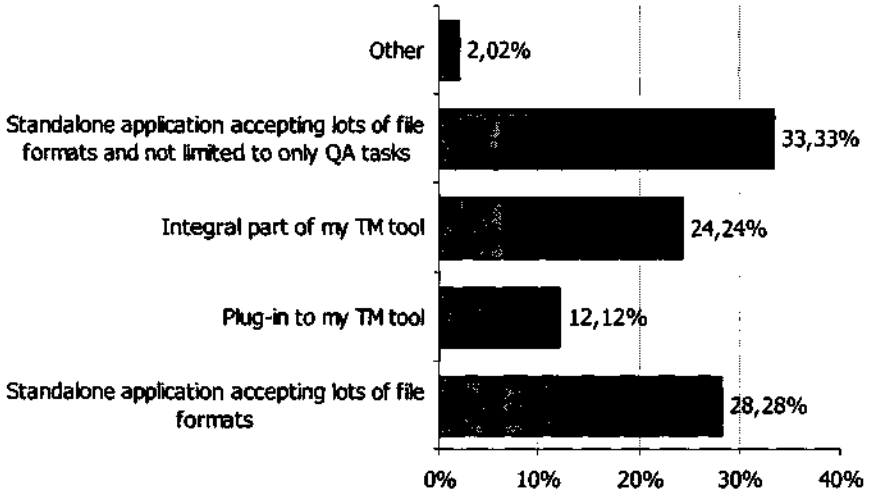

Figure 41. Preferred QA tool architecture

The next figure shows weighted evaluation of desired features of a QA tool where 4 means the most desired feature and 0 corresponds to the less desired one. Integrated spell-checking, ability to maintain client-specific and project-specific checklists and ability to check consistency at the level of individual words or passages in addition to segment level are the most desirable (but not easy to implement) features. Support for more file formats and the ability to compare translations with the project TM "on the fly" are also an obvious necessity.

The two least desired features are the ability to run on multiple platforms (and this is only natural if we recall that most of the respondents work either under Windows or under Windows plus some other OS) and the possibility to switch interface language (which is also predictable because all the respondents are localisation professionals, which means they all are rather fluent in English).

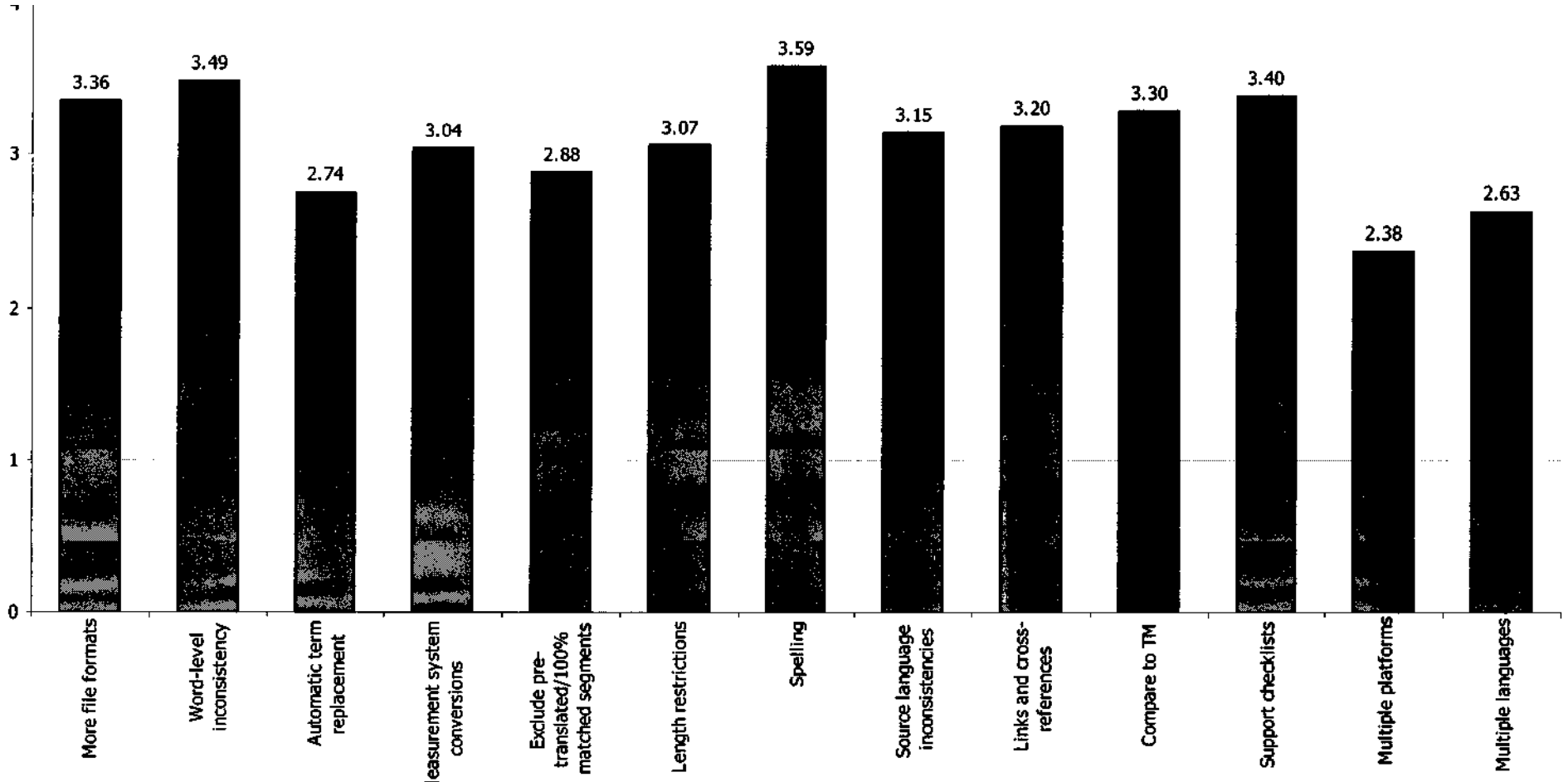

Figure 42. Desired QA tool functionality

Benchmarking available quality assurance tools

In order to verily QA tools capabilities and features as well as to measure error probability[15], we created a short test file consisting of 22 sentences; each of them included an error of one of the types specified above. The file was saved in HTML format and translated into 8 target languages using Trados and TagEditor (TTX format). A glossary with only one term was created for each language to evaluate terminology check abilities. Target languages were selected based on the script and included all common script types: right-to-left (Arabic, Farsi, Hebrew), CJK (Chinese Traditional), Cyrillic (Russian), Eastern European (Polish and Czech) and Western European (French). Rare languages with less common scripts were out of the scope of this research.

Additionally a single huge English-Russian file was used to measure run time. However, all tools required almost equal time to run through those tests[16], therefore this result was not included into the report.

The following tools were benchmarked:

- **Déjà Vu X Workgroup** version 7.5.302, manufacturer: ATRIL (http://www.atril.com):
- **ErrorSpy** 4.0, build 001, manufacturer D.O.G. Dokumentation ohne Grenzen (http://www.dog-gmbh.de):
- **QA Distiller™** 6.0.0 (build 188) (http://www.qa-distiller.com), manufacturer: Yamagata Europe (http://www.yamagata-europe.com):
- **SDLX 2007 QA Check,** build 7014, manufacturer: SDL International (http://www.sdl.com):
- **Star Transit XV Professional,** version 3.1 SP 21 Build 617, manufacturer: Star AG (http://www.star-solutions.net):
- **Trados QA Checker** 2.0, plug-in to SDL Trados 2007, manufacturer: SDL International (http://www.sdl.com):
- **Wordfast version** 5.51t3, manufacturer: Wordfast LLC (http://www.wordfast.net):
- **XBench** 2.7 (build 0.183), manufacturer ApSIC (http://www.apsic.com).

The benchmark revealed that all checked tools generate false terminology errors by checking untranslated and empty segments, while it is rather obvious that segments of those types may not include correct terms. Another common drawback for almost all the tools (with a few exceptions) is their poor ability to handle multilingual projects as well as right-to-left languages. In addition to problems of glossary application in multilingual projects, many tools check translation consistency between different languages and generate false positives because apparently the same segment is translated differently into different languages.

QA capabilities revealed

The table below summarizes QA tool pricing and file formats it may check as well as supported glossary, TM and report formats. It clearly shows that free XBench is an absolute champion with regard to file formats.

[15] It must be noted that grading the tools and selecting the best one in no case was the aim of the benchmark.

[16] Some of the tools perform the checks in a non-batch mode, so it was impossible to adequately measure test run time for them.

	Déjà Vu	SDLX QA Check	Star Transit	Trados QA Checker	Wordfast	ErrorSpy	QA Distiller	XBench
Software organisation	Plug-in	Plug-in	Plug-in	Plug-in	Plug-in	Standalone	Standalone	Standalone
Pricing	€490 €990 €2250/ €1490[17]	€795	Unknown	€795[18]	€250	€249 €490	€249 €1000€ €2500	Free
Regular Expressions/ User Checklists	SQL queries	Yes	No	Yes	Macros	Yes	Yes	Yes
Input File Formats	Internal	ITD (internal)	Internal	TTX (internal)	Internal	Trados Delimited Trados TM Trados TTX Transit	TTX RTF TMX	Delimited All Trados SDLX Star Transit IBM TM Wordfast Microsoft Glossary XLIFF TBX Mac OS X Glossary
Glossary Formats	Tab-delimited Excel Access internal	TDB	internal	MultiTerm	Plain text	XLS TXT MultiTerm	DICT (internal) TBX	Delimited Wordfast Glossary Microsoft Glossary TBX MultiTerm Mac OS X Glossary

Table 3. Pricing and formats supported by different tools[19]

[17] If more than one price is indicated, different prices refer to different versions of the application.

[18] The price indicated is for freelance version; to check terminology MultiTerm with an additional license is also required.

	Déjà Vu	SDLX QA Check	Star Transit	Trados QA Checker	Wordfast	ErrorSpy	QA Distiller	XBench
TM Database Formats	Internal TMX	No	TMX	Trados	Internal TMX	No	No	TMX Trados TXT IBM TM Wordfast Memory
Report Formats	No	HTML	No	HTML	RTF	HTML	HTML\ XLS plain text	TXT, HTML XLS, XML

Table 3 (continued). Pricing and formats supported by different tools

The following table summarizes semi-default checks supported by different tools. We called them semi-default because all checks enabled by checkboxes were included into the table while sometimes some checkboxes are not set in the default configuration of a tool.

The table shows that QA Distiller is the most comprehensive tool supporting almost all types of checks without any additional customisation, and ErrorSpy and SDLX QA Check follow it.

Check	Déjà Vu	SDLX QA Check	Star Transit	Trados QA Checker	Wordfast	ErrorSpy	QA Distiller	XBench
Segment-Level Checks								
Empty Translations	✓	✓		✓		✓	✓	✓
Forgotten Translations		✓		✓		✓	✓	
Skipped Translations				✓			✓	
Partial Translations		✓				✓	✓	
Incomplete Translations		✓		✓		✓	✓	
Corrupt Characters		✓		✓			✓	
Inconsistent Sentence Count		✓						
Inconsistency								

[19] For the purpose of this paper, checks implemented via macros, SQL queries and regular expressions are excluded from the report and considered to be non-supported.

Source Inconsistency							✓	✓
Target Inconsistency	✓	✓		✓		✓	✓	✓

Table 4. Checks supported by different QA tools

Check	Déjà Vu	SDLX QA Check	Star Transit	Trados QA Checker	Wordfast	ErrorSpy	QA Distiller	XBench
Punctuation								
Punctuation at the End of Segments		✓		✓		✓	✓	
Spaces Before Punctuation		✓				✓	✓	
Double Spacing		✓		✓	✓	✓	✓	
Double Dots		✓		✓		✓	✓	
Double Punctuation		✓				✓	✓	
Quotation Marks						✓	✓	
Brackets and Parentheses				✓		✓	✓	
Numbering								
Number Values	✓		✓	✓	✓	✓	✓	✓
Number Formatting			✓			✓	✓	
Measurement Unit Conversion							✓	
Digit to Text Conversion						✓	✓	
Terminology								
Project Glossaries Adherence	✓	✓	✓		✓	✓	✓	✓
Identical Untranslatables						✓	✓	
Tags								
Identical Tags	✓		✓	✓		✓	✓	

Table 4 (continued). Checks supported by different QA tools

Déjà Vu

Number of checks supported. The number of default checks is rather limited; however, the application supports custom SQL queries which most probably allows for extending the amount of possible checks and further customisation.

Multilingual project support. While checking several files translated into different languages, DejaVu applies the TM and glossary for the first language to all files no matter what their target language is.

Right-to-left language support For Arabic and Farsi it reported terminology errors even where the translated term could be easily found using Find feature.

Reportability. The style of error indication is probably convenient for a translator who want to check the translation "on the fly", but is rather inconvenient for a dedicated quality assurance department.

Conclusion. In general, this is one of few tools whose declared capabilities are close to real ones. All in all, the tool is only suitable for checking its native files. Although it supports other most common formats including Trados, SDLX and Star Transit, conversion is quite time-consuming and is not in general worth it.

SDLX QA Check

Supported checks. Basic set of checks performed is also rather limited. A user can extend and customise it using regular expressions; however, regular expressions are often beyond the qualification of a QA manager.

This tool does not check number values and does not check number formatting, double punctuation marks and brackets unless you set up a regular expression. It also does not check tags, and though it is hard enough to change tags in SDLX, TTX files converted to SDLX format may contain corrupt tags which won't be detected.

Skipped translations are not converted from TTX files and therefore are also not found.

SDLX does not allow specifying Chinese full stops as a valid punctuation mark.

False positives. QA Check generates false positives for forgotten translations (counted as partial translations as well). Also many false positives are generated for partial/incomplete translations (they are not differentiated in SDLX) because incompleteness is determined only by translation length, not taking into account the number of sequential source words found in target segments.

Multilingual project support. As many other tools, QA Check Checks translation consistency between different languages.

Right-to-left language support. QA Check displays those left-to-right which hinders work with files and leads to reporting non-existing terminology errors.

Conclusion. With additional customisation, this tool is quite a good solution for SDLX users that do not want to involve additional standalone tools into their work processes.

Star Transit

Supported checks. Star Transit employs the most limited number of checks without any further customisation. Available customisation is provided via fixed value lists and does not allow to add e.g. custom delimiters which in our case was necessary for Farsi[20].

False positives. Due to the limited number of checks supported by Star Transit it generates one of the lowest number of false positives.

File import. Import of TTX files is not correct enough; tags are represented in an unusual manner which hinders work with files.

Right-to-left language support. This tool proved to be surprisingly good at checking terminology in right-to-left languages. It also showed probably the best handling of right-to-left languages in general.

Reportability. The tool does not provide any reports, all errors need to be corrected "on the fly".

Conclusion. In general, the real functionality of the tool is closest to the claimed one; however, it is too limited.

[20] In Farsi, a slash (/) is used as a decimal delimiter which is not supported by default.

SDL Trados QA Checker

Supported checks. Basic set of checks performed by SDL Trados QA Checker is rather extensive compared to other plug-in tools and may be extended using regular expressions. This tool does not allow to specify Chinese full stops[21] as valid punctuation marks. Moreover, Arabic and even Easter European characters cannot be included into forbidden characters list. It also does not check quotation marks and number formatting.

False positives. Unlike all other tools, it generates false positives by counting skipped and empty segments as incomplete ones.

Conclusion. In general, the tool is good enough for translators who work in Trados TagEditor, but may be hard to employ in dedicated quality assurance departments where batch processing of mono- and multilingual projects is normally required.

Wordfast

Supported checks. The amount of checks supported is rather limited, but may be extended using custom macros.

File import. Wordfast determines HTML files by <html> tag at the beginning, but not by the real content, whereas in real life this tag may often be omitted.

Multilingual project support We failed to make it check terminology against the correct glossary. After checking the Arabic test file, Wordfast continued to apply Arabic glossary to the rest of the languages despite of numerous setup changes, glossary recreation, deletion etc. Even when there was no Arabic glossary existing on the computer, Wordfast still reported Arabic terminology errors. It might have got much better scores if it used correct glossaries.

Right-to-left language support. It doesn't properly handle right-to-left languages and just like all other tools reports terminology errors in untranslated segments.

Conclusion. As many other plug-in tools, WordFast provides quite a good solution for those who select it as a TM tool and do not want to implement a standalone QA tool.

ErrorSpy

Supported checks. The total set of supported checks is quite extensive with some specifics listed below. ErrorSpy includes presets for some languages, but they are sometimes incorrect (e.g. incorrect quotation marks for French). It does not support Chinese Traditional as well as right-to-left languages without additional customisation[22] and does not support specifying more than one set of quotation marks in case of nesting. Although it allows to specify decimal and thousand separators to check number formatting, we failed to make it check it. In fact it only reported unmatched figures.

File import. The tool cannot check for skipped segments because it does not import skipped segments at all. This is not convenient if you need to locate any segments that were left untranslated.

False positives. ErrorSpy reported "space required after punctuation mark" errors even if the corresponding checkbox is deselected.

Right-to-left language support. No support by default; however, the tool allows to create new languages. For Arabic, it reports Latin characters to be punctuation marks although they were listed as valid characters in language configuration.

Additional observations. English user interface contains translation errors (for example, one of the checkboxes reads: "spaces that require a space before").

Another drawback is that ErrorSpy sometimes corrupts the first letter of language names.

This tool does not remember the directory it recently worked with. It is quite inconvenient when you work in a non-default directory.

For some reason, more and more interface elements switch to German with each test run. The interface gets back to English after restart.

---

[21] Full stops in traditional Chinese are double-byte characters that look differently compared to conventional ones ("$_0$" instead of ".")

[22] To add languages that are not preset, you need to open the database in Excel, add languages there and then configure them in ErrorSpy.

The tool crashed on each attempt to check Russian. This may be a problem of this particular installation, but may also be a problem of the whole release. It must be noted, however, that version 3 ran smoothly on the same computer.

Conclusion. Although the tool significantly improved compared to its version 3, the first build seems to be quite unstable. However, with such rapid progress, this tool is rather promising.

QA Distiller

Supported checks. This tools supports the widest number of possible checks which still may be extended using regular expressions. However, a serious drawback is that it does not check tags identity. The test file included a hyperlink which was intentionally changed in all translations; however, QA Distiller ignored it.

Additionally, we couldn't find a way to check untranslatables in Distiller. There are two places where you can set a list of untranslatable items, and our idea was that the tool should make sure they are identical in source and target text. However, Distiller did not report it missing untranslatable which existed in the test file.

Another weak point is number formatting check. Distiller only makes sure the number includes separators specified in parameters of the target language, but does not check the order of the separators. So, for example, it will consider 1,222.33 and 1.222,33 to be the same numbers with regard to number formatting.

Multilingual project support. In addition to reporting inconsistencies between different languages, Distiller also handles multilingual batches together with multilingual dictionaries in a strange way. For the first file it encounters, it tries to match all the glossary files in spite of the language indicated in the translated file and the glossary, which results in numerous "ignored terminology" errors. For the second target language, it tries to match it to all the glossaries until it finds the correct one. Then it perfectly matches the rest of the translated files with correct glossaries and doesn't generate error messages.

Right-to-left language support. While this is the most comprehensive QA tool so far, it definitely lacks right-to-left languages support. Sentences in those languages are still aligned left-to-right, and if a segment ends with non-Arabic/Farsi/Hebrew words or digits, QA Distiller often handles the end of the segment incorrectly which results in a false error message. Additionally, it reports terminology errors in almost every segment because of incorrect RTL text handling. If you open the same file in MS Word and do a simple search for the glossary term you will be able to locate it easily while Distiller insists the term translation is missing.

It does not support Farsi by default, so we had to define a new language which resulted in reporting too many corrupt characters (480 occurrences).

Additional observations. Inability to change error severity may also be considered as a disadvantage (at least it makes the software less flexible and customisable).

Conclusion. At the moment, this is the most comprehensive, yet rather expensive standalone solution on the market.

XBench

Supported checks. This tool officially does not support Unicode, and this is probably the main drawback of the application that eventually resulted in a rather high error level. For this reason, it does not support checking Arabic, Chinese, Farsi as well as Czech and Polish TTX files.

It does not have punctuation checks enabled by default; however, they are easy to enable via XBench rules which may significantly improve its error reporting in real life.

False positives. XBench reported corrupt characters for all non-Latin and non-Cyrillic characters.

Multilingual project support. Like many other QA tools, this one finds inconsistencies between translations into different languages and reports terminology errors in untranslated segments.

Conclusion. This tool is very new to the market, but probably one of the most promising tools to date. Its extensive file format support, additional functionality and extension capabilities together with the fact that the tool is currently free allow to suppose many companies, particularly small ones, may want to select it as their QA solution.

Error Level Comparison

A simple method was used to calculate the average error level. If *d* is the amount of all errors detected by a tool, f is a number of false positives and *n* is the number of non-detected errors then the total number of erroneous results of a tool will be *e=f+n.* The weighted error level may be calculated as

$$l=e*100/(d+n)$$

We have calculated two values for each tool's error level, one based on all check types being benchmarked, and another one based on only the checks supported by each particular tool. So, while the first value indicated the overall tool's usefulness, the second figure shows the tool's proximity to the functions it claims to support. In general, the lower is error level the better tool's behaviour is.

It has to be noted that benchmark was performed with minimum customisation for each tool, i.e. each tool was customised in such a way as if it was customised by a user with poor technical skills. This means that if additional checks had to be enabled by setting a checkbox, the checkbox was normally set when appropriate. However, if some checks had to be enabled via regular expressions/SQL queries and even rules (in case of XBench), no such checks were enabled.

The graph below shows that Wordfast is the least reliable tool when it comes to quality assurance checks. This may in fact be false. As it was mentioned before, we could not apply different language glossaries to appropriate test files because Wordfast always applied Arabic glossary instead. So, most errors that lead to such poor score were terminology ones.

QA Distiller has proven to be a more reliable tool and the most comprehensive one. On the other hand, it is the most expensive standalone tool available. It is followed by ErrorSpy (as well as with regard to supported check types). Déjà Vu which has the lowest error level according to its supported functionality seems to be the most adequate tool with Star Transit (whose functionality, though, is too limited) and Trados QA Checker coming in second. Taking into account Trados QA Checker's support for further customisation via regular expressions as well as a moderate price, we may consider it to be a good choice for freelance translators.

All in all, standalone tools tend to be more reliable compared to TM tool plug-ins. This fact together with their support for numerous file formats makes them a good choice for language service providers who employ dedicated QA teams. It also must be noted that XBench, the only free tool among those considered, most probably can be customised well enough to provide more check types and lower error level.

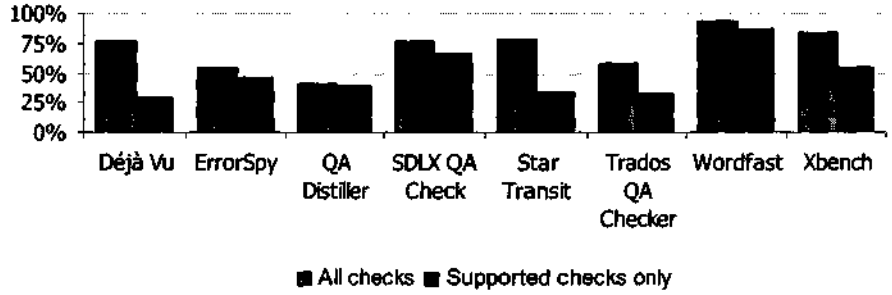

Figure 43. Summary error level by QA tools

Although right-to-left languages prove to be most troublesome, especially for terminology check, the error level shows almost no direct relation to the language type. While it is generally slightly higher for Farsi, the deviations in general are not significant.

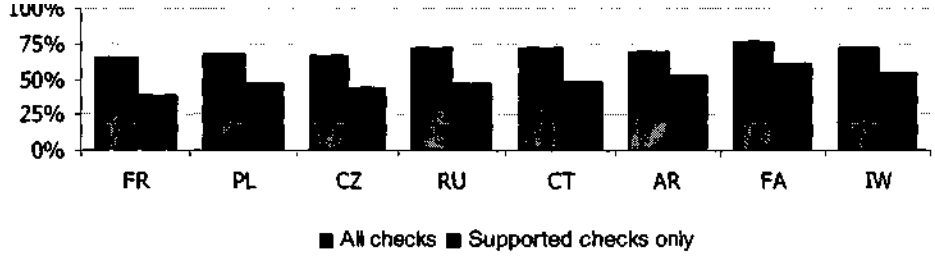

Figure 44. Summary error level by languages

If we look at the distribution of the error level by both languages and tools, however, we may notice such a trend: plug-ins deal with different languages almost equally while standalone tools do much better with some languages (such as QA Distiller with European languages) and much worse with other (such as QA Distiller with Farsi and Arabic and ErrorSpy with Russian and Chinese).

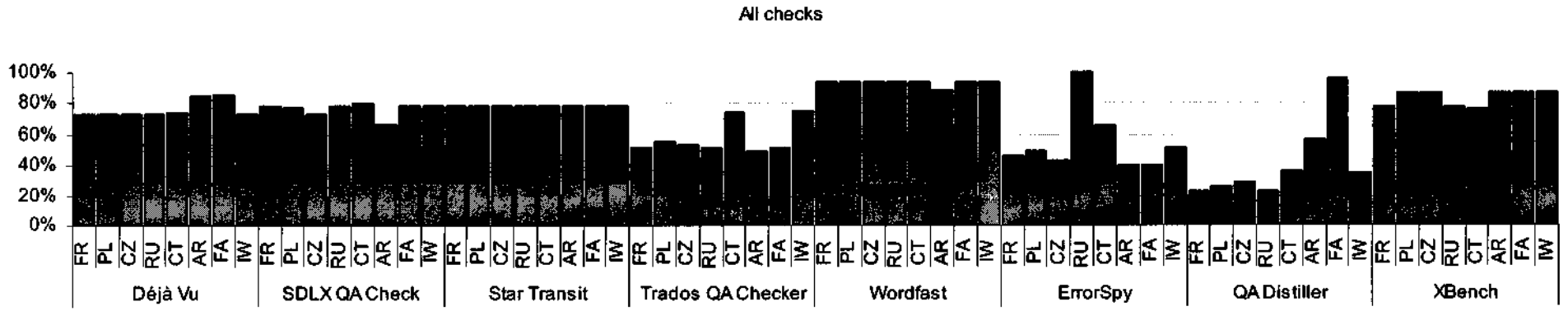

Figure 45. Detailed error level for all types of checks

If we take a more detailed look taking into account only the checks really supported by the tools, we'll be able to reveal that Déjà Vu handles Arabic and Farsi quite poorly whereas it gives the lowest error level for the rest of the languages. Trados QA Checker's rather high overall error level is achieved mainly because it handles Chinese and Hebrew poorly while ensuring relatively good reliability for all other languages. XBench, on the other hand, turns to be the most reliable tool for Chinese.

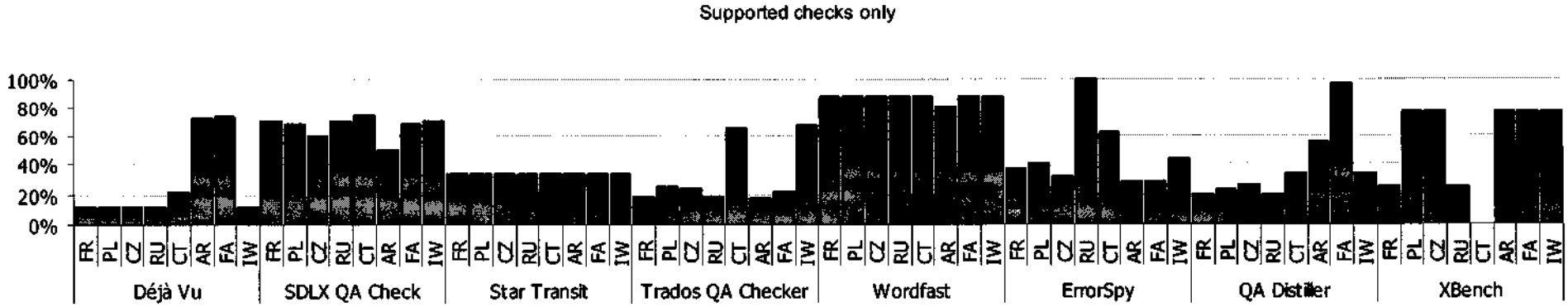

Figure 46. Detailed error level for supported checks only

Looking into the future

The future of QA tools looks rather promising. For about 7 years of their existence they have been developing from straightforward checks (as in Star Transit) to a bit more complicated ones represented by regular expressions or rules formulated in some other ways. This is just the first wrap of logic which can be compared to global search and replace that translators used when no TM tools existed. Apparently QA tools may at least adopt and implement fuzzy logic used in TM tools as well as employ probability and statistical methods to detect some kind of inconsistencies or even poor style. Such systems may become really trainable in the future and learn to detect errors not by regular expressions but by examples encountered.

The nearest future of QA tools, however, will most probably be related to batch processing, improved handling of individual languages (such as right-to-left ones), expanding supported file formats and eliminating some false positives (such as terminological errors in untranslated segments etc.). Closer relations with project TMs and wider support for project-specific and client-specific checklists are also easy enough to implement, so users may expect all tools to employ them sooner or later.

Usability is another good direction for development. Due to the amount of checks being performed, user interfaces of almost all programs are currently rather complicated. Implementation of additional checks will make them more difficult to work with, so it is really necessary to think well about end user convenience before extending capabilities of the QA tools.

Conclusion

The QA tools survey examined the acceptance and use of such tools by the translation community, and allowed us to make certain conclusions about the relationship between translation professionals and translation quality assurance tools.

Additionally, the benchmark made it possible to evaluate real capabilities of existing QA tools as well as their error levels and correlate them with the survey findings to visualise possible developments of this technology in the nearest further and later on.

The survey has shown a rather high penetration rate of QA automation tools (they were used by over 81 % of those who responded). On the other hand, it proves that the awareness of such tools is still low enough (over 11 % of all the respondents confirmed they had never heard of the fact such tools existed).

The survey has also confirmed that the most popular QA approach to date is to perform any checks that are easily automated and not too time-consuming while neglecting rather important, but more complicated ones due to time constraints. This finding together with the benchmark results reveals a definite perspective for the QA tools. The fact that QA tool developers are aware of this situation and have plans to develop them in this direction ensures translators one day will obtain more sophisticated quality assurance tools, and the overall translation quality will be constantly increasing, at least where it is really important.

The survey also revealed that the translation professionals are quite satisfied with the environments they work in, and there is no particular need to extend QA tools to different platforms and to add support for many interface languages. It also confirms that most probably neither TM plug-ins nor standalone applications will be able to force out another architecture. Most probably plug-ins will be used by both freelance and in-house translators while standalone applications will be employed by QA departments of language service provider companies.

Both the benchmark and the survey demonstrated urgent necessity of extending languages, encodings and file format support by QA tools as well as the tools usability. Main factors that keep translation professionals from wider employment of QA tools are lack of support for some particular encodings, languages and file formats as well as lack of time and skills necessary to perform all additional customisation to achieve higher level of error detection.

One of the file formats most needed by translators was Adobe PDF, whereas the most necessary encoding is definitely Unicode, and first language group that requires additional support is right-to-left languages.

Benchmark tests conducted at Palex have shown that QA tools available on today's market vary greatly in the matter of usability, learnability, architecture and support for different languages, encodings and formats. Although all the tools speed up QA process and increase translation quality to some extent, their error level is still high enough, which means people who perform QA spend most of their time deciding whether an error reported needs to be corrected or not.

Benchmark results, together with the fact that QA tools are young enough and are mostly developed internally by language service providers allowed us to suppose that QA tools usability so far was not a priority for the developers. However, before extending capabilities and adding more complicated logic, the developers surely need to make their tools more usable and think through the ways of adding this extended logic to the user interface in such a way that non-technical users would be able to fully employ QA functions.

Last but not least, this first research showed many weak points of the survey and benchmark tests and revealed more directions for the research development. This is a good basis to refine methodology and make this research an on-going project.

Acknowledgements

I would like to thank Katherine Weller of SDLX, Christiane Glaeser of Star Group, and Thomas Vackier of Yamagata Europe for offering me their help with test licenses and additional information for respective applications. I am extremely grateful to Palex QA and project managers who did their best to make survey questions and reply options clearer and more comprehensive as well as to everyone who spent their time to promote the survey and to respond to the survey's questions. Last, but not least, I am thankful to Aleksey Chernobay and Julia Otmakhova of Palex Ltd. whose extremely valuable comments helped me to make this paper more clear and coherent.

References

[1] Structured Query Language (SQL) (HTML). International Business Machines (October 27, 2006).

[2] King, Margaret (1997) "Evaluation Design: The EAGLES framework" Konvens 97.