



The Next Generation of Localization Tools, a Technical view on....

Introduction

For years people working in companies with a vision to open up foreign markets are forced to setup a localization process quickly.

Often the decisions to go global are made overnight, without recognizing the impact localization will have on the complete organization.

And in most of these cases the one in charge of localization got this task next to his or her daily marketing or technical publications career, as if it would be too expensive to hire a dedicated head.

At first this approach seems to work.

The one responsible for localization will outsource everything which will need translation and send this to one or more localization vendors after the quotes have been approved and the budget has been assigned.

The cost aspects related to this approach could be extreme, because the translation vendor will need to perform numerous tasks to determine what to translate in the package delivered by the client.

After a while when the original people assigned to localization realize that this is a full time job, and have changed positions to become localization professionals, they will realize that a lot of time and costs is related to **filtering** what should be translated.

Leaving this task to a translation vendor is less cost efficient because it is likely that you have a much better understanding about the product and documentation components which would be needed for translation then the localization firm will ever have.

Getting the product specific information can be obtained quicker internally, then allowing the localization firm access to this information from the outside, because contacts may not always share the information in the same way with an external party.



So what if there would be a method to filter your product no matter the size or type of material and select and send only the text for translation to a localization agency.

That would remove the effort at the translation vendor side to dive into yet another complex product to determine what to translate the hard way. Remember the questions you got from a translation vendor to deliver the complete build environment of your latest product to be able to determine which of the strings in the product will show up in the user interface and which are used for code purposes only?

So filtering the material before you send a translation request is a great way to reduce cost, and as such it will free up the localization budget for more languages on the long term.

Now we have determined the need for efficient filtering, let's see what's available in the Localization tools market and how it will match the requirements.

Localization tools availability and how they match the requirements

In the localization tools market there are a number of organizations which deliver various localization and translation tools.

They all focus on specific areas of localization which in general has been driven by the localization needs of clients at the time they were designed.

As soon as documentation translations became a requirement the tools to enable translations of this type of material started surfacing.

In general the same applied to software localization tools and localization project management applications which have been designed and released following the huge demand to be able to control the localization process in better way.

Based on your requirements you will find that the commercial available tools will fit a number of more or less independent groups.

- Tools which will handle your Software localization requirements
- Other tools which will take care of your Documentation localization requirements.
- Tools which will handle the project management side of your localization projects.
- Tools which will limit the translation cost by reducing documentation source material, and the typical single source publishing tools which are available in a number of different flavors.
- And the tools which will focus on Machine Translation.

Depending on the organizational structure and translation requirements your will find that in most of the cases the difference in translation requests can't be covered with just one tool.

The functionality and file type support is not matching your requirements, so it will become clear that if you decide to purchase you'll need at least two or more tools to handle your projects.

As a result a number of large corporations decide to figure out if there would be a possibility to design and develop localization software internally and fill the caps between the commercial tools available on the market, and the requirements they had internally.

Many organizations have followed this path and have build tools varying from simple batch files on one side to highly complex and fully integrated localization solutions on the other side.

I can even give you examples which will proof that these internal localization tools have made the difference in relation to cost reduction and increased throughput.

However it should be clear that you will need engineering resources and the C-Level support when you consider following this path towards an in-house made full blown localization solution.

You also have to make sure that people are aware of the potential benefits before you even start, so your high level support will not diminish before you finish the implementation.

Different levels of management will also need to understand that there is no way back when these tools are available, because any deviation on delivery methods or major changes in the delivery process will result in additional work for the localization team which is now spending all their time on process control and software maintenance.

In the section above you'll understand that in a number of cases you can make a quick decision, and go for buying when the internal support or team size simply isn't big enough to sustain the tools development on the long term.

So what's important when you are buying tools?

Obviously they should generate benefit in one way or another, so let's start with some of the cost aspects first.

As translation cost will consume a large section of your localization budget it should be clear that no matter what tool you purchase it should be able to reduce the translation cost.

Reducing translation cost can be achieved by re-using translations which is a quite common method in the tools business, and in fact is one of the most important reasons a number of tools exist on the market today.

But when you start off with Localization you may not see this as a clear benefit immediately, as your first localization attempt will need to be paid in full.

However after you have translated the same product for a number of times during the course of the year, you may find that you can potentially re-use 70-90% of your translations, depending on the version differences between current and previous version of the material.

Another efficient way to reduce cost would be by filtering or even reducing source material before it will be sent for translation.

Any string which has been removed from your source material would immediately reduce your translation cost across all languages on your localization list.

So when a localization tool is capable of figuring out the difference between strings you see in the final material and strings which are used for debugging messages / code (like you may have in software products) or strings which are used for control purposes the translation cost will be reduced by eliminating these strings from the package which will be sent for translation.

Next to the translation cost aspect, the tool should reduce the workload by improving the localization efficiency in your process environment.

It should process the translation request from moment of intake to moment of return in such a way that the impact of the localization process would be negligible in comparison with the time it takes to handle the actual translations.

This way it will reduce the manual labor cost which is usually underestimated when dealing with large translation volumes from day to day.

On top of this a manual process is not very predictable.

In general the translation cost aspect will be clearly recognizable, while only a few will understand that the time involved in processing files before they can be loaded into- and extracted- from a tool, should be identified as cost as well.

The localization software should have an understandable user interface which would not require a rocket scientist to get familiar with it.

Preferably for the novice users it should present a project wizard on demand, to guide this user level through the process from intake to delivery.

Following the generic list above there are specific functions which are related to the material you would like to localize.

Software localization by nature requires a localization tool which can handle another/different set of file types then when you have the intention to localize documentation. All respectable tool vendors available in the market today will have these file type support lists on their websites or in their documentation, so matching the file type support against your requirement makes sense.

If the file type support list is longer then you require today, you'll need to see this as an investment into your future localization needs, instead of overkill. You may get a request to handle a product tomorrow which is using a complete new set of files, which you didn't need to support today. Being able to process these files with the same tool because the file type support was already available will be a big benefit, as usually you don't have the time to wait for a fix from your tools supplier.

Adding file type support your self would usually require a development background and the support from the localization tool.

While a highly configurable solution may give you the opportunity to work on existing filters without the need of development resources

Another important aspect would be the localization tool design which should be modular to allow for updates on different file types which may be released in future without adding a requirement on your side to purchase their latest version.

To remain competitive your tools vendor will continuously need to stay committed in delivering state of the art solutions and filter updates. Remember that in your day to day localization job you might run into a situation which you can't handle with your current toolset, a flexible tools vendor should normally be able to discuss the possibility for a solution.

For software localization it is essential to be able to see a graphical representation of the user interface of your product, when the file type supports resizing. This will be needed to be able to handle potential resize problems which could have been introduced during the translation phase.

Another important feature is the ability to resolve hotkeys. Remember that hotkeys could appear on both static and dynamically generated UI pages which will mean that an automatic method of hotkey resolution should be preferable.

Having a method to define internals would also greatly improve the usability of the localization tools, by having a way to avoid translation of specific keywords, which may result in the breakage of the final application when translated erroneously.

Automated resizing on your translated material may sound good as well, but not in all cases. In specific User Interfaces the usage of dynamically generate pages can not be resized automatically because the actual link between UI elements is only established during runtime.

For documentation localization a different set of file type support would be needed, and here terminology and translation memory maintenance should be a part of the software package.

After summing up what's important you will realize that some of the benefits mentioned above will reduce you cost considerably.

And that's exactly way we didn't talk about Localization Tool license cost up till this moment.

In fact the cost of the license is completely irrelevant (within limits) when you consider the savings a flexible solution will generate over time.

What would be good enough to go for the title:

“Next generation localization tool”

The Ideal localization tool combines all aspects of software and documentation localization tools into one and the same package.

This will limit the diversity of localization tools, reduce cost of ownership by limiting the number of licenses within an organization and reduce the file handling, because each file (software or documentation related) will follow the exact same flow through the tool.

The tool should be driving the translations from one and the same master repository so every single string or string segment can effectively be used for translation of user interfaces and documents, which will improve consistency between both of the two streams.

It will also improve consistency in terminology between the software and documentation because both streams deal with the same repository as well.

And what if it would be possible to justify a localization tool by using its own repository to deliver detailed reports on string re-usage and match rates, which is why cost measurement functionality would be good to have as well.

When the tool can produce overviews on string re-usages it would be possible to calculate cost savings immediately. As any string which will be extracted from the TM is one less string you need to translate.

So in fact the software will justify itself by accumulating the cost differences and show how many re-usages have been performed over time.

In an ideal world the tool should support every single file type currently available both on software and documentation side and be capable of generating a graphical representation of all user interfaces for those file types where this can be generated from the source.

It should monitor terminology as well, as in many practical cases we have found that translators do tend to switch terms even if they have been instructed not to. In reality who, will go over 100.000 words translation to check for terminology changes after the translations have been delivered?

So if you can verify terminology with a localization tool, in such a way that terminology which has been determined during the start of a project will actually match against the terminology usage in the returning translations, this would enable you to find deviation in terms, which may be on purpose or not, at least they are identified.

For reasons of intended terminology switches between versions this may need override switches.

And finally a localization tool should have the possibility to override its default setting on filter set behavior to handle potential exceptions and allow for processing of new or minor changing file types with minimal effort.

By Erik Groeneveld

For more information:



Marsmanlaan 41, 1422 DJ Uithoorn, the Netherlands

Tel: +31 (0)297-527950

Fax: +31 (0) 297-591290

Web: <http://www.lc-t.com>

e-mail: erik.groeneveld@lc-t.com

