



White paper

Does Global Management Software make for an Agile Localisation Process?

ALPHA LTD
St Andrew's House
St Andrew's Road
Cambridge CB4 1DL
United Kingdom

 @thisisalphalive

thisisalpha.com

Within an Agile localisation process there are a series of "Waterfalls." These are short bursts of activity that move on to the next stage.

Translation is often seen as a sprint. A deadline is put in place, and targets need to be met. However short the duration of the sprint, typically one to two weeks, it starts with a planning meeting. This will determine the work to be done during the sprint.

The localisation workflow

In many types of activity, there is an optimum order in which work should be done. This is particularly true of translation, which is far more like a manufacturing process than a development process.

Translation lends itself to a fairly standard waterfall process:

While the order of these last two stages could be swapped, they can only occur after the approval process. Approval operates as a freeze on content, which is then set for updates to the TM.

In traditional waterfall project plans the Localisation Manager would schedule most of the translation effort after what used to be known as the “GUI freeze,” a concept possibly unknown to today’s generation of Software Development Managers. This is the stage at which the content is frozen, with no more edits taking place.

But these freezes are also necessary at other stages in the process. Translated content needs to be finalised before moving on to proof-reading. And with good reason: you cannot proofread text that has not yet been translated. You also cannot review against a glossary that has not been finalized. The most that can be done is to overlap each stage so that, for example, the first content that is translated can be proofread before the full content is complete:

Translate ➡ Proofread ➡ Review ➡ Approve ➡ Rebuild ➡ Update
Translation Memory

Notice that we do not overlap two of the stages with each other. Attempting to have review undertaken before the translation is complete becomes cumbersome and difficult. And even these overlaps are not always possible. In many workflows there are stages when all files need to arrive at the same time because they are all needed to complete the next stage.

So can a translation process ever be “agile?” and if not, what does an LSP need to do in order to fit that waterfall process into a customer’s agile development process?

Localisation agility

“Agility” can mean many things. It can mean the capacity to react quickly to change. It can mean the ability to change one’s own processes within a short space of time. But these kind of improvements are also part of a waterfall process.

In terms of the Principals and Values of the Agile Manifesto, translation as an activity is at best a partial fit, for several reasons.

Firstly, there is the structure of any localisation team. An agile team, under the above principles, is self-organising, with the authority to make its own decisions. It meets regularly to analyse its own performance, and it *chooses the order in which it will tackle its work*. It has technical excellence and can be trusted to get on with the job.

For most companies, LSPs are kept as external providers. They are sent projects and they return them, without being brought into the localisation team within the firm.

An agile client will likely have several teams, in reality: Development, Web, Documentation, etc. Because the nature of an agile team is that it is small. This is what gives it agility and facilitates face-to-face communication on a daily basis, removing any misunderstanding.

But as is clear from the above, an agile localisation process would ideally bring the LSP into the heart of the team, to allow for early planning in respect of issues that will affect translation. This would also ensure that change as requested by the LSP is welcomed, not fought.

In a situation where the LSP must remain external, a particularly close and trusted partnership must be developed with the customer in order to be effective contributors to an agile process. Procurement is usually wary of single-vendor agreements, with good reason, and so LSPs may also need collaborate with each other in non-competitive ways, or else be selected as the single supplier to one team. Even this is difficult, because there are genuine benefits to collaboration between teams, not least the removal of “fat” or “waste” from a process.

Lean processes

“Lean” is a concept which is extremely well understood by LSPs, and is applicable beyond agile processes. It is already prevalent throughout the translation industry where waterfall processes are in operation.

The Toyota Production System, applying the principals of “Reduce, Reuse and Recycle,” and “Just in Time,” is a good analogy for the machine-aided translation. A factory floor or conveyor belt localisation system is operated by LSPs who have been deploying CAT and

GMS systems for many years. The very fact that all the waste and fat has already been removed from the translation process means that any margin (both financial and in terms of lead-time) is already wafer thin. This provides very little scope to further shorten the sprint duration, or to react when what is delivered by the customer is unexpected, or even un-fit for purpose. It is certainly not an agile process. Only the development team may introduce change without consulting the LSP, who has to react to problems rather than adapt to improve processes.

So, once the duration of the sprint has been set and the files have been submitted to the LSP, translation is in fact a waterfall process; and any issue which causes someone to stop the conveyor belt is going to result in empty shelves where translated content should have been.

Finding a compromise

The partial fit is that we may adopt the principals and values of the Agile Manifesto, in terms of trust and daily communication and being open to change.

Most customers are familiar with the concept of Translation Memory (TM) which is the basis for a reuse and recycle process. Essentially, for those unfamiliar with it, previously-translated content is stored and suggested afresh to the translator when a similar or identical phrase emerges.

However, the aim of these two processes is to reduce wastage. Whereas the main target for a lean process is to maximize the work that is NOT done.

It is not enough that the software is able to present a 100% or even an In-context Exact Match (ICE); what matters is the true accuracy of that match, and what's more, assuming that it is accurate, that anybody involved in the time-consuming tasks of translation, editing or review is well aware which segments can be reliably skipped.

Most tools offer a workflow which can direct these segments straight to the reviewer: the translator either doesn't see them or they are locked. In order to do this, tools need to be able to interpret the meta data stored with each segment. As the assembly belt moves from stage to stage, it is essential to pass on the status and origin of the translation.

"Just in time," as applied to translation, is about determining the proper stage in a workflow to tackle a particular part of the procedure and how to deliver the necessary segments and supporting information (glossary, reference material, machine translation) directly to the worker. This does not always mean at the latest possible stage in the workflow, but it certainly means at the point of need, and in a state of accuracy, if waste is to be avoided.

Applying Machine Translation

Most GMS offer the option to apply Machine Translation (MT). MT isn't perfect, so MT segments should be considered in a similar manner to fuzzy matches: they need to be corrected, sometimes discarded. In a similar way to fuzzy matches, the process of correcting the segment should not take longer than any gains provided by the MT or match. In the real world this often happens, particularly where segments contain placeable tags. Translators will tell you that they prefer MT matches to be presented by the CAT tool as options alongside fuzzy matches, and not pre-populated into the file. And so, while the analysis and quote which takes place on the GMS needs to account for MT, it is optimal to move MT look-up to the live translation stage, which usually means in the CAT tool.

The CAT tool needs to be able to query the MT engine, the terminology database, and one or more TMs within one second of the translator opening the segment, something which has only truly become a reality in the last few years, with multi-core CPUs and fast internet. Even now it cannot be taken for granted. All tools, servers and networks in the workflow need to be stress-tested and optimized.

Most purveyors of GMS will promote their online web-translation tools, unless they also happen to sell complementary CAT tools. The selection of a GMS may be based on commercial decisions, workflow requirements, or the need to connect to a particular Web-CMS, and may not be a perfect fit with one of the two industry-leading CAT tools provided by your LSP. Content and meta-data can be exchanged between tools in a variety of package formats: OASIS Xliff, Xliff:doc, TIPP and how this content is read will determine whether the CAT tool needs a live link back to the original TM, Term database (TD) or MT engine of your GMS.

Optimisation

Optimisation is something your LSP should provide as part of the Sprint Review. That is, at the end of each sprint, or single pass through the workflow. Just as the mechanic may provide the Tour de France rider with food, water, adjustments to the brakes or changes to saddle position on-the-go, we must be careful not to do something that will put a spanner in the spokes. Feed the wrong content into the workflow, use the wrong TM or TD, unfinished or un-confirmed material, and the whole system can trip up. It's difficult to do something about this until that particular sprint is complete, because of the waterfall nature of the translation process.

As an example, the Adobe InDesign localisation team found that 3% of their source UI strings fed into the first Sprint contained spelling mistakes. Over 25 languages, this represented a waste equivalent to 75% of a single language scope. One of our MMORG customers found that their XML authors were continually inventing new syntax for formatting nodes: <bold>, , <ibold>, etc. such that nobody was aware that some content was not being exposed for translation until the delivery was assessed at the end of the sprint.

Changing during the sprint

It is possible to adjust segmentation filters, TMs and TDs, and to tune MT engines on-the-fly and feed any affected files back through the workflow, providing that issues are discovered before the end of the sprint. At the same time, it is essential to remove garbage and defects from the TM and TD before too many files are affected. And it is important to recognise that not all sprints are of equal importance and to share that information with your LSP.

Take, as an example, a situation where a firm is a long way from finalizing any of the UI. Perhaps a pseudo-translation is more appropriate as a way of identifying potential errors. If a company needs an alpha version of a product with far too many strings in too short a timeframe, then a machine translation may be a good start. (An adjustment to the workflow will be necessary to ensure this "garbage" doesn't end up in the TM.) Why pay money for a human translation, if this sprint doesn't warrant it? If you have reached Beta² and Beta¹ is still in translation... halt it. If your translation process cannot be shortened, maybe you should choose to translate every other sprint.

The importance of planning and review

In order to pick up on many of the issues outlined above, good planning stages and reviews are key. Testing new product lines in a sandbox on a GMS with pseudo-translation or MT can hugely reduce later issues. Giving the engineers time to prepare and test new file filters and align old material for Legacy TMs does more, and allowing adequate time for Subject Matter Experts to build the term database means far fewer corrections.

While we are emphasising valuing individuals and interactions with them over tools and processes, remember that change is good. LSPs should explain the KPIs and how they will respond to the indicators. If this involves un-plugging one tool and swapping it out for another: who should a firm trust to give them an objective view of the impact that will have? We would hope that it would be the LSP.

Change is upon us, whether we would risk it or not...

Its worth bearing in mind, that many GMS while providing a centralized workflow, portal and server for TMs and TDs, have been up to 10 years in creation. Some are a cluster of servers each with its own database and supplying a different service. Not all of them offer Single Sign-on allowing the user to access project files, TMs and TDs in one location. Sometimes, data is cached. The TM you think you have downloaded is actually last week's export.

As we move ever more towards true Cloud Computing, some GMS features appear more agile than others; but what is really happening here is just the shortening of the waterfall process by breaking tasks into ever smaller chunks – something which is only effective if work is kept online, in the Cloud. Already some tools exist which operate at the segment level: as soon as the status of one sentence changes to “translated,” the reviewer has access to edit it. With content connectors, there are tools which build translated web pages hour-by-hour, paragraph by paragraph so that there are no surprises at the end of the Sprint.

However, such tools are at present targeted very specifically at certain types of content: there is no single GMS equally suited to all types of web, software or marketing content, in the way that an *individual* tool is able to offer WYSIWYG translation editing.

Typically these online tools are 50% slower for translators to use than the leading CAT tools. They should not be used to tackle a 50K word project, much less a large contract. But by its very nature an Agile process should only be requesting a maximum of perhaps 5k words per Sprint.

Herein lies the conundrum: the LSP is already an expert in lean waterfall processes offering the most competitive rates per word, based upon tools in which we are experts, purchased at our own expense. We now need to re-think the way translation is purchased. The agile framework requires that LSPs take part in frequent meetings, and prepare input for the Sprint Review, for which there is no budget. The rate per word barely leaves time to translate in-situ and no scope to transfer to more productive CAT tools.

Some of the savings of these new processes need to be directed towards a higher rate-per-word, or an on-demand, by-the-hour remuneration system. Banks, online market places, sub-way travel... these systems already recognise the importance of point-of-contact payment. The technology already exists. We now need to create a lean administration process, removing the need for invoicing and purchase orders, cutting out the weekly financial statements or the need to chase for payment.

How do we build a trusted partnership of equals that is true to the principals of the Agile Manifesto?

A trusted partner is one who is open about the profit margin they are making. The partner feels valued, because their customer understands what is a fair margin for them to make. They are able to explain every part of their input to the process and where it can be optimised, so that their customer trusts that they are receiving the most cost-efficient service available. The partner does so with competence and excellence.

Found that interesting?

If you're not already signed up to our newsletter, email marketing@thisisalpha.com to receive lots more great articles that will help you on your journey to going global.