transifex®

Localization for Agile Teams

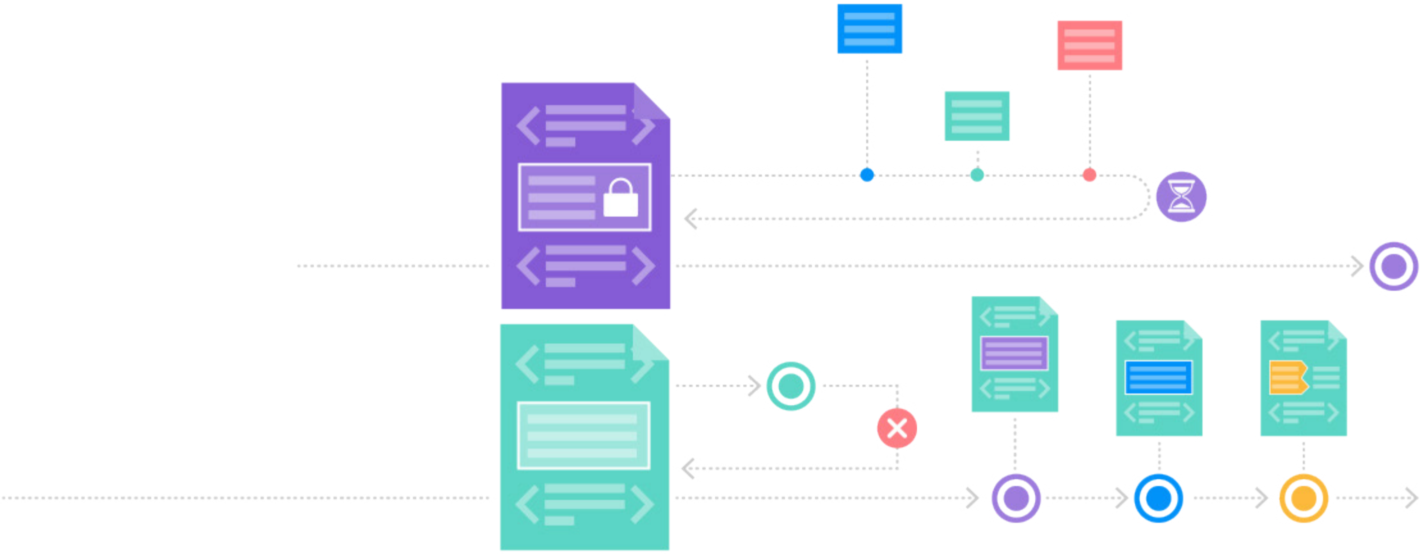
Translate at the pace you develop code

Speed is King

Software companies are deploying code faster and more often than ever in order to go to market sooner with new products, features, improvements, and languages.

Facebook pushes code twice a day as part of their commitment to "*ship early, ship twice as often.*" Etsy moves even faster, pushing code almost 25 times a day. This raises an interesting question for global companies.

How does localization fit into the big picture with agile development methodologies and their emphasis on speed?



Traditional Software Localization Workflows

Traditional localization workflows are inconsistent with agile methodologies.

Localization is crucial in reaching new markets and meeting the expectations of customers in existing markets. Developers traditionally approached localization in one of two ways:

1. String Freeze - Some development teams add an extra step to the release process where strings (also referred to as "segments") cannot be added, edited, or deleted - only bugs may be addressed. Known as a string freeze, this step gives translators the necessary time to work on and test translations without fear that the original strings will change.

Depending on the size of the application, a string freeze period could last anywhere from a week to more than a month. After the strings are translated, they are returned to the developers for use in the final release. The process is then repeated for the next release. And with each release, the developers must identify any changed strings that need translation.

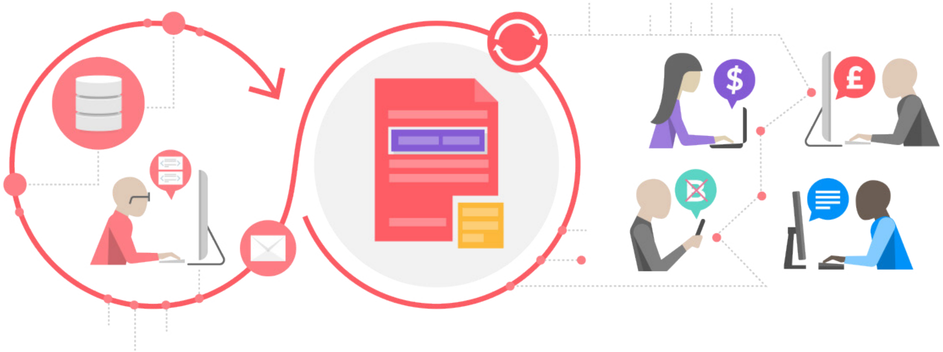
2. Post-Release - Another traditional localization approach is to release the software and add translations afterwards. With this approach, companies are unable to release in multiple languages simultaneously. The gap between the time software is released and when translations are completed also means that sections of the software may not be translated at all.

If errors are found in the source content during the translation process, shipping localized versions of the software may be delayed even further as the development team must go back, address the issue, re-release, and then begin the translation process again.

Using either of the above mentioned software localization workflows, whether ordering professional translations or crowdsourcing, means localization will be performed in large batches. Not only is this cumbersome and inconsistent with agile methodologies, it also delays the availability of the software in multiple languages. And every delay is a missed opportunity to create new customers and generate more revenue.

New translation methods further complicate traditional workflows.

Traditional translation methods include hiring freelance translators and working with translation agencies or Language Service Providers (LSPs).



Some of today's leading global organizations including Facebook, Twitter, and Reddit are moving away from these traditional methods and instead leverage their user communities, asking volunteers from around the world to contribute to translations (which are often done for free or for a low cost). Internal staff or professional translators then review translations for quality and consistency before the final release.

Leveraging user communities can lead to higher quality translations as crowdsourced translators are often users who understand the product and its lingo.

Unfortunately, in the traditional localization workflow, crowdsourcing translations carries its own logistical challenges. For instance, companies must figure out the best way for translators to access source strings or files and submit translations; emailing translation files back-and-forth is slow, insecure, and impossible on a large scale, and systems like GitHub are suboptimal for translators, most of whom aren't programmers.

Companies must also decide how they'll communicate with and manage hundreds, thousands, or even hundreds of thousands of volunteer translators who are translating the product into dozens of languages. Add in the fact that code is deployed multiple times a day, and crowdsourcing translations adds even more headache to the traditional localization workflow.

A **Modern** Software Localization Workflow

The rapid pace of today’s development processes means a continuous localization process is needed to run in concert with continuous deployment. Such a process must respond and adapt to the speed at which changes are made and updates are delivered. In other words, there's a need for a modern approach to localization that’s in sync with how development teams work today. This continuous localization process can solve the problems presented by traditional localization workflows and even allow for the crowdsourcing of translations.

However, in order for continuous localization to happen, a robust **Continuous Localization Platform (CLP)** is needed.
A good CLP will support a continuous localization methodology by allowing:

- Translations to take place **daily** and be delivered as soon as possible
- **Deep integration** with developer build systems

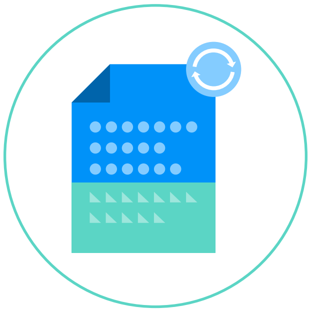
Integrating Continuous Localization into a Typical Build Cycle

Many development teams use a Continuous Integration (CI) tool such as Jenkins or Bamboo to build and test software automatically and continuously. Every time the source code changes, the CI tool is notified and a new build cycle starts. A default build cycle includes stages like validate, compile, and test, among several others. This allows development teams to deploy changes rapidly with the confidence that the code has undergone testing and quality assurance.

A continuous localization process can be established with a few additions to the standard build cycle:

Create and Upload i18n Files

The first step is for the CI tool to automatically create an internationalization (commonly abbreviated as “i18n” because of the 18 letters between the “i” and the “n”) source file and send it to the CLP via an Application Programming Interface (API) or a specialized tool. An i18n source file is simply a file with the translatable strings, e.g. in English, which developers have marked and extracted from the source code. Each development framework has its own i18n file format. For instance, iPhone and OS X apps use the .strings format whereas Android apps use their own XML-based format. A CLP should support a wide variety of file formats so that multi-platform apps and software can be localized in the same place.



Translate Content

Once the source file is uploaded, translators are notified by the CLP and start working as soon as possible. Speed of delivery, quality, and consistency are equally important. Use of a CLP can maximize all three and make translations happen more efficiently in a cost effective manner.

CLP features such as translation memory (TM) can save both time and money by leveraging previous translation work. TM creates a database of completed translations so translators can quickly find and reuse previous translations. Only changed or newly added strings are translated in subsequent build cycles.

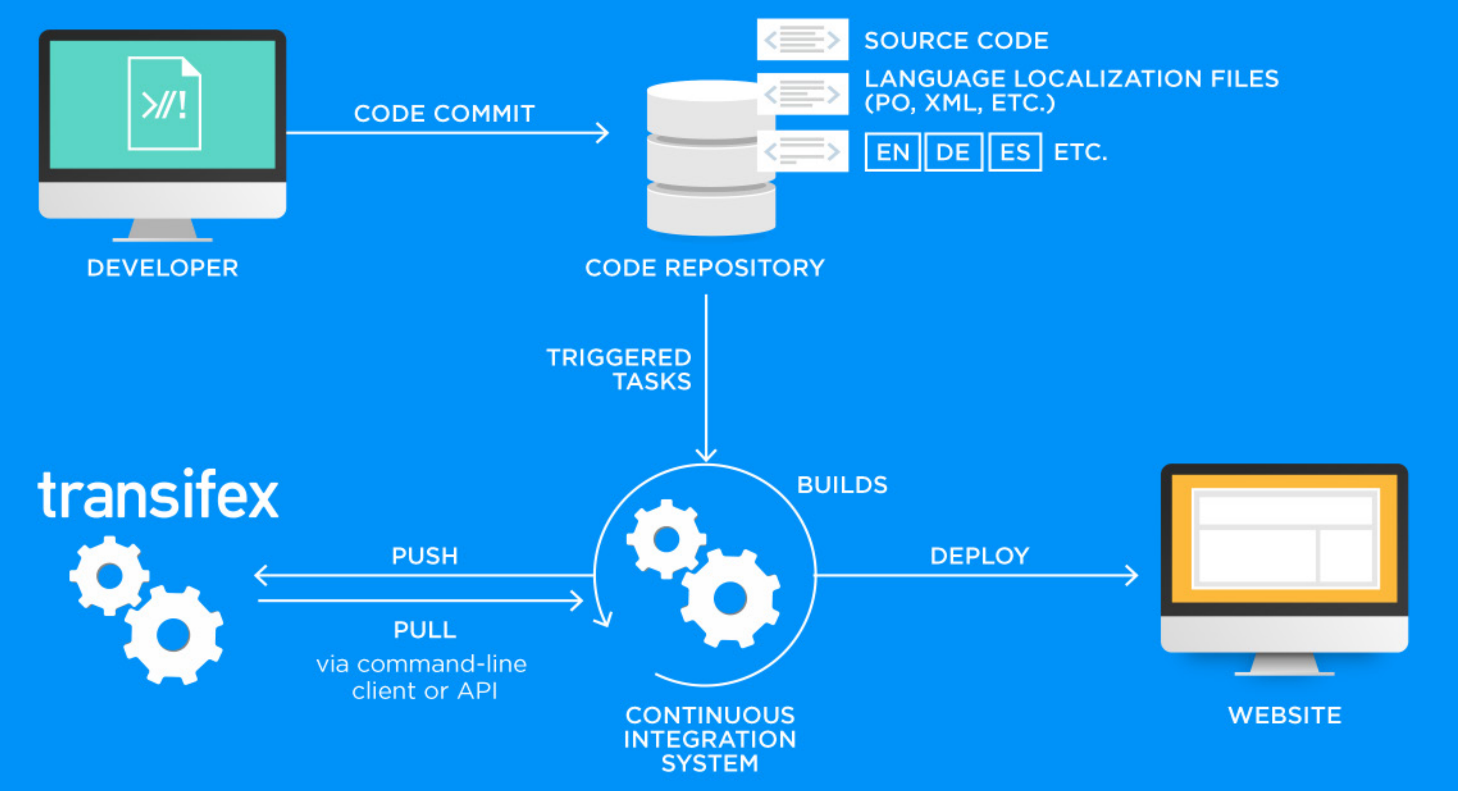
A glossary is another CLP feature that serves as a reference for translators so industry- or product-specific terms are translated the same way every time, creating a consistent brand and user experience.

Retrieve Translations

After translations are completed, reviewed, and approved, the CI is alerted and pulls the translations and commits them in the code repository (e.g. GitHub). The code along with the necessary translations can now be deployed. And if a team prefers to deploy sooner, they can set the CI tool to pull translations when they surpass a certain threshold, e.g. over 90% complete.

By integrating the CLP with a CI tool developers already use, localization becomes part of the build cycle. The whole localization process – from the creation of the i18n file to the submission of translations into the code repository – is automatically repeated each time the build cycle runs. This results in smaller release cycles, which is critical for teams deploying several times a day.

Continuous Localization Stack



PROCESS DESCRIPTION

A continuous localization process occurs within an internationalized (i18n) framework to support the automated transfer of assets from step to step

1. A developer commits new code to the code repository; customerfacing strings are stored in language localization files (format dictated by code language/environment)

2. The Continuous Integration (CI) system listens for changes in the code repository and automatically pulls new strings from the repository and pushes them into Transifex for processing in the translation workflow

3. Upon completion/approval of the translation workflow, the CI system will recognize and pull approved strings back for publishing or inclusion in a new software build

Benefits of Continuous Software Localization

In addition to making localization possible for agile teams, there are several other benefits to continuous software localization:

Faster Launch

Software can be launched to new markets faster because development and translations run in parallel. A well-executed localization effort can create devoted fans and new customers on every continent and in every time zone. This can translate to increased revenue as 56.2% of consumers said that obtaining information in their native language is more important than price. Nitro PDF, for example, has seen a 60% increase in Localized Product License Sales as a result of their localization efforts.

Increased Focus

With a CLP, localization becomes less of a concern for developers. Without the need to spend time creating a new i18n source file or informing translators every time a string is changed, developers can focus on coding and building a great product for users and let product or localization managers handle the interactions with translators. Of course, developers must still keep in mind that strings may change in width after translation and that dates, time, and currency should not be hard-coded.

Significant Savings

Automation reduces the amount of time spent by developers and managers on manual, repetitive tasks, which can provide significant cost savings for companies and result in fewer translation errors. Because localization is done continuously as opposed to intermittently in big batches, there is no need for re-work or rush jobs which are almost always more expensive than standard translations. Companies that choose to leverage existing translations via TM save even more and ensure that customers see consistent translations while having access to the latest version of the software simultaneously in all markets.

Total Flexibility

Translations make up a large part of the continuous localization process. With a CLP, companies have the flexibility to translate content using the resources that best match their goals, budget, and quality requirements. Crowdsourcing translations becomes a viable option because a CLP solves many of the logistical challenges that crowdsourcing can create. Not only does this allow enthusiast users to translate new strings as soon as they appear in the CLP, companies can take a fully agile and modern approach to development, translations, and localization.

Going Beyond Software

In addition to the product itself, product documentation, landing pages, help centers, and other supporting content are essential to users. Companies often choose to localize these additional pieces of content because they're part of the overall user experience and conversion process.

When localizing other forms of digital content, companies can also use a continuous localization approach, similar to the one outlined for software localization. Rather than extracting strings and uploading an i18n source file, a plain text (.txt) or HTML file might be used. Companies can decide to use the same translators who worked on software localization. Not only do they already understand the product, they are familiar with much of the vocabulary and syntax surrounding it.

When supporting content is localized using a continuous localization methodology, companies can publish new or updated content to coincide with the launch of a new product feature for a fully automated localization workflow.

Conclusion

Don't let localization slow down the agile process you've worked so hard to create. A continuous localization approach supports the way you work, not just in terms of speed — also by providing high-quality translations regardless of the translation method you employ. Choose your localization tool with the same rigor you'd apply to selecting the rest of your development stack by choosing Transifex.

Get started with continuous localization today and launch to new markets sooner using our cloud-based Continuous Localization Platform.

transifex®

Transifex is a global leader in
Localization Automation solutions.

Its Transifex Live platform is an ideal solution for mid-market companies that need an end-to-end solution for translation management and global site hosting.

For companies with more mature, continuous localization requirements, integrating with Transifex using our API or command-line client is the optimal solution.

+1 855-418-5000

www.transifex.com

1050 Chestnut Street, Suite 203
Menlo Park, CA 94025 USA

5 Chimaras, 3rd Floor
Marousi, Attiki 151 25, Greece