lokalise


# **Building Your Localization Tech Stack:** The 5+ Tools You Need and How to Choose Them

# Introduction

Localization is a lengthy process. It involves much more than running a set of text through Google Translate. From your code to your interface designs and marketing materials, localization touches every part of your software or mobile application product.

For this reason, it's wise to build a tech stack that can help with every step of your localization workflow. In this guide, we'll be unpacking why a localization tech stack is important, how it fits into your established development process, and what tools you may need to incorporate.

# What is a localization tech stack?

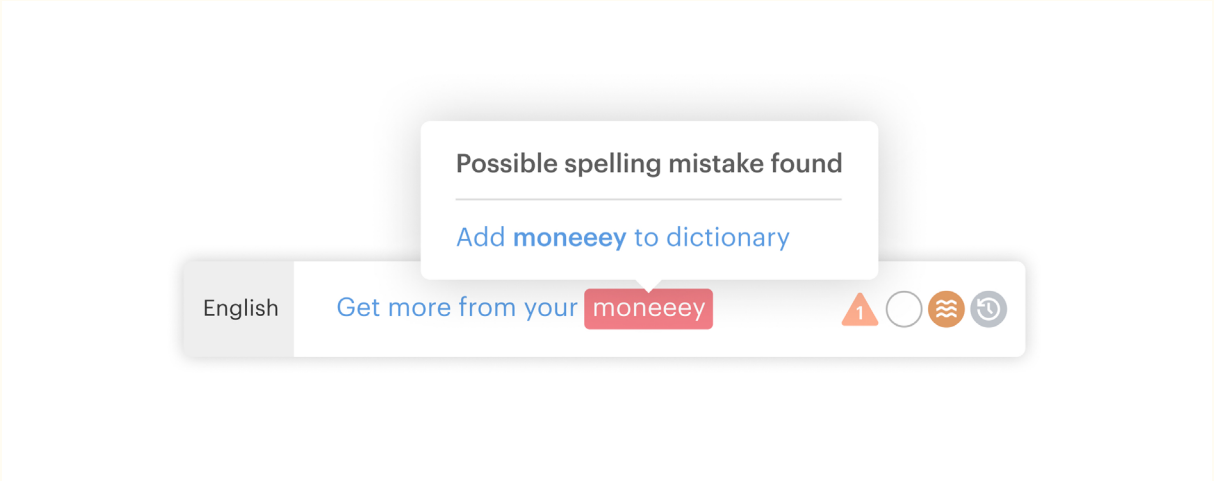
A **localization technology stack** consists of the software, products, programs, and tools needed to successfully localize a software or mobile application product. It refers to the combination of technologies that powers your localization strategy, from integrating your code base to formatting your designs to delivering the final product to different locales.

Localizing a software or mobile application product from scratch is complex. The variety of technologies, programs, and tools alone — not to mention the vast differences between cultures and languages — are enough to overwhelm even the largest organization and translation team.

A localization tech stack is important because it **efficiently simplifies and saves money** on the overall localization process. Localization can be a very lengthy process due to the number of technological frameworks and cultural specificities. Technologies like machine translation (MT) and computer-assisted translation (CAT) tools can cut down on the time and cost of localizing your software.

Doing localization manually without specially dedicated tools could take up to six months, while you can achieve the same result in a matter of weeks if you have a solid localization tech stack.

A localization tech stack also **ensures consistency and quality** of translations across your product. CAT tools, like translation memory and quality assurance (QA) technology, can ensure that details from phrasing to punctuation are consistent and correct for your users.




# Building Your Localization Tech Stack

Before we dive into the various tools that make up a localization tech stack, we want to introduce a new perspective on the concept: your localization tech stack shouldn't be a standalone set of tools. Your focus should not so much be how to build a separate tech stack for localization but instead how to introduce localization tools to your established software development workflow.

The concept of a localization tech stack doesn't only include the localization-specific tools we'll discuss below or a translation management solution like Lokalise. It should also consist of the other technology you use alongside it, such as code repositories like GitHub or BitBucket and any software development-oriented tools like JIRA.

The localization solution(s) you choose should also blend nicely with your software development tool stack and the processes you've already established, especially if you've thought about going global and localizing your products from day one. It would be more seamless to integrate a solution like Lokalise into your software development stack versus introducing a new, standalone localization tech stack.

In the past, localization was an afterthought — a process that takes place after product  development. Now, it's viewed more as an integral part of the development process and workflow.

That's why your localization tech stack should blend with the tools you're already using to develop your product. (Keep this in mind as we walk through the localization tools and technologies in the next chapter.)

The same goes for the design process, as it's just as important to consider how different languages and cultures may affect the finished design of your product.

For example, let's say you're developing a mobile application in English, but your team didn't consider localization until after your design is complete. You want to introduce your product to the Egyptian market and need to translate your product to Arabic, which is a right-to-left (RTL) language. When you introduce the RTL language, the components of your product (like graphics and text) can't resize and handle the new orientation.

Languages like German and Swedish present a similar issue. Translating English to German causes text to expand up to 35%; English to Swedish does the opposite and causes text to contract by 35%. Localization after-the-fact can cause significant damage to your product design. (We'll talk more about design-specific localization tools next.)

**Takeaway**: If your company wants to go global and eventually operate in multiple markets,  the localization workflow must start at the ground level, alongside your product's development and design process. If you don't, it will be much more expensive to apply new languages and cultures down the line.

# 5+ Localization Tools to Consider for Your Tech Stack

We've discussed how your localization tech stack should integrate seamlessly with your software development and design tools. But when it comes to the localization part, what tools should you introduce to your established processes?

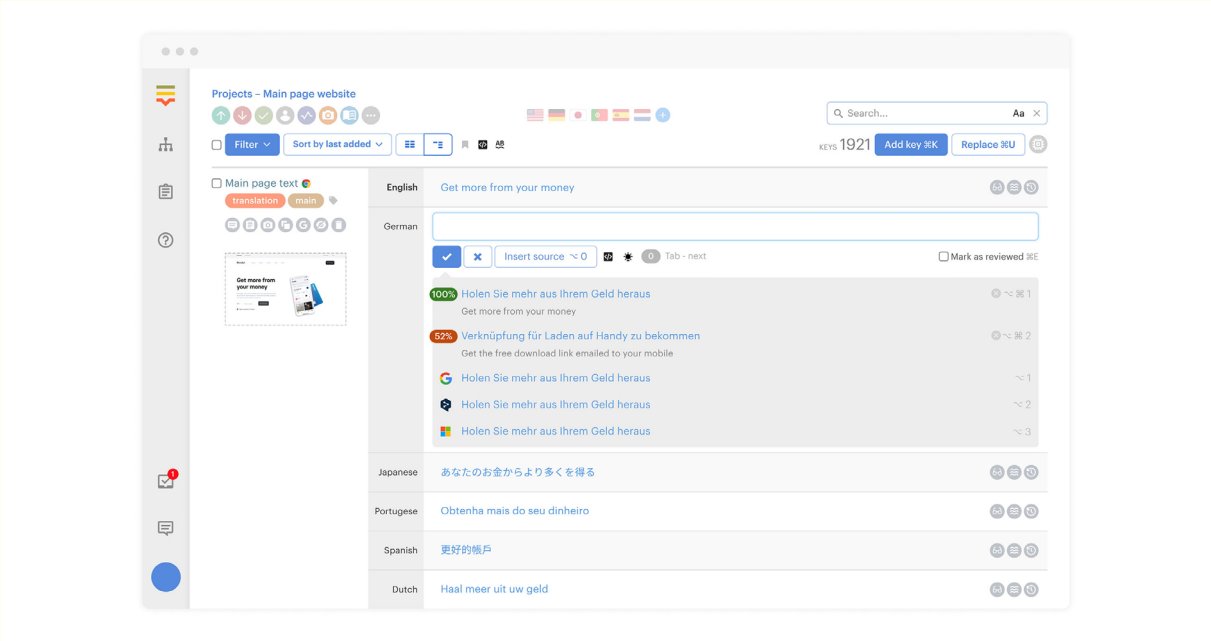
Well, there are different types of localization tools. Some check multiple boxes, and some specialize in very niche processes. Likewise, you'll see that each tool solves for a distinct problem or set of problems we discussed in the introduction.

## Translation Management Systems (Best Overall Localization Tool)

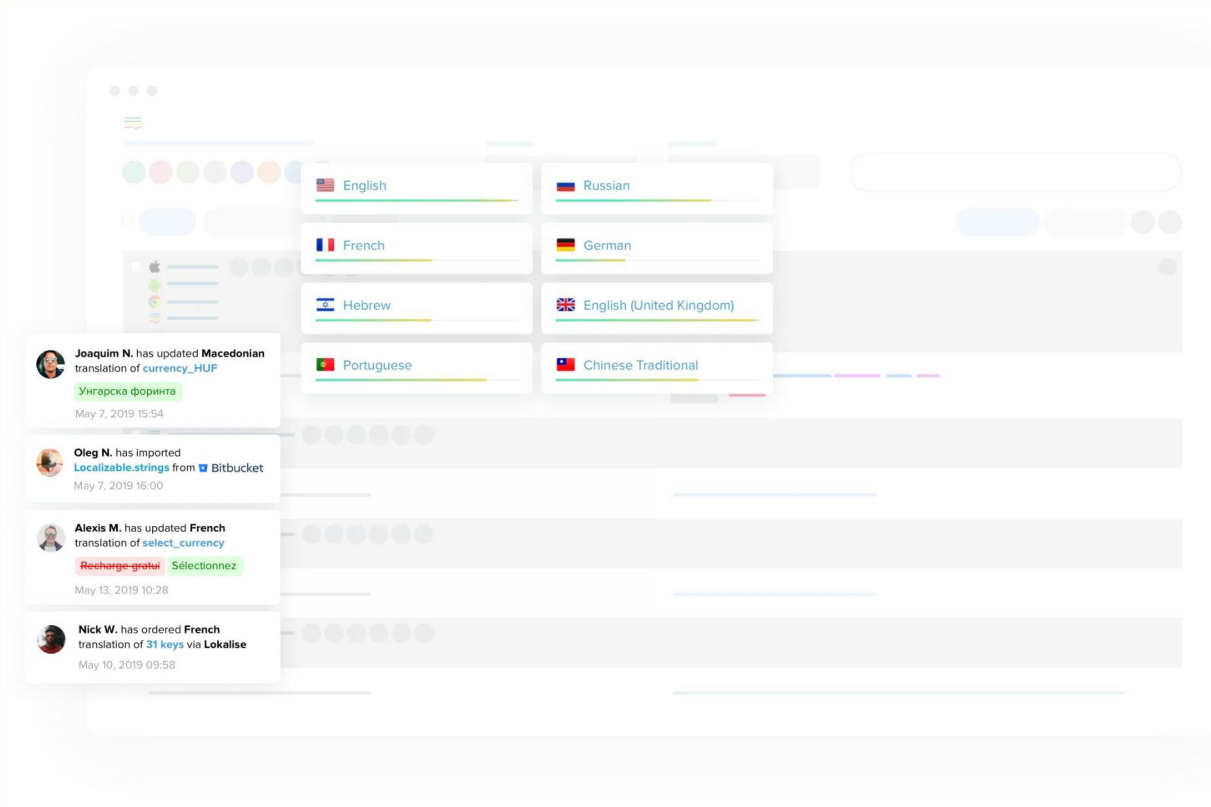
Foremost, let's talk about translation management systems. Translation management systems (TMS) are an excellent example of a localization tool that checks many boxes.

In fact, a TMS typically brings together many of the separate tools mentioned elsewhere in this guide, including CAT tools (i.e. translation memory and glossary), machine translation, and workflow automation tools (i.e. code repositories, integrations, and API).

A TMS combines many critical localization tools under one umbrella. But that's not all — a TMS also helps organize, manage, and automate the localization process by providing unique project management capabilities like dashboards, tracking systems, and task assignments. For this reason, virtually everyone in the localization process — developers, translators, designers, and project managers — can use a TMS.




Project managers at big translation agencies may use a TMS to organize and manage their translation projects and translators. If you're managing the jobs of multiple freelance translators worldwide, a TMS can help you ensure your team hits deadlines and receives fair compensation based on the translations they complete.



Is a translation management system the best choice for everyone? Well, it depends. In terms of budget, some TMS choices may be too expensive for smaller teams. If a developer only has a small website to localize, a TMS may be overkill. Instead, if you're working with two to three people, you could just use a code repository to store code and collaborate on translations.

If you're localizing a large website, software or mobile application product that involves many translators and stakeholders, it's a good idea to use a TMS to organize, manage, and automate your localization workflow.

**Note:** Not every organization and localization workflow requires a translation management system. The rest of this list will walk through the various separate localization tools your team may need depending on the size and budget of your localization project.

## Computer-Assisted Translation Tools

Initially, translation technology started with computer-assisted translation (CAT) tools. These tools solve one important problem across the whole localization spectrum — helping translators translate text efficiently.

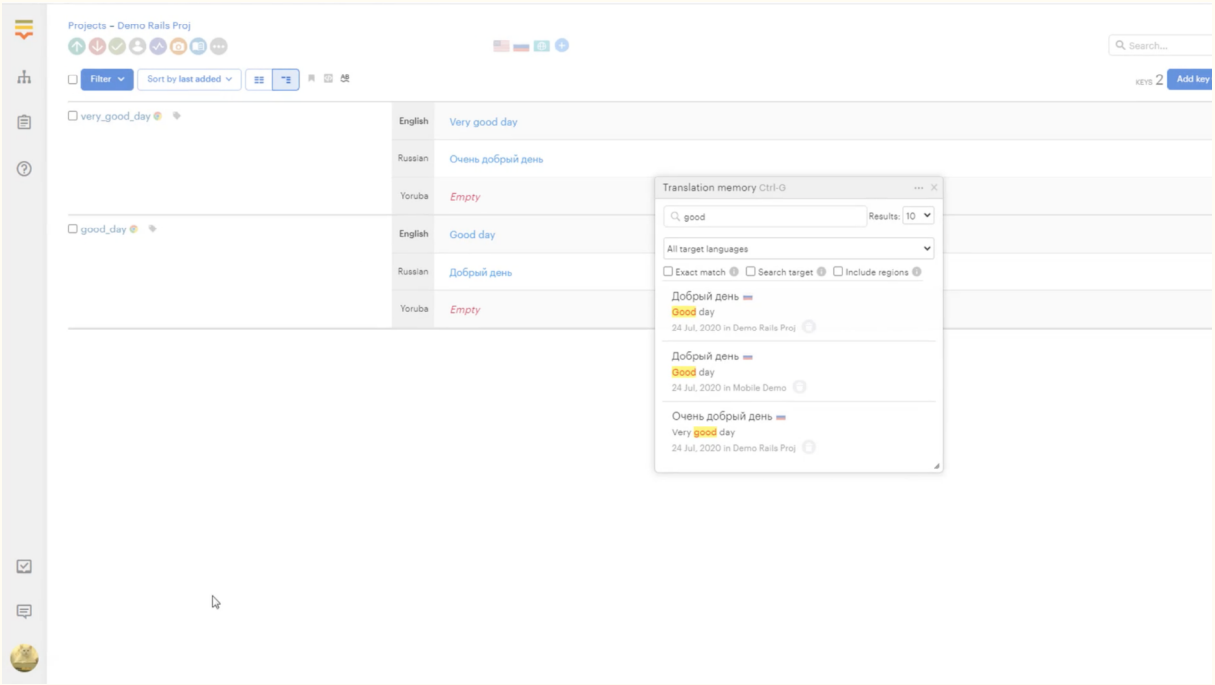
Whether translating a single webpage or a large volume of text for a software or mobile application, it's important to keep the text consistent across the entire product. Consistency can take time, however. To be efficient and timely, you don't want to spend valuable time on strings that have already been translated during the project.

CAT tools significantly ease a translator's workload and improve translation speed, quality, and consistency. They provide consistent, efficient localization through simple importing and exporting for your text files and an easy-to-use, convenient editor for your translations. Lastly, while CAT tools don't provide the same level of project management as a TMS, they do allow you to assign tasks to various translators and keep track of who's working on what across your localization team.

CAT tools should also include:

## Translation Memory

A translation memory keeps track of the content you've already translated. Each translation is added to a database of sentences, paragraphs, and words. Moreover, everything that's typed in the editor or uploaded via an API is automatically saved in the translation memory for future use.




## Translation Glossary

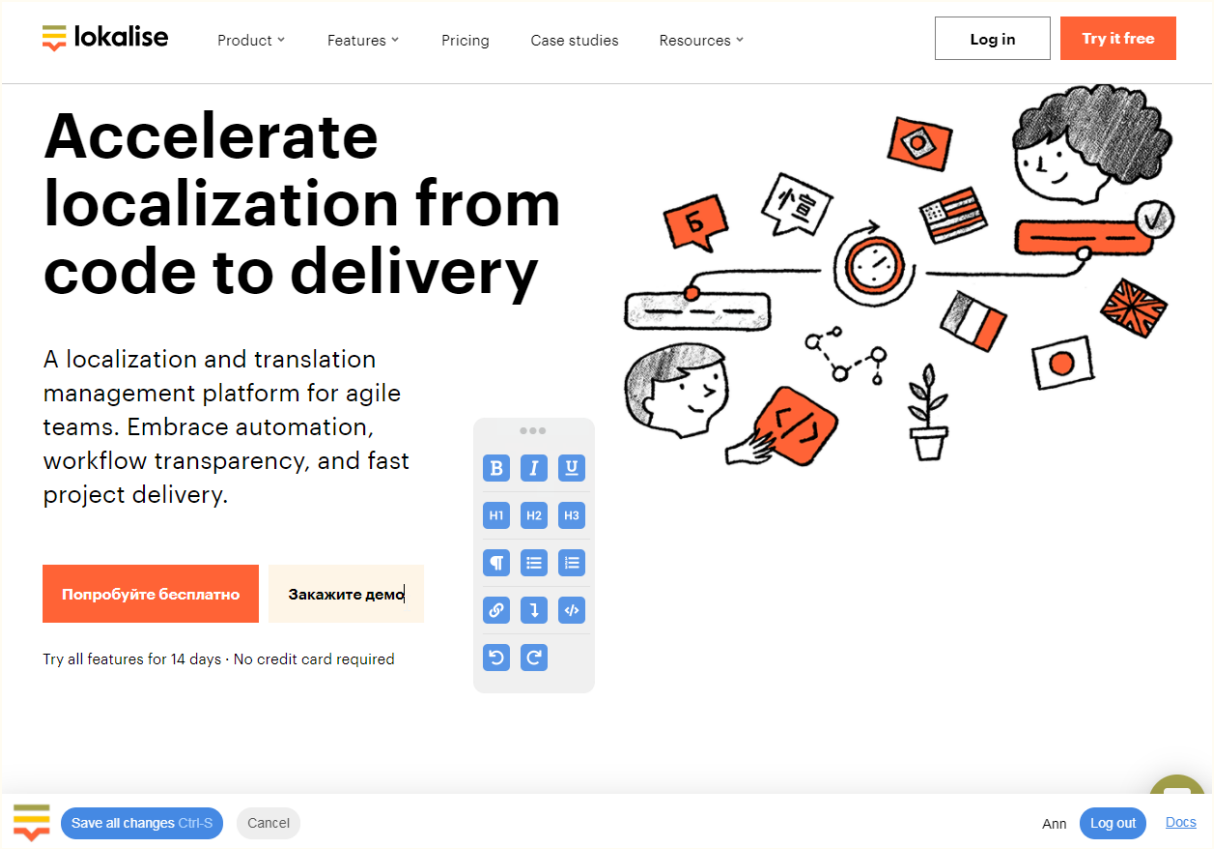
A translation glossary ensures that company-specific technology (e.g. product names, campaign jargon, or special company terms) is translated consistently across your website, software product, and any other public-facing material. Even if these translations are made by different people, a glossary confirms that your most important terms will be the same.

## Quality Assurance

A quality assurance (QA) checker reviews your translations to detect and notify you of any errors. These include spelling and grammar mistakes, formatting issues, inconsistent placeholders, leading and trailing white space, or and inconsistent HTML. (Lokalise offers 13 different QA checks.)

Grammatical errors can range from weird to offensive depending on the language, which is why a QA tool is important to use with any localization project. It's also critical when localizing any legal, medical, or financial documents.

Similarly, you can incorporate QA with an in-context editor. These tools provide helpful context by showing where and how the translated text will be displayed in your final iteration. When a translator can see the environment in which the word or sentence is used, they can provide a more accurate, natural translation.




For this project, in-context editor was used to translate the call-to-action to Russian

Depending on the size of your localization project, there are plenty of dedicated solutions for this. Lokalise offers an in-context editor for mobile applications (iOS) and websites.

# Machine Translation

The second piece of the localization puzzle is machine translation (MT). As the world localization evolved, organizations and agencies wondered if they would leverage machine learning to translate certain strings of text for which they didn't want to pay translators.

Despite its convenience, MT doesn't provide the same level of quality and consistency as human translators. However, it helps you save money and time with two major use cases:

- To test your product in a new locale for interest and viability
- To "pre-translate" text and save money

MT can help you affordably translate your product and test it in a secondary market before investing in the full localization process.

This tool can also help you pre-translate text using a machine translation engine and then, with post-editing, hire translators to review the MT output for quality. This significantly saves time and money as it costs much more to translate text from scratch.

Examples of MT engines include Google Translate, Yandex Translate, and Microsoft Translate, and some engines are better at translating certain languages. For example, Google Translate is better in German, whereas Yandex Translate is better in Russian. As a result, there are some companies that will aggregate the MT engines and allow you to pick the best one based on what languages you're translating.

As machine learning evolves, so do machine translation engines. The content we input for translation allows them to learn more about languages and how each relates to the other. Today, most MT projects require human translators to review for accuracy, but in the future, that may no longer be the case.

# Integrations

Integrations are important when considering localization tools as the localization process can extend beyond your software or mobile application product. It can also affect components like your marketing materials, technical documentation, CMS and CRM, and even your helpdesk software and customer support experience.

As we discussed in the previous chapter, your localization tool stack should seamlessly integrate with your current tech stack — so much so, in fact, that your focus should not so much be how to build a separate tech stack for localization but instead how to incorporate localization tools to your established workflow.

Integrations can help you do this. While integrations aren't in themselves localization tools, they can facilitate the localization process. When looking for a new localization tool, work with your team to map out your most important integrations and how they can be used with the new tool.

From developer and designer tools to content management and support ticket translation, Lokalise offers 40 integrations, including:

- Design tools like Sketch, Figma, and Adobe XD that allow you to see how your translations impact the layout, design, and overall UX of your product before you invest time in writing the code
- Code repositories like GitHub, GitLab, and Bitbucket that make for easy code importing and exporting
- Engineering tools like Jenkins, Grunt, and Docker that allow you to quickly and seamlessly build your product
- Workflow tools like Asana, TimeCamp, Trello, Slack, and email that assist your project managers with assigning tasks, monitoring projects, and analyzing your team performance

- Support tools like Zendesk and Intercom that translate support tickets and knowledge base documentation and equip your service team to assist customers from all regions, regardless of their primary language

- CMS tools like WordPress and Contentful that allow you to create, publish, and distribute content in multiple regions

If you encounter a localization tool that doesn't offer an integration you need, don't fret. Check that you can create a custom API or Webhooks. This allows you to automate your localization workflow with that technology or software tool.

Beyond the integrations listed above, the Lokalise API can help you build delivery chains or automate otherwise manual localization processes.

## Internationalization

While it's not exactly a localization tool, internationalization is an important component of the overall localization process. Localization can't be done successfully without it.

Internationalization is the process of preparing your product for localization by enabling your code to handle different languages and designs.

The primary step in the internationalization process is taking the text within your application and storing it outside the code itself. Rather than hard-code the text within the software, create resource files by language so you can export and import them without affecting the rest of your code. Then apply placeholders within your code to call back certain terms from your resource files.

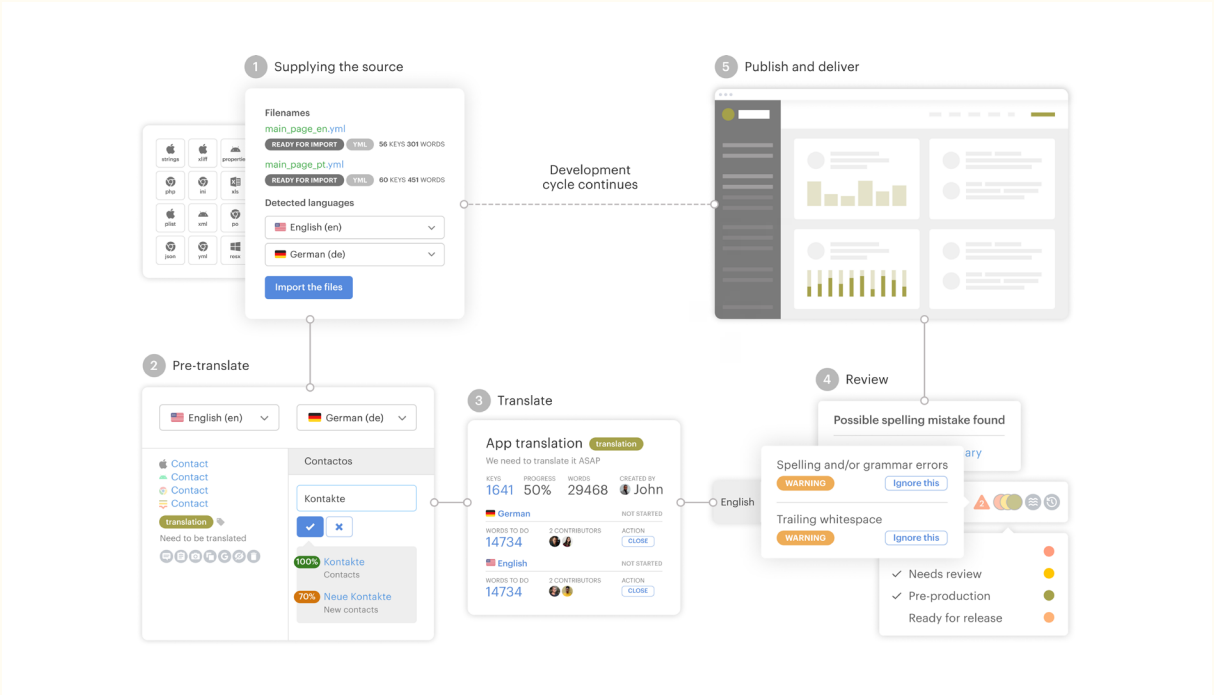
To properly prepare your code for localization, you may need to incorporate other features into your software, such as CSS support for vertical languages, right-to-left languages like Hebrew or Arabic, or other specific language markings, like accents or punctuation.

Once you implement the internationalization process and begin localizing your software, you may notice some places where you didn't code properly to handle the language differences. For this reason, you shouldn't consider your product fully internationalized until you've localized it a time or two. Check out internationalization tools like Globalyzer to add to your tech stack.

As we discussed in the previous chapter, a problem that most of the companies face is failing to consider localization from day one and writing their code for one locale and language. It's quite complex and time-consuming to extract hard-coded text from a software or mobile application and then translate it to different languages. This is why internationalization is so important in the localization process.

# Where Does Lokalise Fit Into the Localization Tech Stack?

You've seen Lokalise mentioned throughout this entire guide. That's because Lokalise is a seamless fit into any localization tech stack. Not only does it incorporate and bundle most of the tools and features mentioned in this list, but it's a remarkable solution for everyone involved in the localization process, from developers to designers and project managers to translators.




At Lokalise, we see our product as a platform. Instead of a product that solves a single need, Lokalise has worked to revolutionize and simplify the localization process from code to delivery and provide most (if not all) of the localization tech stack tools you may need.

Try Lokalise free today

or

Get a custom demo