



A Complete Software Localization Checklist from Crowdin

Start and set up the localization process

- Identify your target languages and locales for localization.
- Choose your translation strategy: professional translators, machine translation engines, or community engagement.
- Invite your team of translators or order professional translations.
- Configure [machine translation](#) engines for faster translation.

Ensure Continuous Workflow

- Connect your version control system ([VCS](#)) with the localization platform to streamline the translation process.
- Set up a [continuous integration](#) process to run localization in parallel with development.

Design and Localization

- Integrate with [design tools](#) like Figma, Sketch, or Adobe XD for smoother collaboration between designers and translators.
- Create designs that allow for text expansion and consider right-to-left (RTL) languages.
- Design flexible layouts to accommodate varying text lengths.
- Run [pseudo-localization](#) to test the impact of translations on the UI.



Translation and Localization Process

- ☐ Ensure correct formatting and consistent use of strings.
- ☐ Create a terminology list to explain key terms used in your product.
- ☐ Enable quality assurance (QA) checks to identify translation errors.
- ☐ Develop a style guide for translators to maintain consistency.
- ☐ Provide context to your translation team using screenshots, in-editor preview, and comments.
- ☐ Use a localization tool that supports standard file formats for seamless import/export.
- ☐ Mark similar strings as duplicates to avoid redundant translation work
- ☐ Mark similar strings as duplicates to avoid redundant translation work

Common Localization Issues to Look For

- ☐ Check for clipped text or text overlapping UI elements.
- ☐ Ensure proper line wrapping for longer strings.
- ☐ Verify correct word breaks and punctuation in translated content.
- ☐ Check layout direction and text direction for languages that read from right to left.
- ☐ Ensure all text is properly translated and nothing is left untranslated.



Do`s

- ✓ Do plan for localization from the early stages of software development.
- ✓ Do use Unicode for character encoding to support multilingual content.
- ✓ Do provide context and resources to translators to ensure accurate translations.
- ✓ Do use professional localization tools and content management systems.
- ✓ Do test the localized software thoroughly for linguistic accuracy and functionality.
- ✓ Do localize user interface elements, error messages, and user documentation.
- ✓ Do maintain communication with translators and provide ongoing support for updates.

Dont`s

- ✗ Don't rely solely on machine translation without human review.
- ✗ Don't overlook the importance of linguistic accuracy and cultural adaptation.
- ✗ Don't neglect testing across different languages, platforms, and devices.
- ✗ Don't assume one-size-fits-all localization; adapt to specific regions and cultures.
- ✗ Don't ignore date, time, and number format differences between locales.
- ✗ Don't forget to update and maintain translation memories for future use.
- ✗ Don't underestimate the time and resources required for successful localization.

Learning Resources

Software Localization:
Taking Your Product to
the Next Level



Apps, Integrations,
and Tools for
Development Teams
Dealing With
Localization



Tips for Developers on
Running Development
and Localization in
Parallel



[Free e-book] Continuous
Localization for Your Team



[Webinar] Spend less time on
localization with Crowdin + Git
integrations



Localize your app with Crowdin

Get started to automate localization, release multilingual versions of your app, and reach global users

Create your free account today

Schedule a demo call