

Going Global for Product Managers

What is globalization, how do I plan for and accomplish it, apply it successfully and scale.

Software Globalization in a Nutshell

What is it?

Software globalization encompasses both internationalization and localization.

Internationalization is primarily an engineering task, and localization is primarily a content transformation task.

- **Internationalization:** The process of building a software application so that it can be adapted to various languages and regions without significant engineering changes.
- **Localization:** The process of adapting internationalized software for a specific region or language by translating text, adapting images and concepts, and adding locale-specific components.

Specific tools exist to help both these activities scale and absolutely should be used to enable the production of global products in Agile environments.

Planning Globalization

Iterative approaches to large software product globalization require detailed planning and strategy. Taking products global also means **evaluating the local market fit for each target market**.

Products may not be the same in all markets/locations due to user **cultural preferences**, local norms, **regulatory concerns**, and **payment preferences**.

Business change timing may be under the control of **governments, regulatory bodies** or **workers** unions in different locations, creating complexity for global organizations.

Different outcomes and timelines for each market / locale are expectable and need to be planned for and implemented accordingly.

To address these differences, SaaS companies need to adopt a model for supporting locales and languages that addresses financial boundaries, regulatory boundaries, as well as geographical boundaries.

Legal and regulatory requirements must be

- understood
- planned for
- communicated broadly
- and tracked

This needs to include where strings live, their systems of record, and recording changes to user experiences. (Reporting on these changes is required by regulatory agencies in some jurisdictions such as Australia.)

Identifying New Markets

SaaS companies entering new markets should do a product fitness analysis for each market expansion program. When expanding into a new market, the following business competencies should be considered:

- can I **market and sell** in a given country
- can I **deliver product service** and **in-language support** to the target country
- who will you be as a company and brand to **attract local users**
- how will your product or service become **an indispensable part** of users' lives
- and what language support is **necessary for users** or **required by law**

Some markets

- require functionality to meet **regulatory or user requirements** that differ from those needed in a product's core market
- require **localization without translation**
- will need **translation for multiple languages** along with localization

For example, [Canada](#) has government regulations that require **French and English** for Canada to be available in product interfaces. US companies can localize but do not need translation when entering the UK or Australia. However, the UK and Australia have different regulatory and legal requirements than the US, so localization is necessary.

To reach out to potential **customers** around the globe, SaaS companies need

- Global **business** acumen
- Global **partners**
- Internationalized **technology** offerings
- **Interoperability** of products and services
- **Interoperability** with local markets

Assessing Language Needs and Costs

There is work to do beyond understanding the user, business, language, and regulatory requirements of each market. The **language needs and costs** across organizations should be factored into any market expansion plan, including

- **marketing** teams
- **sales** teams
- **user support** teams

(Some chat interactions can be dynamically processed with MT, but verbal communications require resources with language skills for each target market.)

The cost of quality and cost of marketing and support are more than the cost of translation.

Planning

There is no universal primer on how and where to start localization. Different organizations have different needs and obligations, offer varied services and focus on different demographics and sources of revenue. Consider the following planning framework.

1. Establish Country and Language Goals

The first step is defining the goals of global expansion efforts. This includes gaining support for those goals and establishing a plan to reach them. Planning requires understanding how a product or service fits in the intended market by **defining who will use them and the needs of these users.**

2. Understand That Localization Is a Growth Engine

Universally, localization should be positioned as a growth opportunity at all levels of the SaaS organization — not an expense. The team that develops products or provides services is the core revenue engine of an organization. **Globalization and localization create the opportunity for global growth.**

SaaS company executives are not looking to spend a lot of money translating words. Executive goals for growth usually include more funding for the product teams, with the expectation that “global will just occur” without budgeting for more growth in the localization team and other groups who need to support this effort. This puts pressure on localization teams to continually automate work

- to keep on top of volumes without adding more internal resources
- and to get the new work done within the old budget

Once SaaS companies saturate their core market, localization enables them to continue growing.

3. Achieve Financial Alignment Across the Company

Finance teams need to understand and accommodate the operating expenses needed to go global. Development costs and timelines must take into consideration the effort it takes for internationalization, localization, and translation.

Agile localization processes require changes to

- internal financial controls
- accounting and auditing processes
- using a rolling funding model in financing mechanisms

Internal localization resources should partner with finance and accounting to show how localization resources add value to the organization, provide revenue, and reduce costs in the functions they support.

4. Share Localization Knowledge

Achieving **goals built on localization business, operational, and technical knowledge enables companies to grow.** These successes are bolstered by sharing localization knowledge.

Specifically

- how to internationalize products
- understanding the work of a language services team
- the meaning localization (l10n) and internationalization (i18n) have to the business

5. Educate Engineering on Best Practices

Globally successful engineering teams excel at

- **Training core developers** to reduce the number of internationalization issues in products by exposing engineers to l10n and i18n problems and how to solve them. Engaging with international customers increases an internal team's understanding of l10n and i18n requirements and issues.
- **Leveraging open-source internationalization libraries and tools**, such as ICU or JavaScript i18n. Engineers can reuse code that has been built and validated by other teams instead of re-inventing the wheel. These existing i18n libraries
 - support 100+ locales
 - required a significant amount of research and development time to build
 - and have already been used by both companies and customers globally
- **Software validation** done by peer review or by using enablement tools such as Globalizer or Artificial Intelligence solutions. These practices
 - prevent i18n problems
 - flag l10n issues before they are pushed downstream
 - and help developers continue to hone their coding skills

When developers know their code will be reviewed, they pay more attention to code quality.

- **Pseudo-localization testing and reporting**, where code is immediately run through a non-translation localization process and automatic testing is run on features and products. This practice connects developers to international user experiences — before they move on to other things, making changes more of a challenge.

6. Create Cross-Functional Teams

Localization teams and PM teams face the common challenge of many internal stakeholders, with an uneven understanding of what localization is for. **Successful localization teams build inroads with many internal groups and partner tightly with PMs** to share knowledge and drive best practices.

Like sporting teams, **language teams develop playbooks (like this) that guide their organizations through the localization process.** Playbooks build understanding and help demonstrate that localization is the voice not only of customers far from corporate HQ, but also internal and domestic customers.

Educational programs that are easy to digest can rapidly bring new team members up to speed and create successful interactions.

Different internal resources need different services. This can mean

- Slack and code repository integration for developers
- Confluence and emails for Project and Program Managers
- Dashboards that provide quick views of ongoing work for leaders

7. Build Localization Awareness Early

In certain cases, localization can be introduced early in content and product design.

Localization can begin before coding starts by starting with wireframes and experience designs. Before the content is written and final you can MT it; product content can be finalized without engineers, and localized independently. This works well when internationalization is a given, and content in all languages is checked into the correct branch in the code repository — then it is always available to be interacted with.

8. Build Effective SaaS-LSP Partnerships

For SaaS companies, being good at localization does not mean having a huge internal localization team. There is a simple solution instead: **treat language expertise as a managed service obtained through partners.**

Regardless of their size, SaaS companies need to consistently maintain focus on their core business and should partner with companies that hold localization and internationalization as their core business. These partnerships solve the pain SaaS companies experience when

embedding localization into their CI/CD process. It helps SaaS companies go global sooner, to localize sooner in their evolution, and not have to “build it all” themselves to do so.

9. Treat Global Users and Local Users Equally

Local and in-country users should not be treated differently when products are planned or problems are found. International issues should not just “go to Jira and die” in a backlog. Catching and **addressing problems as they arise is always better for the business, its resources, and the customer** rather than building a huge backlog of issues. The more problems pile up, the more it seems like there’s not enough time to solve them.

The trick for SaaS companies is catching code and product problems when products are being developed or as code is checked in. SaaS companies can do this by doing product and code validation and using MT for pseudo-localization.

Lack of translation and other user experience problems impede user growth, market traction, and revenue.

When the Jenkins screen turns red, software developers take action. Breaking a build is always bad and not being able to check in is a huge motivator for developers. This requires balance because getting in the way is also a bad idea; developers will push back or may even leave for another organization. Tools and processes must support “the fast fix” for developers and the other team members involved, such as writers, linguists, and quality engineers.

When this happens, the whole team can be all-in on the core business of making great products, from the salespeople to the developers. Localization teams can focus on adding value as well, instead of just telling teams to “give me the files.”

If this part of the business has failed to work for your core market, going global will exacerbate the problem. Well-developed processes that understand users and address their problems accelerates growth when entering new markets.

10. Establish Feedback Loops

Feedback functions need to support different languages and locales for multilingual feedback to be handled correctly by organizations. User-feedback mechanisms on web sites and in products are useful tools for gathering data, but can also set teams up for more work and create false expectations for users.

When SaaS companies are unable to read all the feedback they receive, they have options to

- **disable feedback functions** (not recommended)
- route it to teams with the **correct linguistic skills** to understand it (costly)
- use **machine translation** to understand it (fast*)

Adding MT to the process to understand feedback and provide insights to product and service builders in their language is recommended. Feedback is crucial to the iterative process of releasing more engaging and higher quality content over time.

** Even when teams use MT to convert all feedback into a single language, the source location and feedback language should be retained to ensure it can be used effectively.*

Multilingual Sentiment Analysis

Sentiment analysis produces an important metric for SaaS companies. Most sentiment analysis tools are monolingual. They perform well in one language and can also be used after machine translation into other languages. Feedback from other cultures will not directly align with feedback for a core language due to

- cultural differences
- differences in the features and functions of a product

11. Use Metrics Effectively

SaaS companies look at addressing target markets during both the customer discovery process and after they have onboarded users using metrics such as

- Monthly Active Users
- Total Transaction Volume
- Net Promoter Score
- Engagement, dwell-times, click-throughs, sign-ups, etc.

For companies going global, user KPIs need to include both who and where. Local language and user metrics revolve around

- how well the language used is performing in each supported location
- local revenues
- local sign-ups
- local monthly active users
- local issues and feature gaps
- changes based on local user demographics and local regulations

Getting localization and globalization requirements and KPIs correct ahead of translation prevents surprises, delays and missed targets.

12. Establish Localization Best Practices in Standard Operations

Achieving high-velocity localization in SaaS organizations requires steady collaboration, consistency in processes, and reliable internationalization.

Localizable components have to be ready for the next step to create low-friction movement through language production workflows. To do this, SaaS companies develop guide rails to keep things moving smoothly for software and information developers, testing resources, and linguists.

Coding standards and style guides for content development and localized content are key to streamlining processes because they help clarify expectations and avoid downstream issues that prevent products and features from being released to users.

Often, SaaS companies invest a great deal of time in defining standards for their domestic market, but those new to localization may not put in much effort into thinking about the needs of users in other markets. Planning for regulatory, legal, product, and language requirements for multiple locations needs to be part of the DNA of organizations with global markets.

Not having clear standards and processes

- delays releases
- disappoints customers
- causes rework
- increases external and internal costs
- contributes to burnout in the development, sales, and product teams of SaaS companies

Furthermore, users will cry for help and report bad experiences on social media, creating customer churn and a lot of work for SaaS company staff.

Metrics

High-growth SaaS companies need to be adept at

- **Tracking [metrics](#) such as Net Promoter Score (NPS), total customer value, and customer retention.** Tracking these metrics can be as (or more) important in the early stages of growth than other, more traditional measurements of business success.
- **Understanding the nuances of interpreting data.** For example, NPS is a measurement of how likely a customer is to recommend a product or service. It is

easy to measure and track, but needs to be interpreted sensitively. Cultural factors mean region-to-region comparison is not meaningful (e.g., Japan NPS scores will always be significantly lower than those from the US). [Hofstede's Cultural Index Scores](#) are a good example of quantifying these cultural values and can be used to guide teams in designing for and measuring success internationally.

Operationalizing Globalization

The common thread through the customer life cycle is the customer. The people marketing to, building products for, or supporting customers see only a part of this life cycle.

Localization is the common denominator in virtually all global customer interactions.

SaaS companies new to localization may not anticipate the impacts of localization when planning and implementing products and features. These organizations risk leaving localization and global customer issues to be discovered too late to adapt to without significant work or re-work. **The solution is to consider global markets and localization during planning rather than after development.** Doing this provides visibility into globalization issues before they occur, delivers understanding of internationalization and localization tasks, timelines, and how global customer needs can be addressed.

Real World Application

Example: Adobe in MENA

When Adobe launched their creative applications into Arabic speaking countries, they found that

- localization of **marketing materials was needed** to gain traction in the market
- users needed **help and support materials fully translated**
- users **preferred the user interface to be in English**

This meant that the user-interface references in support and help needed to be in English even though the rest of the documents were translated into Arabic.

Arabic localization is highly complex as

- there is more effort for engineering to support
- there are legal ramifications for errors
- there are a lot of locale-specific dialects across the Arabic-speaking world

Which dialects would be supported was a politically charged decision. Producing a single Arabic offering could offend users in some countries and have significant impact on adoption. Producing Arabic software for each country would be significantly more work. Therefore, revenue and ROI would be unfavorable unless they could thread this needle correctly.

By meeting with users in the target countries, and researching the use of media and advertising across the region, the Arabic common to Egypt was chosen. Egypt is a source for a lot of media across the region, and this version of Arabic was widely understood and accepted.

The analysis resulted in producing a single version of Arabic-language products that could be marketed and sold into the 25 countries that claim Arabic as an official language. Today, Adobe is a highly successful company across the Arabic-speaking world and the largest SaaS company worldwide.