SaaS Localization Readiness Checklist

Is your product ready to go global?

A practical self-audit for product, engineering, and growth teams

simplelocalize.io

How to use this checklist

Who this is for:

- SaaS founders
- Product managers
- Engineering leads
- Localization managers

How to use it:

1. Gather product + engineering (and optionally marketing).
2. Go section by section.
3. Score each item:
    - Not started (0)
    - Partially implemented (1)
    - Fully implemented (2)
4. Calculate your readiness score.
5. Identify weak areas before launching new languages.

SECTION 1 — Strategy & Market Validation

Goals & Justification

- □ We know why we are localizing (growth, retention, expansion, enterprise deals)
- □ We identified target markets based on data (traffic, revenue, demand)
- □ We defined measurable success metrics
- □ We prioritized 1–3 launch languages (not 10)
- □ We validated product-market fit in primary language first

Score (0–10): _______

SECTION 2 — Internationalization (i18n) Readiness

Code & Architecture

- □ All UI strings are externalized
- □ No hardcoded strings remain
- □ Pluralization supports multiple forms
- □ Placeholders are structured correctly
- □ Date/time formats adapt per locale
- □ Currency formatting is locale-aware
- □ RTL layout is supported (if needed)
- □ Language switching works without reload issues

Technical Risk Notes:

Score (0–16): _______

SECTION 3 — UI & UX Localization Readiness

- □ Layout supports text expansion (20–40%)
- □ Buttons and modals resize dynamically
- □ No truncated strings in key screens
- □ Language selector is visible and intuitive
- □ Translated UI tested on mobile
- □ Error messages are localizable
- □ Emails & notifications support localization

UI Gaps Identified:

Score (0–14): _______

SECTION 4 — Workflow & Ownership

Roles

- □ Product owns language prioritization
- □ Engineering owns i18n implementation
- □ Someone owns translation workflow
- □ Someone owns QA review
- □ Release responsibility is defined

Workflow Structure

- □ String extraction process is documented
- □ Translation updates are version-controlled
- □ There is a defined review step
- □ There is a release QA process
- □ No manual copy-paste workflows

Score (0–20): _______

SECTION 5 — Tooling & Infrastructure

- □ Translations are centralized (not scattered across files)
- □ Translation memory is used
- □ Glossary exists for key terms
- □ Version control is integrated
- □ CI/CD supports translation updates
- □ Machine translation policy is defined

Tooling Notes:

Score (0–12): _______

SECTION 6 — Budget & Resource Planning

- □ Translation model selected (human / MT / hybrid)
- □ Ongoing update cost estimated
- □ Review & QA budget allocated
- □ Marketing content included in scope
- □ Support/documentation included in scope

Budget Risks:

Score (0–10): _______

SECTION 7 — Risk Assessment Matrix

Risk Area	Risk Identified?	Mitigation Plan
Technical	□	
UX	□	
Workflow	□	
Ownership	□	
Legal/Compliance	□	

Total Possible Score: 82

Your Score: _______

0–30 → ❌ Not Ready
Focus on i18n and workflow basics before translating.

31–55 → ⚠️ Partially Ready
You can localize, but expect friction and rework.

56–70 → ✅ Ready to Launch Carefully
Start with 1–2 languages and iterate.

71–82 → 🚀 Scalable Localization Ready
You have the foundation for continuous localization.

What to Do Next

If your weakest areas are:

Strategy → Read: Ultimate Guide to Localization Strategy
i18n → Review: Internationalization Guide
Workflow → Implement structured translation management
Tooling → Consider centralizing translations

Teams that centralize localization workflows early typically reduce manual coordination and release friction. Learn more at simplelocalize.io