



Benefits of Continuous Globalization of Software Development

Global product communities are important constituents among technology customers.

Global product communities are important constituents among technology customers. Their needs require attention to internationalizing development efforts to support locale requirements, including language and locale formatting, as well as updating translations for ongoing sprints and releases. These global customers often represent 50% and more of product revenues and growth planning, yet approaches to the demands of software globalization are often ad-hoc, unlike systematic approaches to other product requirements.

Many global companies have not automated globalization tasks encompassing the internationalization and localization of their software development processes.

Error-prone, manually maintained workarounds that are invisible to and out of sync with development team sprints usually are devised. Consequently, development time and costs increase while the application/product quality suffers.

Continuous Globalization will:



Sync internationalization and localization processes with product development and reduce errors that often linger in backlogs



Establish visibility that enables more informed decisions and more cohesive collaboration



Take guesswork out of measurement, so localization managers can better estimate future translation time and costs

Key points:



Lingoport Suite, including Globalyzer, Resource Manager and Dashboard, automates and tracks continuous globalization.



With the Lingoport Suite, the time and expense for globalization are reduced, and localization managers can establish visibility, eliminate human error, measure performance and help speed development.

What does this mean?

- Internationalization or I18N is the planning and implementing products so they can be adapted to local languages and cultures.
- Localization or I10N is the process of adapting content to a specific market.
- Globalization is the process in which things such as goods, ideas, and services spread to different cultures

Introduction

Almost 15 years ago, the Agile Manifesto was drafted, marking the start of a software development paradigm that accelerated development, reduced errors and resulted in applications that end users found truly met their existing and changing needs.¹

Iterative, cyclical coding, integration, testing and release has become the de facto standard. However, globalization – internationalizing the code and localizing (translating) the user interface – to this day is not elegantly dovetailed into the workflow.

Continuous integration, an enhancement that extends agile advantages, can help save weeks of development time and create opportunity cost savings for every feature.

For example, according to a 2014 CollabNet value stream analysis, if a mid-sized enterprise has five development teams doing 20 sprints per year, they would execute 100 sprints per year. If each team works at a rate of five features per sprint, together the teams would produce 500 new features per year. For an enterprise developing 500 new features per year, two days of time saved per feature would translate into 1,000 days saved. Assuming (conservatively) a \$40/hour wage per developer, this translates into \$325,000 saved or a productivity increase of three persons per year.²

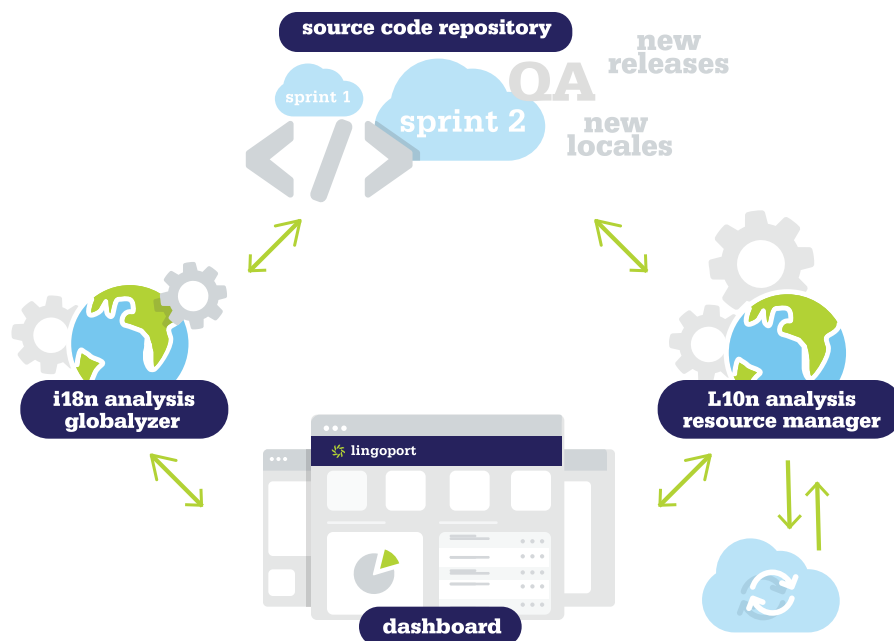
While agile-practicing teams strive to increasingly perfect the methodology, all too often project managers separate globalization, and in particular localization, from the plan. One reason for this is that localization includes translating the source language into several different languages using a resource outside the company. The work is not an automated process that's bolted into the system. It's not part of the QA cycle, even when that testing includes internationalization. Instead, localization managers tend to devise manual workarounds (often an Excel spreadsheet manually populated and maintained or manually administered scripting) that inherently introduce error and wasted time. Sprints occur, builds happen daily, but the translation occurs independently, with the localization manager "babysitting" a spreadsheet and manual processes and reacting to code changes as best as possible.

Ultimately, the disconnect between localization and sprints delays global release, increases the cost for fixing bugs, tends to generate animosity between developers and localization teams, and makes process measurement difficult. The explosion of cloud-based services and APIs makes globalization more imperative than ever before, since hosted applications are more globally accessible. To support growth objectives for an expanding set of enterprises, websites, applications and content needs to have global functionality and local relevance.

Continuous Globalization Process

Continuous Globalization can become integral to the workflow without disrupting development cycles.

Continuous Globalization can become integral to the workflow without disrupting development cycles. Until recently, automating the process was impractical. As discussed, internationalization and localization have been resource-intensive, predominantly manual tasks that require engineers with special skills to find errors and non-optimized translation processes. Continuous Globalization automates the process to streamline the activity and reduce human interaction as much as possible while creating the visibility and measurement that, together, keep the results of each sprint globalized.



Naturally, this begs the question: What does that shift in the process entail? Let's begin with how Continuous Globalization can become inserted into the workflow.

- During a sprint, each iteration of feature code is pushed to the internationalization tool for review. Detected issues are displayed within the developer's environment and on a dashboard. The development team is notified of any problems. The visibility enabled through the dashboard unifies otherwise disparate developers, QA teams and localization managers.
- The source repository is simultaneously reviewed for changes to resource files that need translation and relays that information to the dashboard as well. Resources that need translation are automatically accumulated into bundles, called a "resource kit," which is then verified and put into the translation process, which can include a Translation Management System or localization vendor portal. One advantage in this step is eliminating manual tracking of translation. Human error is avoided and the variation in progress of translation for each language is visible to all stakeholders. This aligns nicely with agile thinking, whereby it is understood that problems, such as differences in the amount of time required getting content translated, will inevitably occur.
- The resource kit is returned to the resource manager. Two very useful outcomes emerge at this point: first, localization process measurement that enables localization managers to better estimate future translation time and costs. Second, and most importantly, it ensures that errors are not reintroduced into the resource bundles destined to go back into the build.
- Finally, the resources are sent back into the feature code, which has also been through the internationalization process.

At this point, the development team sets out on another sprint, turns the software to a group for evaluation and feedback, or releases it to the market.

Experienced enterprise project managers know that there's always resistance when moving legacy process to new approaches. One cause of inertia can be a perceived threat to jobs. Indeed, one of the advantages usually associated with automation is personnel reduction. However, automated localization simply replaces the existing, cumbersome manual process that the vast majority of localization teams use to interact with the workflow. Localization managers find they can focus efforts on orchestrating teamwork and producing quality products that function in multiple languages, instead of dedicating time to the unproductive task of managing spreadsheets and tracking down translations.

Establish Visibility, Speed Development and Improve

Continuous Globalization offers enterprises a variety of benefits. As discussed above, significance opportunity costs can be quantified via value stream mapping.

Furthermore, as any QA manager would attest, the cost for finding and fixing bugs in software increases the longer a team delays testing. “Globalization bugs” impact software development costs just as much. Less tangible yet quite valuable benefits are realized in three primary areas:



Visibility

Visibility establishes better project management and accountability among all teams involved in the process. Through a dashboard, violations and changes can be tracked, bridging the gap between development and localization teams.



Automation

Automation eliminates human error and streamlines the analysis, verification and exchange of localization resource files to the build for staging and linguistic QA.



Measurement

Measurement allows for tracking coding quality and translation timing, so that you can plan and improve processes inside and outside core agile methodology.

With these advantages, project managers at all levels are able to establish a proactive position in which the day-to-day details of any development initiative are known with greater clarity across disparate teams. That position facilitates better decisions in respect to resources, expenses and planning for future development.

Impact at the Enterprise Level

Global enterprises creating custom applications that improve processes will find that Continuous Globalization establishes the competitive advantage they seek sooner than their rivals.

In general, application development tools are helping experienced programmers and even non-developers shave weeks or months off projects.³ The return on the investment in these tools comes in greater efficiencies—the ability to improve the customer experience they shape and more engaged, more capable employees.

Companies developing software as a product will speed time-to-market through Continuous Globalization. Automated, Continuous Globalization supports improved collaboration between developers and localization teams. Ultimately, the products that an enterprise brings to market are more rapidly deployed and successful in terms of functionality, adoption and profitability.

Whether developing software as a product offering or to optimize internal business processes, enterprise brand is improved, costs are better controlled and resources work smarter, anywhere on the globe.

Continuous Globalization Solution: Lingoport Suite

Lingoport's suite of automated globalization solutions has been built to fulfill our core mission of supporting development while managing the complex issues introduced with the linguistic and locale requirements inherent to globalization.

The Lingoport Suite – including Globalyzer software, Lingoport Dashboard and Lingoport Resource Manager – provides development teams and localization managers with the tools they need to establish visibility, eliminate human error, measure performance and speed development.

Learn more

To begin improving how your team can improve software development for global deployment, contact Lingoport at +1.303.444.8020 or visit www.lingoport.com.

References

1. www.agilemanifesto.org/history.html
2. www.collab.net/content/building-value-continuous-integration
3. www.mrc-productivity.com/blog/2015/01/5-key-enterprise-application-development-trends-of-2015