

Business Perspectives on Internationalization **- By Adam Asnes of Lingoport**

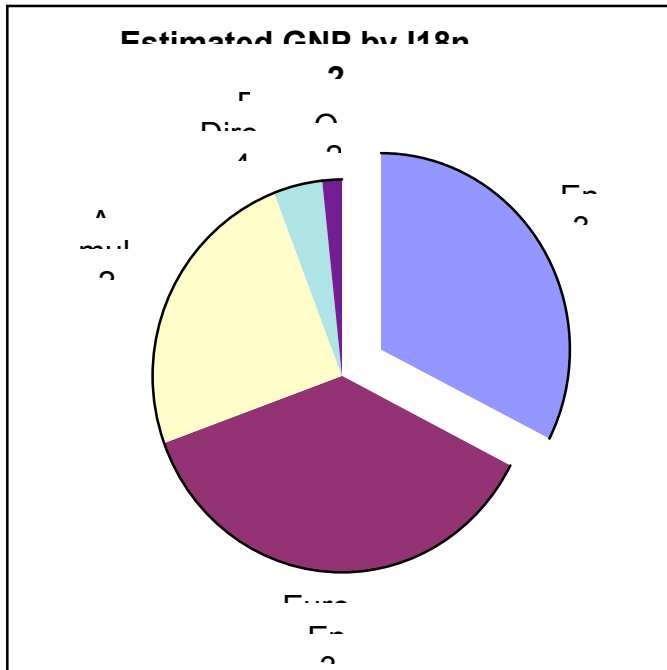
This article appeared in *ClientSide News*, June of 2003:
<http://clientsidenews.com>

We get all sorts of questions regarding the technical aspects of internationalization, but before any of those answers can become truly relevant to a client's efforts, the business decision questions must first be addressed. It's hard for some managers to understand just what is involved with internationalization. Why is it different than translation; what is the benefit of internationalization products and services; how do you measure the benefit; and how do perform it better, faster, and more affordably?

Our company, LingoPort is focused exclusively on software internationalization solutions including internationalization development services, our i18n software products, and technical training. We partner with select localization companies to provide complete globalization services. Given our focus on software development, we are brought into the globalization process differently than a localization company.

I suspect that if you're reading this article, you're involved in some way with the globalization industry, so I'll keep the definitions to a minimum. Internationalization (also referred to as i18n) is important because it lets you efficiently adapt your products for multiple locales, while minimizing support, maintenance and localization costs worldwide. Internationalization consists of any and all preparatory tasks that will facilitate subsequent localization efforts. It can be performed on source code, the controls on a fax machine, a method of writing a document, etc. With regard to source code, internationalization is adapting software to support worldwide character sets and cultural formats (e.g. numerical formats, currencies, date/time formats, etc.), global interface requirements, data access, storage and retrieval and business rules.

Sometimes companies will internationalize their software in stages, thinking in terms of specific locale targets, such as a European markets, the Middle East and Asian Markets. Of course that's an incredibly broad set of language and culture possibilities, but from an engineering process the hurdles involve supporting Latin character sets, Cyrillic languages, bi-directional languages (e.g. Hebrew and Arabic) and what are often referred to as double-byte languages (e.g. Chinese, Japanese, Korean, etc.). From a world market perspective, it's interesting to look at the percentage of world GNP that constituent regions of the world make up, grouped by how you would internationalize (see figure 1).



Source: compiled from world GNP figures from the Encyclopedia Britannica 2001 edition

Internationalization benefits affect both top-line sales opportunities and bottom line cost reductions. Simply put, if you do a good job internationalizing your software, your global customers will get a better quality product and you have an easier time of taking care of them over the years.

Even if you never localize your software, your customers may have global businesses concerns. They may have to support globalized data within their own organizations and customers. Internationalizing back-end functionality of your products will help them store, manipulate and retrieve data no matter the format or character set.

If you do localize, proper internationalization will get localized software into your customers' hands faster. Revenue expectations are often tied to releases and/or specific delivery targets. Ask yourself what the revenue impact is if a product is not delivered on time, or does not meet the necessary specifications for a set of markets. Chances are there are hard numbers in a company's plan to relate to. The quality of an internationalization effort directly affects the quality of a localized product. Particularly in software, quality tends to be an invisible but highly appreciated attribute. A really good interface isn't very noticeable. It just works very well. It stands to reason that to be competitive, you want your products to function gracefully as a user in any locale would expect.

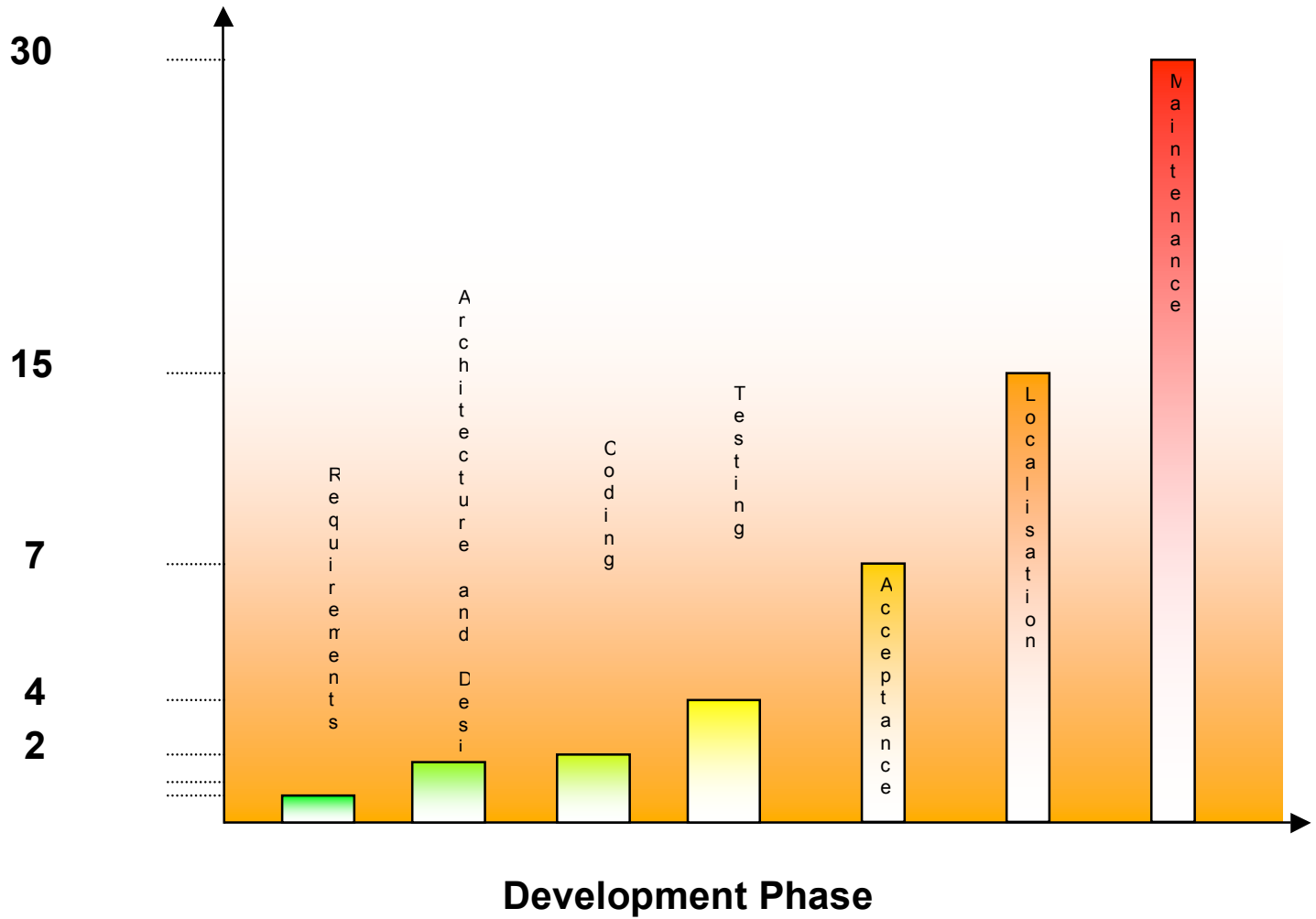
There's a wide range of internationalization cost possibilities depending on existing product technologies, architectural concerns and product lifecycle decisions. In our business we have both customers that are undergoing their first internationalization effort, and others that have been global for years. They have somewhat different issues, but the same ultimate objectives.

Internationalization can scare off an engineering manager quickly. Given all of his or her demanding deliverables, refactoring code may not be very high on the list of things a development manager wants to add to product requirements. Additionally, if a company is just beginning marketing efforts abroad, it's very possible that the bulk of their domestic customers won't regard internationalization as a top priority enhancement that will make them want to buy more software. For some, that makes internationalization particularly appropriate for outsourcing. However, internationalization efforts are often performed in parallel to other development efforts. So outsourcing has to be an extension of your development team. The internationalization effort has to be well defined architecturally and procedurally. You are going to live with any internationalization effort for a long time, so agreeing on requirements, coding standards, build management, code merging, testing and acceptance is critical.

Another route companies will take to lessen the cost impact is to have a distributor internationalize their software with a particular market or set of markets in mind. While this approach typically looks much less expensive at first, it creates expensive support and maintenance problems later. You likely will end up with a forked version of code that must be supported independently and doesn't lend itself well to updates.

It can be quite daunting to revisit up to millions of lines of source code and adapt it to support Unicode and other internationalization requirements. People will refer to internationalization as providing Unicode support, but there's more to it than that. Besides, there isn't just one Unicode method. Optimizing character set handling over an application will vary depending upon how that application gets used and its components. If your development team is undertaking internationalization development efforts, it's hard to know and control all the issues that relate to good internationalization coding practices for immediate tasks and for years to come. Internationalization becomes an ongoing requirement that needs support. We have debuggers, compilers and automated testing tools, but there had been a gap in tools that would help both our own services efforts and our customers' development teams with internationalization.

In performing internationalization services, we needed a way to accurately sort through volumes of code quickly. Our customers had a similar need for an internationalization auditing system that provides diagnostics and "just in time" internationalization answers directly relevant to their own developer teams' tasks. Our Global Investigator software lets us quickly sort through it all and provides a flexible, rules-based system that can support our clients' development teams. In terms of ROI, Global Investigator saves months of diagnostic efforts and greatly improves project estimation. For companies that already consider internationalization an ongoing requirement, Global Investigator provides a system to support their global development goals. Actually, it's this second group that tends to be our best product customers.



We have all heard that bugs get more expensive to fix the further they progress through the development and release cycle. Some might argue that internationalization issues are not actually bugs, as the product may still function in its natively intended programming language. But to a customer or sales team in another locale, they are truly bugs. Figure 2 illustrates the escalation of costs to fix internationalization bugs as a product moves through its globalized development cycle.

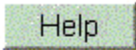
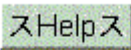
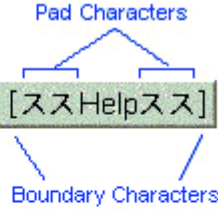
As an example, a well-known printer manufacturer needed to provide internationalization support for new development efforts using .NET technologies, particularly C#. This company recognized that the cost of finding internationalization issues during the localization process is extremely high, forcing resolutions back into development and testing. Of an even greater concern were the costs of missed marketing objectives. They did not want to

miss planned global release dates, or release a product that didn't meet their locale-specific quality standards. They use our Global Investigator software to review source code before completion, and to aid in the ongoing development so that new internationalization issues are not introduced.

Another client, well known for making virus checking and network packet sniffing software, was having trouble addressing various internationalization issues buried in large amounts of C++ and Java. We applied our Global Investigator software to pinpoint those issues and provide help on how to solve them. In this case we also provided internationalization consulting and training.

To add productivity to long and tedious internationalization tasks, we have created implementation tools that speed up the process of string externalization and code refactoring, based on interfacing with our Global Investigator detailed report output. The framework of our string extractor can be quickly adapted to best suit application externalization requirements.

The next challenge for companies is to perform testing. As we saw in figure 2, the cost of fixing an internationalization bug during localization is over three times greater than fixing it during testing. However your team may not have the skills to test a Japanese or Italian product. Since it's before localization, you probably don't even have the translated strings to work with even if you had a tester with the necessary linguistic skills. We have developed pseudo-localization tools that insert gibberish pad characters around the existing strings within the product interface. Now the tester can visually inspect for character set support and

		
Original	Pseudo-Localization Display failed.	Pseudo-Localization Display passed.

interface expansion, based on the requirements of the languages in question.

This graphic shows how performing pseudo-localization lets a tester visually inspect an interface for internationalization compliance.

Code merging is another risk that needs monitoring for any internationalization effort. How this is performed depends on business and technical requirements,

but it needs to be minded closely. I have seen businesses that internationalize as a branch of their code, and that branch never properly makes its way into future product releases. Their global customers essentially are frozen at one release, while the gap to the current release just gets larger with each passing quarter. We prefer to perform regular merges at set intervals to avoid this problem. This also requires good coordination with all development groups.

It's also important to note that some internationalization efforts can be compartmentalized through the application so that parallel localization efforts can be performed, compressing time-to-market. In other words, when one section's internationalization is complete and all strings have been extracted, those strings can be passed to a localization company for translation and testing, while the next section of code is internationalized. This takes tight coordination with your localization company with development and testing efforts. We have done this on multiple occasions with our localization partners - it should be noted, we have a good feel for working with one another.

There are many internationalization war stories, and the engineering process is typically perceived as anything but glamorous. But we do it because internationalization efforts are an important investment in sales opportunities worldwide. With good planning and execution, internationalization can go smoothly and become a routine part of your development process.

Adam Asnes is President of LingoPort, Inc, which specializes in internationalization engineering and training and publishes Global Investigator software. Adam can be reached at aasnes@lingoport.com - visit <http://lingoport.com> for more information