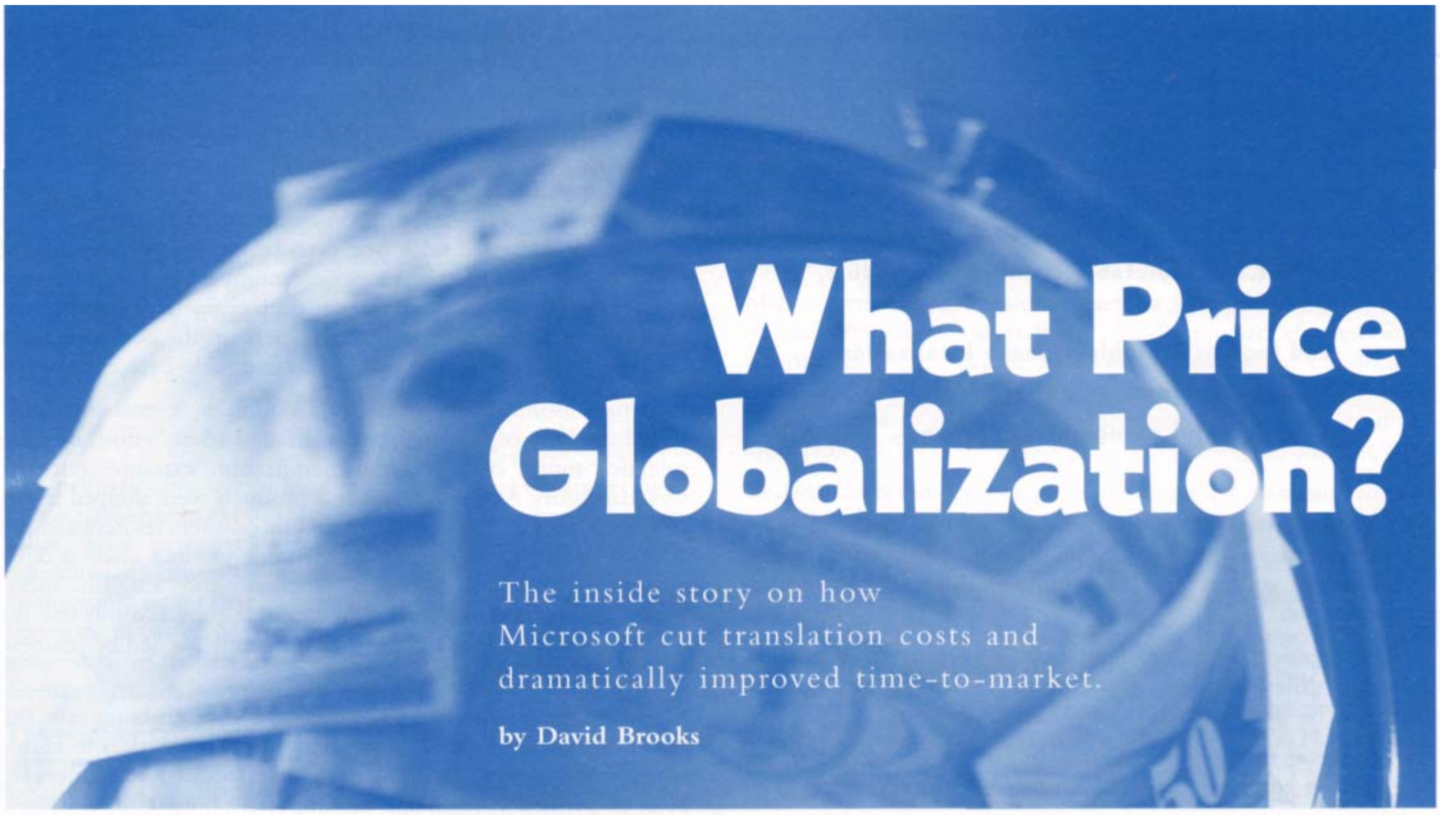

What Price Globalization?

The inside story on how Microsoft cut translation costs and dramatically improved time-to-market.

by David Brooks

In fiscal 1998, over 60 percent of Microsoft's revenues came from markets outside of the United States. The majority of these revenues come from non-English-speaking markets, and a key component of Microsoft's international strategy has been to lead the industry in the delivery of localized products to these markets. In that same period, Microsoft's revenue from localized products exceeded US$5 billion. A mere five years ago, these figures were a fraction of what they are today, and as revenues have grown, so has Microsoft's investment in localization.

As Microsoft's product range and scope of localization grew, executives cringed at the rapidly rising cost of localization. Chairman Bill Gates characterized localization as "just a linguistic process" and expressed frustration over what appeared to him to be runaway cost increases. Worse yet, Microsoft's ability to ship localized products was being constrained: the company was having trouble delivering the breadth of localized products the market wanted and meeting customers' demands for prompt, simultaneous release of localized products.

The Problem

When Gates and Paul Maritz, then head of Microsoft's Platforms and Applications product-development group, began to look into localization in the early '90s, the situation was as follows:

- The company's ability to ship localized products was hampered by delays, bottlenecks inside and outside the company, technical barriers, and costs.
- More products and languages, combined with production bottlenecks, resulted in projects being prioritized into tiers, with the last tier delayed up to a year or more after the US product release.
- The Redmond, Washington product groups had nominal responsibility for localizing their products, but the problem fell mainly on the shoulders of the Microsoft teams in Ireland and Asia.
- With a few exceptions for secondary languages or products, most software was localized by Microsoft staff and outsourcing was limited to manuals and help files.
- Despite the lack of metrics, management believed the company's localization process was inefficient, expensive, and unable to meet market demands.

Under the sponsorship of Maritz, a campaign (or *jihad*, in Microsoft-speak) was launched to bring these problems under control. Maritz asked me to drive this initiative. I had joined his organization several years earlier as director of business operations, and had a background in management consulting and cost management. Needless to say, I faced a steep learning curve in grappling with Microsoft's localization problems.

My first step was to find knowledgeable people in the company's localization community. This led to a "localization summit" in the summer of 1994, which brought together senior localization people from around the company for two days of discussion.

Early in the game it became clear that localization is not really about translation—in fact, it is an extension of product development. Most people think "translation" when they think about localization. In a perfect world, localization would be, as Gates characterized it, "just a linguistic process," but the US-centric design of the US product, unpredictable changes in schedule, and logistical problems make localization a painful reengineering process.

The summit resulted in a meeting with Gates later that year where a game plan for attacking localization costs was outlined. The plan was based on the following findings from the summer meeting:

- Localization is not just translation—in fact, the translation component is the more straightforward part of the job.

In a perfect world, localization would be, as Gates characterized it, "just a linguistic process," but the US-centric design of the US product, unpredictable changes in schedule, and logistical problems make localization a painful reengineering process.

- The majority of localization effort goes into remedial engineering to enable the US product to work with foreign languages, and testing to assure nothing breaks during localization.
- The logistical aspects of localization—keeping track of individual files, managing changes, creating the "golden master"—require extensive project management.
- Changes in the development schedule of the US product wreak havoc on efforts to plan and execute localization projects.
- Localization costs are hard to capture; managing costs was not a priority, and there was little agreement on how to define or measure efficiency.

Maritz's Localization Directive

Maritz focused on the engineering aspects of localization, and the goals he set for the development groups reflected his assessment that US-centric design was the heart of the problem. He distilled this into a directive to the product-development teams:

- Globalize the US product so reengineering is not required during localization.
- Adopt a common set of localization tools and processes across the company.
- Reduce the level of technical skills required to localize products so that external vendors can execute many localizations in parallel.
- Maintain a single codebase across all localized versions of a single product to assure compatibility across languages.

We knew that the hardest task would be breaking through the US-centric mentality of the development community. The US is the most competitive and innovative software market in the world, and software engineers looked to the US trade press and competitive analyses as a report card on their work. Globalization and localizability were not major issues in the US, and in the minds of most developers, these chores were simply a distraction from the challenges of building new, compelling products. Nonetheless, we knew that achieving our goals required bandwidth from the development community dedicated to these issues.

Global Design

Prior to the *jihad* kickoff, localization was handled as an afterthought by downstream localization teams and vendors. Developers and engineers in the core (i.e., US) teams paid little attention to international issues, and their naïve view was that "localization equals translation." They were generally unaware of the biases and engineering limitations inherent in the US product they were building.

One example of engineering limitations is codepage support: all characters, numerals, punctuation marks, and other symbols commonly used in English (and many other Western European languages) fit on a single list of 256 characters (the codepage), each of which is addressed with one byte. Japanese uses thousands of symbols, thus the Japanese codepage numbers in the thousands and requires two bytes to enumerate each character. Therefore, the protocol for addressing specific characters on the codepage must be a two-byte system (also referred to as double-byte encoding).

Codepage issues are pervasive, and double-byte enabling comprised the bulk of the Japanese localization teams' effort. Because so much source code required modification, extensive retesting was mandatory. As a result, Japanese products were shipped a year or more after the English release, and inconsistencies between the English and Japanese versions of the same product made it difficult to share files between Japanese and English users. Furthermore, there was very little available time and too few resources to add features and functions urgently needed by Japanese customers.

Maritz's directive meant building double-byte enabling into the US code, as well as adopting a variety of other practices to eliminate other engineering limitations that required reengineering prior to translation. Although this imposed a burden on US-based developers, it was obvious to Maritz that fixing the problem at the source would be a far more cost-effective and timely solution.

This initiative went under the heading of globalization, and encompassed all aspects of the development cycle, including tasks such as product-setup localization, build process, and test scenarios. Globalization was the cornerstone of the strategy to control costs, and to enable Microsoft to localize more products without a commensurate increase in staffing or costs.

Localizability

The problem proved more complex than we first thought. Even a well-globalized product can be difficult to localize: globalization only assures that a program can accommodate foreign-language data—it doesn't assure, for example, that nothing will break when the user-interface is translated. The majority of these problems are simple, stupid mistakes that should not occur in the first place, but that are almost unavoidable.

For example, a well-globalized product may fail when English commands are replaced with translated versions, because the number of spaces set aside in the code for the command (i.e., the size of the string buffer) may be insufficient to handle the translated command. These problems, which we record and report as bugs, require diagnosis and engineering to identify and solve. It is easy to automatically adjust the size of the string buffer to the size of the text, but hundreds of people may work on a piece of Microsoft software and it is inevitable that someone will forget to do this.

Another frustrating localizability problem is related to the use of hot keys. In many Microsoft products, users can change a text to bold by highlighting it and simultaneously pressing the "Alt" and the "B" keys. This makes intuitive sense in English since "bold" begins with B, but this is not true for other languages. Localizing hot keys like this can be difficult: what do you do when there are several hot-key functions that begin with the same letter—change the name of one of the functions (and, of course, the help files, manual, and all other places where the function is referenced), use an arbitrary key with no mnemonic value and force the user to learn another arbitrary command, or simply eliminate the hot-key function?

Recognizing this class of problems was embarrassing, but resulted in development of tools and procedures to eliminate the problem at the source—i.e., in the core code. Core teams were asked to take direct ownership for German and Japanese localization in order to ferret out localization problems, and developers around the company began building tools to find and eliminate localizability bugs by automatically translating user-interface (UI) elements into a worst-case "nonsense language" before shipment of the US product.

Localization Planning

The primary objective of localization is to meet consumer needs in international markets. Decisions about which products to localize, how extensively to localize them, and the delivery timeframe are driven by market considerations. We define market linguistically rather than geographically: the Spanish market is the aggregate of Spain, Mexico, Argentina, etc.—a total of 22 countries—not just Iberian Spain; "German" is similarly an aggregate of Germany, Austria, and part of Switzerland.

The primary criterion for assigning markets to categories is revenue from localized products. While there is an element of circular logic in using localized-product revenue as a basis for planning, our experience shows this is the best single indicator of sales potential. Other parameters, such as population, have little bearing on the software market, but to assess international markets we do look at the number of PCs sold into certain markets, growth rates in PC sales, and nonquantitative factors such as the degree of protection offered for intellectual property rights.

We use four major categories ("tiers") in the planning process:

Tier 1. The largest international markets (Japanese, German, French) for which the majority of products are localized.

Tier 2. Markets large enough to justify substantial investment in localization (Dutch, Korean, Brazilian Portuguese) but an order of magnitude smaller than Tier 1.

Tier 3. Small but growing markets (Portuguese, Arabic, Hungarian) for which a subset of products with broad appeal is localized.

Microsoft handled localization internally until the early 1990s. A single group in Redmond handled localization for all products, and was staffed with a small army of young college graduates and expatriates. [...] Senior managers and engineers in the core teams were largely unaware of localization, how much it cost, or how it was done.

Tier 4. Emerging markets of limited potential (Thai, Romanian, Vietnamese) for which only core products such as Windows are localized.

For planning purposes we group our products into categories that correspond to market segments—"desktop" products, such as Windows, Office, and Internet Explorer used by individuals for personal productivity; "business systems," such as NT Server and Exchange used by large organizations to run their operations; "development tools" used by engineers and developers to create software, etc.

Market tiers and product categories are arranged in a grid; based on the requirements of each tier and the product characteristics, we decide on the degree or level of localization. Localization levels, beginning with the lowest, are summarized below (each level is incremental, i.e., the localized product is both enabled and localized):

Enabled. Users can compose documents in their own language, but the software user-interface and documentation remain in English.

Localized. The user-interface and documentation are translated, but language-specific tools and content remain in English.

Adapted. The linguistic tools, content, and functions of the software are revised or re-created for the target market.

For a tier-1 market like France, virtually all desktop products are enabled, localized, and adapted with French content. Not only are the user-interface and documentation translated into French, but linguistic, formatting, and stylistic tools like spell-checkers, business-letter wizards, and Internet links (in Internet Explorer) are redesigned to conform to and reflect the tastes and interests of French customers.

A critical dimension of localization planning is the schedule for shipping localized products. This is usually expressed as the number of days between the US-product release date and the localized-product ship date. This delay, referred to as the delta, is critical: products that ship with a small delta (under 30 days) can ride the wave of publicity surrounding the US product launch, while products with longer deltas present marketing challenges, because customers commonly stop buying existing products when new ones are announced.

Microsoft localizes key products such as Windows and Word into 25 or more languages, but when the *jihad* was launched, it was not feasible to execute all projects in parallel. Top priority was assigned to top-tier markets; emerging markets suffered from long delays (up to a year or more) before localized products were available.

Operations

Microsoft handled localization internally until the early 1990s. A single group in Redmond handled localization for all products, and was staffed with a small army of young college graduates and expatriates. The International Product Group (IPG) was far from the mainstream and had little influence on product development. Senior managers and engineers in the core teams were largely unaware of localization, how much it cost, or how it was done. Fortunately, the number of products and languages was small, and international customers were far less demanding than today.

As the scope of localization grew, managers of the product groups became concerned about growth in IPG headcount. Mike Maples, head of the product groups before Maritz, anticipated that the amount of localization work was going to grow rapidly and could not continue in "over-the-wall" mode. Maples subsequently split up IPG, divided its headcount among the product groups, assigned them responsibility for localizing their own products, and encouraged outsourcing tasks such as translation. He also established a small central group to support the product groups' localization efforts by finding suitable translation vendors, negotiating master contracts with the vendors, establishing consistent terminology, etc.

Microsoft executes over a thousand localization projects each year. (A "project" is defined as a product/language combination—e.g., French Windows 98, Polish Internet Explorer 3.0.) We do not view translation itself as a core skill for Microsoft, and based on Maples's directive we began to outsource the majority of translation work. Localization is more than translation, and even the relatively simple task of translating software can introduce problems and complexities. As a result, we adopted a mixed-sourcing model where localization of textual material like printed documentation and online-help files was virtually all outsourced, while software localization remained in-house. Most other software-related activities such as testing, compilation, build, and setup localization also remained in-house.

To handle software localization and oversee our European localization vendors, we set up a dedicated localization operation in Ireland in 1988. The decision to locate in Ireland was based on Ireland's physical proximity to continental Europe, the availability of a well-educated English-speaking workforce, a good telecommunications infrastructure, tax incentives, and Ireland's pro-business attitude. The Irish team quickly emerged as Microsoft's localization experts, and became highly skilled in fixing the many engineering glitches and localization bugs inherent in the US product, developing localization tools, and managing vendors of varying sophistication and skill. A similar process later occurred in Japan and elsewhere in Asia.

Getting a Grip

Gates is fond of quoting Andrew Grove, former chairman of Intel: "If you can't measure it, you can't manage it." This was certainly the case with localization. Prior to the *jihad*, the little available localization-cost data were in the form of accounting records compiled at an aggregate level on a fiscal-year basis. These data failed to show how the money was being spent (e.g., how much on testing, how much on translation), neglected to address the project nature of the work, and offered no insight on the impact of increasing complexity of the company's products, shorter deltas, addition of new languages, etc.

The first step in grappling with the problem was assembling reliable, consistent cost information. Because of the mixed-sourcing model, this meant capturing internal and external costs. To get beneath the macro view, it was also necessary to define cost categories that could be used across a wide variety of products ranging from operating systems like Windows 95 to games like Monster Truck Madness.

External costs were recorded under a variety of accounts in the general ledger, and a paper trail of invoices was also available. Getting a handle on the internal costs proved much more difficult: although costs for the localization teams were recorded in aggregate at the department level, none were categorized by task or assigned to projects.

Once the raw cost data had been assembled and categorized, we still had to invent a way of measuring efficiency. We tried and discarded a variety of approaches, including comparisons of localization costs: a) with revenues; b) with the cost of localizing the previous version; and c) with competitors' costs as best we could estimate them. None of these indicators was especially useful because market size has little bearing on localization cost, products change enormously from release to release, quality improves with experience and turnaround time is shortened, and a multitude of other factors.

We adopted a mixed-sourcing model where localization of textual material like printed documentation and online-help files was virtually all outsourced, while software localization remained in-house. Most other software-related activities such as testing, compilation, build, and setup localization also remained in-house.

What We Found

Prior to this effort, localization-cost data had not been collected in a comprehensive fashion. As a result, management found it difficult to accept what the localization teams had been saying about the amount of effort consumed in testing, building, and simply keeping track of what was going on. Instead, they wondered about the IQ of the localization teams. Once the data from our effort were in hand, the depth of the problem became apparent. As expected, actual costs exceeded the perfect-world benchmark, but how and why they exceeded the benchmark challenged the conventional wisdom about what was driving costs.

Translation of manuals and online-help files was the most visible and easily understood component of localization. As Microsoft products grew in complexity, the size of the manuals grew. Conventional wisdom held this was a major factor behind growth of localization costs. With the cost data from the study, we discovered that, for most products, the cost of translating manuals was a relatively minor part of the total spending and was relatively efficient. The actual cost of localizing the manuals turned out to be a low multiple of the benchmark. Taking into consideration the cost of proofreading, frequent changes, preparation of expensive artwork, and similar costs, we concluded that Microsoft's document-localization process could clearly be improved but was not seriously inefficient. The software, however, was another story altogether.

When the benchmark for software localization was calculated and compared with actual software-localization costs, the results were devastating. Software-localization costs exceeded the benchmark by a factor of 10. Software testing alone cost several times the benchmark, and the same was true for project management and other tasks. Software-translation costs also exceeded the benchmark because of rework, technical review, and graphics work included in the category. The data made it clear that software was the big cost-reduction opportunity (not the ever-expanding manuals and help files, as had been believed), and substantiated the claims by the Irish teams that the cost problem was largely attributable to globalization and localizability problems in the US code.

(continued on page 44)

On the Move (continued from page 9)

Corporations such as GE and Palm Computing are using Global Sight's software, according to company statements. Since its inception in 1996, Global Sight has grown to a staff of 75 with locations in San Jose, Boulder, and New York.

Contact

www.globalsight.com

Deloitte & Touche, VistaTEC in Global e-Business Alliance

Deloitte & Touche is joining with an Irish software globalization company, VistaTEC, to offer customers global e-commerce solutions. The two partners intend to bolster the e-commerce services sector in Ireland by providing a comprehensive globalization service.

Ronan Nolan, head of e-business services at Deloitte & Touche said, "This alliance will enable Deloitte & Touche and VistaTEC to combine their technical resources and business knowledge to overcome the barriers that are inhibiting companies from doing business on the Internet. There has been a lot of talk about the effects of e-commerce on Irish businesses in the future—this alliance will enable us to offer enhanced solutions today."

Bowne Announces e-Commerce Initiatives

Bowne Global Solutions, a unit of Bowne & Co., Inc. (NYSE: BNE) has introduced "gCommerce"—a technology and business solution to help customers take advantage of global e-commerce opportunities.

Bowne Global Solutions' content business unit has developed a global content-management system, code-named "Octopus," to help companies manage the complex editorial and publishing process required for a successful multilingual Web site. Bowne has been previewing Octopus to select customers this quarter.

The company had previously announced a partnership with Idiom Technologies to facilitate the deployment of global Web sites.

Contact

www.bowneglobal.com

EnCompass Launches International e-Business Services

EnCompass Globalization has announced the launch of its global Web services, a suite of e-business services designed to help US companies sell their products and services via the Internet to Japan. The suite of e-business services ranges from Web-site localization and systems internationalization to Japanese payment solutions, order processing, and customer-relationship management.

"There are 20 million people online in Japan, with that number expected to double by 2003. And Japanese is the most frequently used language on the Internet after English," said B.J. Lackland, e-commerce marketing manager at EnCompass.

Contact

www.encompglobal.com

Microsoft Case Study (continued from page 20)

Impact of the Jihad

By attacking the engineering complexity and skill level required for localization, Microsoft has made considerable progress toward Gates's goal of making localization "just a linguistic process." For the sample of products we have been tracking, efficiency gains range between 30–50 percent. Other insights include the following:

Efficiency gains in documentation localization have been modest. The gains are attributable primarily to increased recycling of texts from previous versions, application of technologies such as translation memory, and standardization of authoring tools.

Much of the improvement in software-localization efficiency is attributable to effective globalization of the US product and improvements in localizability as a result of "localization-sufficiency testing" of the US product.

Testing costs remain high, but are anticipated to decline as the testing process itself is globalized. Logistics and project management continue to be significant cost elements.

Sourcing strategies have been revised to optimize utilization of vendor resources. Outsourcing for its own sake has been abandoned, as we have found it more cost-effective to localize certain products (such as high-end operating systems) on-site. In some cases we have partnered with vendors to build dedicated capacity to meet our needs.

Globalization has increased our ability to execute localizations in parallel, allowing deltas to continue to decline. Internet Explorer, for example, now routinely ships dozens of language versions within a matter of days following the US release.

Despite rebalancing our sourcing model, localization vendors are a critical part of our "localization machine." In many parts of the world the vendor base is very thin and is still a constraint on our ability to deliver products to these markets. The industry has undergone a wave of consolidation, increasing the possibility of sourcing multiple-language projects to a single vendor and reducing the logistical burden on Microsoft. The days of a single hand-off remain in the future, as the new generation of vendors is still in the process of assimilating recent acquisitions.

About the Case Study

The above is an introductory excerpt from an extensive case study on Microsoft's globalization strategy, and will appear in full in a book to be published early in 2000 by John Benjamins Publishing Co. in the ATA Scholarly Monograph Series. The book, *Translating Into Success*, features real-life examples of language-technology and management techniques at global companies, large and small.

About the Author

David Brooks joined Microsoft in 1990 as the manager of transfer pricing, and since then has held a variety of jobs with an international focus. Today he is senior director of international product strategy, and his mission is to support and coordinate the creation of international versions of Microsoft products.