Globalisation Practices in the Finnish Software Industry

Jarkko Immonen¹ and Jorma Sajaniemi²
University of Joensuu, Department of Computer Science
P.O. Box 111, FIN-80101 Joensuu, Finland
¹{Jarkko.Immonen}, ²{Jorma.Sajaniemi}@cs.joensuu.fi

Abstract

Finnish software industry representatives were interviewed in order to find out the current state of software globalisation processes and practices. On the basis of the interviews, it seems that localisation often equals to translation. In addition, there seems to be many problems that could be solved by using more defined processes and by enhancing the communication and cooperation between software developers and localisation vendors.

1 Introduction

Today more and more information technology users live outside the traditional software market-areas making software industry a global business. The exponential growth of the Internet has offered software developers a new way of selling and delivering their products to customers all over the world (Esselink, 1998). According to Yeo (2001), many large American software developers earn more than half of their revenue from outside the US. In Finland, the value of software export was about 400 million euros in 2001; 45% of the total software sales (Hietala & al., 2002). The revenues from international business grew by 19% in a year. As tempting as the new markets may seem to be, they also bring new requirements for software developers: a product designed for users in one country may not suit users in other countries and cultures.

For software developers, making a global software product increases the development costs at least 10-20% (Hall, 2002). This is due to education of developers, modification of quality assurance plans, additional testing, etc.
However, developing this kind of application increases sales as well. People prefer software products that are in their native language: according to Kaplan (2000), this is true for up to 70% of German users and nearly all of Japanese users. Similar results were found in a research conducted by Pro Active International of Amsterdam: 80% of Spanish and French users and 60% of Scandinavian users preferred web-sites in their native language (Bradley, 2001). In some cases country specific laws enforce software developers to adapt their products to the target country (e.g., Honold, 2000).

Software globalisation (G11N) is the process of developing, manufacturing and marketing software products intended for world-wide distribution. G11N is a general term which is used to cover two separate processes: internationalisation and localisation. During internationalisation (I18N) language and culture specific items are isolated and extracted, elements that are considered to be offensive or incorrect to some users are displaced, etc. I18N also covers technical issues, such as supporting different kind of character sets, sorting, as well as input and output of international data. Localisation (L10N) is the process of translating and adapting software to a particular language and culture (Esselink, 1998). Its target is opposite of I18N but similar kind of knowledge about culture, language, social values and expectations is needed (Taylor, 1992). Although translation is usually the main part of L10N, it also includes taking care of local conventions, images and icons, symbols, colour associations, etc.

There are usually two participants involved in G11N: a software developer and a localisation vendor. To find out how companies have managed to cope with the situation, we conducted a survey of current G11N practices and G11N related problems in the Finnish software industry. We interviewed ten software developers and localisation vendors to get an overview of the software G11N processes and practices and of the communication and cooperation between software developers and localisation vendors. Moreover, we tried to determine the most common problems and development needs related to software G11N. Yet another goal of this survey was to inquire industry representatives about their views on skills and knowledge central to software G11N and determine the common shortcomings of their employees. This kind of information is useful for improving, for example, translation and computer science curricula in universities.

Many books and articles related to software G11N have been published since the early 1990’s. Literature on I18N mainly concerns general principles and
guidelines for the production of global software as well as technical issues in detail. Examples are Taylor (1992), Kano (1995) and Kaplan (2000). There are also a few L10N books available: Esselink (1998), for example, describes practically how to translate the components of a software product. In addition, there are some papers that focus on guidance for processes and practices for software G11N at the industry level (e.g., Rafii & Perkins, 1995). We know only one survey on real practices in G11N industry: Honold (2000) has studied I18N strategies from the practitioners’ point of view and found problems in eliciting culture-specific requirements for products.

The rest of the paper is structured in the following way. Section 2 concerns the research method used in the survey. The main results of the survey with a discussion on the key findings are presented in Section 3. Finally, the paper ends with conclusions in Section 4.

2 Method

We selected 23 companies, having offices in Finland, representing different sizes, ages, application domains, etc. to be interviewed for the survey. Thirteen of the companies were software developers (SD), and ten companies were localisation vendors (LV). Three companies refused to participate in the survey, and six companies did not respond to the requests. In case of four companies, there were difficulties in arranging the meeting with company representatives. As a result, six SDs and four LVs were interviewed. The interviewees acted in management roles but most of them had additional assignments as well.

The method used in the survey was a semi-structured interview. The interviews were based on two structured questionnaire forms, one for SDs and another for LVs (see Immonen & Sajaniemi (2003) for details). The questions for the questionnaires were mainly formulated on the basis of the G11N literature. All the questions were explained with more details, and the interviewees were given some background information about the topics, if necessary. The interviews were documented on questionnaire forms by the interviewer, and they were also audio-recorded. Two interviewees were conducted using phone, and in these cases, notes were made on the fly by the interviewer. To reduce the time used over phone, the interviewees submitted supplementary material, and some facts were gathered and documented be-
Jarkko Immonen and Jorma Sajaniemi

forehand by the interviewer. The interviews lasted between one and two and a half hours with the average of one hour and 45 minutes.

3 Results and discussion

Detailed results of the survey can be found in Immonen & Sajaniemi (2003). Here we present some generalised results, compare them to suggestions presented in the literature and discuss the survey findings.

3.1 Localised components

We start with considering which components of products are usually localised and how the practices correspond to suggestions presented in the literature. As can be expected, user interfaces and online-helps are the most common localised components of products. Also, documentation or at least a part of documentation is often translated; most commonly marketing material, operating manual, installation guide and getting started guide. This corresponds to level 5 in Kano’s (1995) classification that includes 7 different L10N levels (Figure 1). Levels 6 and 7 were also reached in some projects.

| Less risk, less return | 1. Translate nothing |
| More risk, more return | 2. Translate documentation and packaging only |
|                       | 3. Enable code |
|                       | 4. Translate software menus and dialogs |
|                       | 5. Translate on-line help, tutorials, and sample and README files |
|                       | 6. Add support for locale-specific hardware |
|                       | 7. Customise features for locale |

Figure 1. Levels of localisation (Kano, 1995).

SDs tend to determine the L10N level by balancing risk and return (Kano, 1995). Translating components up to level 5 increases acceptability of a product in the target country and may be enough in many situations. Most software projects covered by the survey were based on customers’ requests which means that the requirements and the properties of target audience are quite well known beforehand. In such cases, localising only components that users are dealing with reduces costs and time-to-market. Furthermore, the
main market-areas of products of the companies were Europe and the US meaning easier G11N than targeting, for example, to the Far East or the Middle East countries. According to Hietala & al. (2002), only 8% of Finnish SDs have their first export country outside Europe and the US.

SDs approached international usability by designing a universal user interface, i.e., an user interface that does not include cultural biases and thus does not require cultural adaptation. This goal can also be seen in the answers concerning customisation of user interface elements. Ito and Nakakoji (1996) have, however, argued that there is no universal interface suitable to any culture. Even though nearly all of the SDs claimed to use usability engineering methods at some level during the development, problems in recognising cultural diversity still exist.

Del Galdo and Nielsen (1996) have defined three levels of requirements for international user interfaces and their production. In the first level, software must be able to process and display the user’s native language, character set, notations and formats. In the second level, user interface and user information should be understandable and usable in the user’s native language, and usability methods must be adapted for use in the target countries and cultures. In the third level, design must address specific cultural models that accommodate users’ cultural characteristics. The practices found in the survey correspond to levels 1 and 2. In many projects, L10N equals to translation and no extensive cultural adaptation is usually done. The results of our survey support del Galdo’s and Nielsen’s contention (1996) that most SDs consider only the most obvious cultural differences, such as language and notations.

3.2 Globalisation process

There were some disparities in defining the terminology used in the survey, and in some cases the terminology was quite unknown to the interviewees. Moreover, only four of the eleven interviewees knew any of the books in the list containing important volumes in this area. We can conclude that most practitioners have familiarised themselves with software G11N by a learn-by-doing method, and their solutions to G11N problems are usually based on practice only. Consequently, a lot of diversity exists in the operations of the companies.
Figure 2 shows the generalised G11N process from development to release in the interviewed companies. The upper part of the figure consists of SDs’ tasks, and the lower part describes a simplified workflow of a typical L10N process.

**Software developers’ process:** SDs seem to make decisions about L10N quite early in the development life cycle. In fact, half of the interviewed SDs made this decision before starting product development. I18N was considered to be a part of development process, rather than a separate process. This corresponds to views presented in the literature that I18N should be applied to a product throughout the development. Thus basic issues related to the production of global software are often tackled at some scale during development starting from the analysis phase.

However, not all I18N issues were taken care of during development since only half of the SDs, for example, often isolated language and culture specific items or supported international hardware. In addition, only two companies often avoided concatenated strings and text in images or supported locales. These shortcomings could stem from the lack of experience and knowledge in the area since some of the interviewees considered even the identification of G11N issues problematic.

There are two approaches to L10N: in-house translation and outsourcing. SDs do not view L10N as a part of the development process but rather as an extra cost. To reduce the cost, they localise products mainly in-house into the languages they speak, and L10N is only partially outsourced to LVs. Native language of products is often English, and texts are written or translated by people who are not language professionals but programmers and designers.
Although all SDs assessed the English language skills of their personnel as good or average, one LV thought that in-house translation might decrease the quality of products. On the other hand, when accomplishing L10N in-house many problems concerning, for example, different tools and file formats can be avoided.

There seems to be two basic reasons for outsourcing: either SDs do not speak the target language or they do not possess the time to localise all the components. L10N seems to be considered purely as a SD’s assignment, and LVs are used only if SDs cannot localise all the components themselves. This approach may be one reason for not contacting LVs until at the end of the development process (even if the decision about L10N has been made earlier) and for not including L10N in project schedules and budgets.

Localisation vendors’ process: SDs’ practices vary a lot but LVs’ basic operations do not differ as much. Their procedures correspond quite well to the L10N process described by Esselink (1998). LVs seem to have quite stable working methods, and the available technology is applied extensively. In fact, most L10N problems appear to be mainly technical and translation itself is rarely a problem. Computer assisted translation tools, such as translation memories, terminology management tools and L10N tools, are used widely and glossaries, for example, are used to improve intelligibility of the product and ensure consistency with the environment and between different versions. However, SDs do not use glossaries when designing a product which makes glossaries less useful.

All LVs proofread the translated documents and perform a linguistic test to ensure the quality of products. Whether other tests are performed depends highly on the type of L10N. If a LV receives only character strings, it is hard or impossible to do any testing apart from a linguistic test.

Communication and co-operation between SDs and LVs: Communication and co-operation between SDs and LVs seems to be minor and occasional despite of its importance as described in the literature. Communication is mainly restricted to business activities, and no extensive interaction between the participants exists. SDs create a small L10N kit or simple instructions for LVs, and LVs report bugs and detected I18N deficiencies to SDs. If some problems occur during L10N, LVs contact SDs for help by using, for example, e-mail. More formal methods, such as documentation of L10N requirements and change management, are put into practice very rarely. This stems
partly from the simplicity of L10N projects but it also reflects the immaturity of the processes.

When SDs outsource L10N, they usually deliver only character strings to a LV. Language specific items are often isolated from the source code to language modules or a database to make it easy to extract strings and send them to a translator. This procedure differs from the L10N literature which usually focuses on binary or text-only resource-file L10N. In addition, this practice creates two additional steps to the G11N process: string extraction and a need to import translated strings back into the product after translation. This practice is also problematic for translators because they cannot benefit from context information; an issue that the translators recognised as a significant problem resulting in lots of corrections. According to the LVs, this problem stems from SDs’ lack of understanding about translation.

3.3 Problems in globalisation

There seems to be problems related to G11N in every phase of product development, starting from analysis and continuing to testing, and affecting both SDs and LVs. Improvement needs recognised in the survey can be roughly categorised as follows:

- technical aspects of I18N
- identification of G11N issues
- L10N practices
- G11N process

Technical aspects of I18N are problematic for many companies that have recently started producing global software. Technical issues are often tied to the used environment and may vary greatly from one platform to another. Solving such problems requires practical experience and theoretical education. In contrast, more experienced companies consider the overall picture of G11N as a more problematic issue. For example, they have problems with identifying what kind of issues should be taken into consideration and what kind of effects they have on the development. The interviews revealed that G11N knowledge of the companies does not always cover all the important issues frequently explained in the literature.

L10N practices and G11N processes are partially intertwined since many L10N problems (e.g., frequent updates, lack of context, lack of tools) are
related directly or at least indirectly to the working methods and practices of SDs. Thus many problems in L10N practices stem from the problems of the other issues listed above. Improving G11N processes may require fundamental changes in the whole software development. The companies seem to be aware of the maturity level of their processes yielding good possibilities for improvement.

3.4 Expectations of universities

The interviewees were asked to assess education needs for employees of both SDs and LVs. Table 1 presents a summary of education needs. The areas in which training is most needed are marked with an asterisk and less important areas with two asterisks. Three asterisks indicate areas in which current training was considered to be sufficient and no additional training was required.

Table 1. Sufficiency of current education.

<table>
<thead>
<tr>
<th></th>
<th>Software engineering students</th>
<th>Translating students</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation / foreign language skills</td>
<td>**</td>
<td>***</td>
</tr>
<tr>
<td>Basics of information technology</td>
<td>***</td>
<td>**</td>
</tr>
<tr>
<td>Localisation</td>
<td>**</td>
<td>*</td>
</tr>
<tr>
<td>Internationalisation</td>
<td>*</td>
<td>**</td>
</tr>
<tr>
<td>Software engineering</td>
<td>*</td>
<td>**</td>
</tr>
</tbody>
</table>

For software engineering students and software engineers, more training was required in G11N and software engineering along with English language skills. The importance of G11N education need is emphasised by the fact that there are not too many courses related to I18N or L10N available. Therefore, an extensive I18N course would be advantageous for software engineering students. A suggestion of course topics contains, for example:

- information about cultural diversity
- general knowledge of I18N and L10N
- technical aspects of I18N using different programming languages and platforms
Unfortunately, there is no good textbook that would cover these topics in a coherent way. Most books (e.g., Kano, 1995; Kaplan, 2000) are tied to some specific programming language or platform and must be accompanied by other materials to provide a thorough presentation.

Furthermore, information about international usability and global software testing should be included in current computer science courses, for example, human-computer interaction and testing courses. In contrast, the insufficiency of software engineering education in computer science curricula is surprising because software engineering courses are being lectured in most universities. Since this need was emphasised by the LVs, we assume that this insufficiency in education is only illusory and reflects the indefinite processes of SDs and problems in LVs’ co-operation with SDs.

For translating students, L10N and information technology were the most wanted subjects of training along with I18N and software engineering. L10N and information technology courses are essential for translating students; nowadays having good language skills is not enough for translators but technical skills are also needed. According to the LVs, translators usually have average or adequate computing skills but a lot of variation exists. Topics for L10N courses can be found, for example, in Esselink (1998). Some important topics are

- the overall L10N process
- user interface L10N
- web-site L10N
- L10N of documentation and on-line help
- content management systems
- L10N project management

Translators also need translation technology skills, and courses should include use of computer assisted translation tools, such as translation memories, terminology management tools and concordance tools. In addition, some information about I18N and software engineering would help translators to understand the whole software development life-cycle and the problems in it. It might be a good idea for both software engineering and translating students to co-operate with each others in common projects during their studies.
The interviewees were also asked to give proposals for new research topics related to software G11N and to specify which kind of new tools they would like to have. The basic problem for many SDs seems to be the identification of G11N issues. Additionally, applying different techniques to practice was not considered as an easy task. Therefore, one common proposal for a research topic was the whole process of producing a global software product. This is too extensive as such but many interesting subtopics can be found. LVs seem to have similar research interests since their proposals were almost the same as those proposed by the SDs but they approached the topics from a different viewpoint. LVs were also interested in L10N and translation tools. They found current tools expensive, complicated to use and missing proper features.

4 Conclusion

We interviewed industry representatives to find out the current state of software G11N practices in the Finnish software industry. This paper presented an overview of the results and discussed the impact of the findings on G11N efforts. Detailed facts can be found in Immonen & Sajaniemi (2003).

On the basis of the interviews, G11N practices vary from one company to another, and solutions seem to be based mainly on practice. SDs do not view L10N as a part of the development process but rather as an extra cost. To reduce the cost, SDs localise products mainly in-house into languages they speak. In many projects L10N equals to translation and no extensive cultural adaptation is usually done. In a SD’s opinion, LVs’ only task is to translate the product and its components, and they do not expect LVs to take care of any other aspects of adaptation. If some additional adaptation beside translation is needed, it will be handled by the SD. Many problems seem to be caused by inadequate and occasional communication and co-operation between SDs and LVs.

Acknowledgements

The authors would like to thank all the companies and interviewees who participated in the survey for their valuable time.
References


